



UNIVERSIDAD TECNOLÓGICA ISRAEL
ESCUELA DE POSGRADOS “ESPOG”

MAESTRÍA EN
MENCIÓN: ELECTRÓNICA Y AUTOMATIZACIÓN
Resolución: RPC-SO-09-No.265-2021

PROYECTO DE TITULACIÓN EN OPCIÓN AL GRADO DE MAGISTER

Título del proyecto:
Implementación de un sistema de seguridad domiciliario empleando Arduino
Línea de Investigación:
Ciencias de la ingeniería aplicadas a la producción, sociedad y desarrollo sustentable
Campo amplio de conocimiento:
Ingeniería, industria y construcción
Autor/a:
Martha Verónica Pintado Vaca
Tutor/a:
Mg. Wilmer Albarracin. PhD Maryory Urdaneta.

Quito – Ecuador

2025

APROBACIÓN DEL TUTOR



Yo, Maryory Urdaneta Herrera con C.I: 1759316126 en mi calidad de Tutor del proyecto de investigación Implementación de un sistema de seguridad domiciliario empleando Arduino

Elaborado por: Martha Verónica Pintado Vaca, de C.I: 1718395609, estudiante de la Maestría: MENCIÓN: ELECTRÓNICA Y AUTOMATIZACIÓN de la UNIVERSIDAD TECNOLÓGICA ISRAEL (UISRAEL), como parte de los requisitos sustanciales con fines de obtener el Título de Magister, me permito declarar que luego de haber orientado, analizado y revisado el trabajo de titulación, lo apruebo en todas sus partes.

Quito D.M., 9 de septiembre de 2024

Firma

APROBACIÓN DEL TUTOR



Yo, Wilmer Albarracin con C.I: _____ en mi calidad de Tutor del proyecto de investigación
Implementación de un sistema de seguridad domiciliario empleando Arduino

Elaborado por: Martha Verónica Pintado Vaca, de C.I: 1718395609, estudiante de la
Maestría: MENCIÓN: ELECTRÓNICA Y AUTOMATIZACIÓN de la UNIVERSIDAD TECNOLÓGICA
ISRAEL (UISRAEL), como parte de los requisitos sustanciales con fines de obtener el Título de
Magister, me permito declarar que luego de haber orientado, analizado y revisado el trabajo de
titulación, lo apruebo en todas sus partes.

Quito D.M., 9 de septiembre de 2024

Firma

DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE



Yo, Martha Verónica Pintado Vaca con C.I: 1718395609, autor/a del proyecto de titulación denominado: Implementación de un sistema de seguridad domiciliario empleando Arduino. Previo a la obtención del título de Magister en Electrónica y Automatización Industrial.

1. Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar el respectivo trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Manifiesto mi voluntad de ceder a la Universidad Tecnológica Israel los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor@ del trabajo de titulación, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital como parte del acervo bibliográfico de la Universidad Tecnológica Israel.
3. Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de prosperidad intelectual vigentes.

Quito D.M., 25 de agosto 2024

Firma

Tabla de contenidos

APROBACIÓN DEL TUTOR	¡Error! Marcador no definido.
DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE	¡Error! Marcador no definido.
INFORMACIÓN GENERAL	1
Contextualización del tema	1
Problema de investigación	2
Objetivo general	3
Objetivos específicos	3
Vinculación con la sociedad y beneficiarios directos	3
CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO	5
1.1. Contextualización general del estado del arte	5
1.2. Proceso investigativo metodológico	6
Capítulo II: (Marco teórico y conceptual)	8
2.1. Contextualización general del estado del arte	8
Capítulo III: (Metodología)	9
3.1. Diseño y arquitectura del sistema	9
3.2. Especificaciones técnicas	10
3.3. Diagrama de procesos	12
3.4. Configuración de Microcontroladores	13
3.4.1. Ajustes y configuración del Microcontrolador Arduino	13
3.4.2. Ajustes y configuración del Sensor infrarrojo pasivo HC-SR501	17
3.4.3. Ajustes y conexión de Cámaras de Seguridad Nexxt Solutions	19
3.4.4. Conexión y Configuración de Sensores de Puertas/Ventanas	23
3.4.5. Ajustes y configuración Modulo de Comunicación WI-FI (ESP8266/ESP32)	24
3.4.6. Conexión y Configuración de Sirena de Alarma	26
3.4.7. Conexión y Configuración de la Pantalla LCD o LED	28
3.4.8. Conexión y Configuración de Controladores de Acceso	28
3.4.9. Conexión y Configuración de la Fuente de alimentación	29
Capítulo IV: Implementación	31
4.1. Arduino	32
4.1.1. Entradas y salidas	33
4.1.2. Electrónica y Automatización	34
4.1.3. Internet de las Cosas (IoT)	34
4.1.4. Comunicaciones inalámbricas	34
4.1.5. Programación y desarrollo de software	35

4.2.	Descripción del sistema	35
4.2.1.	Descripción de la propuesta	42
4.3.	Descarga E Instalación de ARDUINO IDE	45
4.4.	Descarga E Instalación de PYTHON	51
4.5.	Descarga E Instalación de Visual Studio Code	52
4.6.	Instalación de librerías	57
4.7.	Clasificadores en Cascada (Haar Cascade) de OpenCV	59
4.8.	Codificación Inicial del Arduino	61
4.9.	Conexión entre Python y Arduino	70
4.10.	Despliegue del Aplicativo	70
4.11.	Costo	73
4.12.	Análisis de Rentabilidad	75
	Conclusiones	77
	Recomendaciones	78
	BIBLIOGRAFÍA	79
	ANEXOS	80
	Anexo 1. Encuesta	80
	Anexo 2 Manual técnico	82
	Anexo 3 Manual de usuario	89

Índice de figuras

Figura 1. Microcontrolador Arduino	13
Figura 2. Sensor infrarrojo pasivo HC-SR501	17
Figura 3. Funcionamiento sensor.....	18
Figura 4. Opciones de sensor	18
Figura 5. Cámaras de Seguridad Nexxt Solutions.....	19
Figura 6. Especificaciones técnicas cámaras	21
Figura 7. Características físicas cámaras	22
Figura 8. Módulo de Comunicación WI-FI (ESP8266/ESP32)	24
Figura 9. Guía de pines del ESP8266	25
Figura 10. Configuración red	26
Figura 11. Arduino.....	32
Figura 10. Entradas y salidas	33
Figura 13. Sensores de Movimiento PIR	36
Figura 14. Cámaras de Seguridad IP	37
Figura 15. Sensores Magnéticos de Contacto	38
Figura 16. Módulo de comunicación wifi ESP8266 o ESP32	38
Figura 17. Sirena Piezoeléctrica	39
Figura 18. Pantalla LCD 16x2 o Pantalla OLED	39
Figura 19. Módulo RFID-RC522 o Teclado Matricial	40
Figura 20. Módulo RTC DS3231.....	41
Figura 21. Adaptador de Corriente 9V 1A.....	41
Figura 22. Buzzer Piezoeléctrico	42
Figura 23. Descarga Arduino	46
Figura 23. Opciones de descarga Arduino	47
Figura 25. Instalación Arduino	47
Figura 26. Proceso de instalación Arduino.....	48
Figura 27. Proceso de instalación Arduino 1.....	48
Figura 28. Proceso de instalación Arduino 2.....	49
Figura 29. Proceso de instalación Arduino 3.....	49
Figura 30. Proceso de instalación Arduino 4.....	50
Figura 31. Proceso de instalación52.....	50
Figura 32. Instalación Phytton.....	51
Figura 33. Proceso de instalación Phytton	52

Figura 34. Proceso de instalación Visual Code.....	52
Figura 35. Proceso de instalación Visual Code 1	53
Figura 36. Proceso de instalación vinculación Visual Code y Phyton.....	54
Figura 37. Visual Code y Phyton 1	55
Figura 38. Soporte de Serial	55
Figura 39. Configuración Python.....	56
Figura 40. Configuración de Arduino en VSCode	56
Figura 41. Pip Install opencv-python.....	57
Figura 42. Pip install opencv-python-headless	57
Figura 43. Pip install numpy	57
Figura 44. Pip install pygame	58
Figura 45. Pip install pillow	59
Figura 46. Pip install pyserial requests.....	59
Figura 47. Librerías Necesarias Arduino.....	61
Figura 48. Definición Pines.....	62
Figura 49. Configuración del Setup	62
Figura 50. Función loop.....	63
Figura 51. Configuración de la Conexión Wi-Fi	64
Figura 52. Envío de datos	66
Figura 53. Lectura del Sensor de Puerta/Ventana	67
Figura 54. Integración con el Sistema de Seguridad.....	68
Figura 55. Conexión entre Python y Arduino	70
Figura 56. Despliegue del Aplicativo	71
Figura 57. Alarma Sound	71
Figura 58. Mensaje de alerta	72

Índice de tablas

Tabla 1 Diseño y arquitectura del sistema	9
Tabla 2 Especificaciones técnicas.....	10
Tabla 3 Descripción de la propuesta	35
Tabla 4. Costos	73

INFORMACIÓN GENERAL

Contextualización del tema

El área de conocimiento en la que se desarrollará este proyecto se enmarca en el ámbito de la seguridad electrónica y la automatización del hogar, disciplinas que han experimentado un crecimiento significativo en los últimos años gracias al avance de las tecnologías de bajo costo como Arduino. La seguridad electrónica se enfoca en la protección de espacios físicos a través del uso de dispositivos tecnológicos como sensores, cámaras y sistemas de alerta, que pueden ser integrados en una red controlada para monitorear y responder a eventos sospechosos. La automatización del hogar, por su parte, implica la implementación de tecnologías que permiten a los usuarios gestionar y controlar diferentes aspectos de su vivienda de manera remota o automatizada. Este proyecto busca unir ambas áreas para crear un sistema de seguridad domiciliario que sea accesible y personalizable, dirigido a mejorar la protección en hogares de distintos perfiles socioeconómicos.

El proyecto se llevará a cabo en un entorno residencial, enfocado en la implementación de un sistema de seguridad avanzado. Este entorno cuenta con el equipo necesario para la instalación y prueba de dispositivos tecnológicos, como kits de desarrollo Arduino, sensores, cámaras, y otros componentes electrónicos esenciales para la creación del sistema de seguridad. Además, se cuenta con acceso a recursos digitales y bases de datos en línea, lo que permitirá consultar fuentes académicas y técnicas relevantes para el desarrollo del proyecto. La experiencia en áreas como la programación, electrónica, y redes de comunicación será clave para guiar el proceso de diseño y prototipado del sistema.

El equipo involucrado en este proyecto estará compuesto por estudiantes de último año de la carrera de Ingeniería Electrónica, apoyados por un grupo de docentes especializados en electrónica y automatización. Se estima que el grupo de trabajo incluirá entre 5 y 7 estudiantes, quienes asumirán roles específicos relacionados con el diseño del hardware, la programación del software, la integración de los sistemas, y la evaluación del prototipo en escenarios reales. Los docentes, por su parte, proporcionarán supervisión técnica, revisando la calidad y viabilidad de las soluciones propuestas, y ofreciendo orientación en el uso de metodologías adecuadas para la investigación y desarrollo del proyecto.

En términos generales, el ambiente de trabajo en la institución es colaborativo y está orientado a la innovación. La institución promueve un enfoque práctico en la educación, donde los estudiantes aplican los conocimientos adquiridos en proyectos reales que resuelven

problemas actuales. Este enfoque práctico es altamente valorado tanto por los estudiantes como por la industria, ya que prepara a los futuros profesionales para enfrentar los desafíos tecnológicos del mundo real. La implementación de este proyecto no solo contribuirá a la formación académica de los estudiantes involucrados, sino que también proporcionará un valor agregado a la comunidad, al ofrecer una solución de seguridad que puede ser replicada y adaptada en diferentes contextos.

Problema de investigación

Actualmente, muchas viviendas en zonas urbanas y suburbanas enfrentan un desafío creciente: la necesidad de proteger sus hogares contra robos y allanamientos. Aunque existen sistemas de seguridad avanzados en el mercado, estos suelen ser caros y difíciles de personalizar para las necesidades específicas de cada hogar. Como resultado, muchas familias no pueden acceder a estas soluciones, lo que las deja vulnerables a posibles intrusiones.

El problema principal radica en la falta de opciones de seguridad domiciliarias que sean accesibles y efectivas, y que puedan ser instaladas y gestionadas directamente por los propietarios sin necesidad de recurrir a servicios costosos o técnicos especializados. Los sistemas comerciales existentes a menudo no se adaptan bien a las características particulares de cada vivienda, lo que limita su efectividad y la capacidad de los usuarios para responder adecuadamente en caso de emergencia.

Además, estos sistemas de seguridad comerciales tienden a ser complejos y requieren mantenimiento profesional, lo que aumenta los costos para los usuarios y reduce su autonomía. Esta dependencia de servicios externos no solo incrementa los gastos, sino que también puede retrasar la respuesta a situaciones de riesgo, poniendo en peligro la seguridad de las personas y sus bienes.

Si este problema no se aborda, es probable que la tasa de robos y allanamientos continúe aumentando en hogares que no pueden permitirse sistemas de seguridad adecuados, lo que aumentará la sensación de inseguridad y desconfianza en las comunidades. Esto también perpetuará una desigualdad en el acceso a tecnologías de protección, donde solo quienes tienen mayores recursos económicos podrán proteger sus hogares de manera efectiva, mientras que los demás seguirán siendo vulnerables.

Por estas razones, es necesario desarrollar un sistema de seguridad domiciliario basado en la plataforma Arduino que sea accesible, personalizable y fácil de instalar. Este sistema permitirá a los propietarios gestionar su seguridad de manera autónoma, ofreciendo una solución práctica

y asequible para proteger sus hogares. Además, al ser una tecnología abierta y flexible, permitirá a los usuarios adaptar el sistema a sus necesidades específicas, mejorando así la seguridad en una mayor cantidad de hogares.

Objetivo general

Diseñar un sistema de seguridad domiciliario basado en la plataforma Arduino, que permita mejorar la protección de los hogares mediante la detección automática de intrusiones, facilitando la gestión autónoma de la seguridad por parte de los propietarios y reduciendo la vulnerabilidad ante robos y allanamientos.

Objetivos específicos

- Detallar los fundamentos teóricos sobre sistemas de seguridad domiciliarios, identificando las tecnologías actuales, incluyendo sensores y actuadores, que pueden ser implementadas utilizando la plataforma Arduino.
- Diagnosticar las necesidades específicas de seguridad en hogares, estableciendo los requerimientos técnicos y funcionales para el diseño del sistema de seguridad domiciliario basado en Arduino.
- Diseñar un sistema de seguridad domiciliario que integre sensores, actuadores y un módulo de control utilizando Arduino, adaptado a las características y demandas de seguridad de los hogares.
- Validar el funcionamiento del sistema de seguridad mediante pruebas controladas, evaluando su eficacia en la detección de intrusiones y la respuesta automatizada, y ajustando el diseño según los resultados obtenidos.

Vinculación con la sociedad y beneficiarios directos

El proyecto de titulación, centrado en el diseño e implementación de un sistema de seguridad domiciliario basado en la plataforma Arduino, tiene un impacto significativo en la comunidad al ofrecer una solución tecnológica accesible para mejorar la seguridad en los hogares. Este sistema contribuye a la protección del patrimonio familiar y la integridad de las personas, especialmente en sectores donde el acceso a tecnologías de seguridad avanzadas es limitado. La utilización de recursos tecnológicos como sensores, actuadores y sistemas de control automatizados permite optimizar la respuesta ante situaciones de riesgo, reduciendo la vulnerabilidad a robos y allanamientos. Además, el proyecto promueve la conciencia sobre la importancia de la seguridad en el hogar, incentivando a más familias a adoptar medidas preventivas y tecnológicas para su protección.

El desarrollo de este proyecto también proporciona un espacio para la capacitación y el aprendizaje, tanto para los estudiantes involucrados como para la comunidad en general. A través de la divulgación de conocimientos sobre tecnologías accesibles como Arduino y su aplicación en la seguridad domiciliaria, se fomenta la formación de profesionales con habilidades en automatización y electrónica aplicada. Además, el proyecto puede dar lugar a la creación de materiales educativos, publicaciones y guías que sirvan de referencia para futuras investigaciones o para personas interesadas en implementar soluciones de seguridad en sus propios hogares. Esta iniciativa no solo beneficia a los propietarios de viviendas, sino también a otros actores como técnicos en electrónica, personal de apoyo en instalaciones, y miembros de la comunidad que buscan mejorar su conocimiento en seguridad tecnológica.

Los beneficiarios directos de este proyecto son los propietarios de viviendas que implementarán el sistema de seguridad diseñado, estimándose que un número considerable de hogares podría adoptar esta tecnología debido a su bajo costo y facilidad de instalación. Adicionalmente, técnicos y especialistas en electrónica, así como estudiantes de carreras tecnológicas, se beneficiarán del conocimiento generado y de la experiencia práctica en el desarrollo e implementación de sistemas de seguridad automatizados. La comunidad en general también se verá impactada positivamente, al tener acceso a una solución de seguridad más asequible y efectiva, lo que contribuirá a un entorno más seguro y a una mayor sensación de tranquilidad en los barrios donde se implemente el sistema.

CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO

1.1. Contextualización general del estado del arte

La implementación de un sistema de seguridad utilizando Arduino surge como respuesta a la necesidad de contar con soluciones de seguridad accesibles y personalizables. En este proyecto, el análisis del problema se centra en identificar las principales vulnerabilidades que existen en los sistemas de seguridad domésticos y comerciales, donde muchas veces se observa una falta de opciones adaptables y de bajo costo. Para abordar esta problemática, la investigación se desarrollará específicamente dentro del ámbito de los sistemas de seguridad electrónica, enfocándose en la integración de sensores y microcontroladores como el Arduino. Los recursos documentales que se emplearán en este estudio incluyen artículos académicos, tesis, y reportes técnicos, seleccionados en función de su relevancia técnica, cronológica, y aplicabilidad, especialmente aquellos publicados en los últimos cinco años. Este enfoque nos permitirá contextualizar adecuadamente la documentación utilizada, siguiendo criterios específicos de búsqueda que aseguren la relevancia y actualidad de los estudios seleccionados. Además, se tendrá en cuenta las tendencias recientes en la investigación, con la intención de aplicar los resultados obtenidos al entorno local y explorar cómo estas innovaciones pueden transferirse y adaptarse a las necesidades reales de seguridad.

Para fundamentar este proyecto, se han revisado conceptos clave relacionados con la seguridad electrónica, la automatización de sistemas mediante microcontroladores, y la conectividad de dispositivos en redes IoT (Internet de las Cosas). Entre los autores más relevantes en esta área, se encuentran Massimo Banzi y Michael Margolis, quienes son pilares en el desarrollo y popularización de la plataforma Arduino, así como Simon Monk, que ha trabajado extensamente en aplicaciones prácticas de Arduino para sistemas de seguridad. Además, se han revisado estudios recientes, como el realizado por (Gutiérrez, 2021), quien implemento un sistema de vigilancia con Arduino en áreas rurales de Bangladesh, utilizando cámaras y sensores de movimiento, obteniendo resultados significativos en la reducción de incidentes de robo. Otro estudio destacado es el de (Portilla, 2021), en el que se desarrolló un sistema de monitoreo de incendios basado en Arduino, demostrando la eficacia de estos sistemas en la detección temprana de incendios en edificios comerciales. Estos trabajos proporcionan un marco sólido para nuestro proyecto, permitiendo aprovechar tecnologías y metodologías que han demostrado ser efectivas en contextos similares.

El análisis de estos estudios previos nos permite orientar nuestra investigación hacia la creación de un sistema de seguridad basado en Arduino que no solo sea accesible y fácil de

instalar, sino que también incorpore tecnologías emergentes como la comunicación IoT para mejorar la eficiencia y capacidad de respuesta del sistema. Al revisar las soluciones existentes, hemos identificado la oportunidad de integrar mejoras en la detección de intrusos y notificaciones en tiempo real, así como la posibilidad de añadir funcionalidades adicionales como el control remoto desde dispositivos móviles. Esta orientación se fundamenta en la clasificación de ideas previas, adoptando lo más relevante de cada investigación, y añadiendo valor mediante la integración de nuevas tecnologías y métodos que no han sido completamente explorados, como la inclusión de sistemas de energía renovable para aumentar la sostenibilidad del sistema de seguridad. Con esta base, nuestro proyecto tiene el potencial de contribuir de manera significativa al campo de la seguridad electrónica, ofreciendo una solución innovadora y adaptable a las necesidades específicas del contexto ecuatoriano.

1.2. Proceso investigativo metodológico

El proceso investigativo del proyecto de titulación sobre el diseño e implementación de un sistema de seguridad domiciliario basado en Arduino sigue un enfoque de investigación aplicada. Este enfoque tiene como objetivo resolver un problema práctico mediante el desarrollo de una solución tecnológica que pueda ser implementada en un contexto real, en este caso, la mejora de la seguridad en los hogares. La investigación se caracteriza por su intención de generar resultados concretos y útiles, combinando enfoques cuantitativos y cualitativos para evaluar la eficacia del sistema propuesto y adaptarlo a las necesidades específicas de los usuarios.

Tipo de investigación aplicado:

Para abordar la investigación, se empleará un diseño experimental. Este diseño se organizará en dos grupos para evaluar el impacto del sistema de seguridad. El Grupo de Control estará compuesto por viviendas que no recibirán la implementación del sistema de seguridad. Este grupo servirá para establecer una línea base y observar las condiciones normales de seguridad sin la tecnología propuesta. Por otro lado, el Grupo de Estudio Experimental recibirá la instalación del sistema de seguridad basado en Arduino, que incluye componentes de hardware y software integrados. El objetivo es observar y comparar los efectos del sistema en términos de mejora de la seguridad y capacidad de respuesta ante eventos de intrusión.

Para el desarrollo del proyecto, se utilizaron diversos métodos teóricos y prácticos. Teóricamente, se realizó una revisión exhaustiva de la literatura sobre seguridad domiciliaria y tecnologías de automatización. Este análisis permitió identificar las mejores prácticas y soluciones existentes en el campo. Prácticamente, se adoptó un enfoque iterativo para el diseño

y prueba del sistema. Se crearon prototipos del sistema de seguridad que fueron evaluados en condiciones controladas. Cada iteración del prototipo permitió ajustar y optimizar el diseño, garantizando que el sistema cumpliera con los requisitos establecidos y fuera efectivo en el entorno real.

La recolección de datos se realizó utilizando varias técnicas para obtener una visión completa del funcionamiento del sistema y su impacto. Se empleó la observación directa para analizar cómo el sistema responde a situaciones simuladas de intrusión y para identificar cualquier problema técnico. Las encuestas permitieron recopilar datos sobre las percepciones y necesidades de los usuarios en cuanto a seguridad domiciliaria. Estas encuestas se enfocaron en obtener información cuantificable sobre la satisfacción de los usuarios y la efectividad del sistema. Además, se llevaron a cabo entrevistas con expertos en seguridad y tecnología, proporcionando una perspectiva más profunda sobre las tendencias actuales y futuras en la seguridad electrónica. Esta combinación de técnicas permitió una evaluación integral del sistema de seguridad y sus efectos.

La metodología de trabajo se basó en un enfoque de desarrollo iterativo, que incluyó fases de diseño, implementación y evaluación. El proceso comenzó con una fase de diagnóstico para identificar las necesidades de seguridad y los requisitos técnicos. Se diseñó el sistema de seguridad, seguido de la creación de prototipos funcionales que fueron sometidos a pruebas rigurosas. Cada ciclo de pruebas permitió realizar ajustes y mejoras en el sistema. Finalmente, se llevó a cabo la implementación del sistema en un entorno real y se realizó una fase de validación para evaluar su efectividad. Los resultados obtenidos fueron analizados para comprobar la hipótesis del proyecto y ajustar el sistema según las necesidades observadas. Este enfoque metodológico garantizó un desarrollo ordenado y enfocado en la solución del problema de investigación, permitiendo la creación de un sistema de seguridad efectivo y adaptado a las necesidades específicas de los usuarios.

Este proceso investigativo metodológico asegura que el sistema de seguridad domiciliario basado en Arduino no solo sea tecnológicamente avanzado, sino también práctico y adaptable a las realidades del entorno doméstico. La combinación de métodos teóricos, prácticos y técnicas de recolección de datos proporciona una base sólida para evaluar y validar el impacto del sistema en la seguridad de los hogares.

Capítulo II: (Marco teórico y conceptual)

2.1. Contextualización general del estado del arte

La implementación de un sistema de seguridad utilizando Arduino surge como respuesta a la necesidad de contar con soluciones de seguridad accesibles y personalizables. En este proyecto, el análisis del problema se centra en identificar las principales vulnerabilidades que existen en los sistemas de seguridad domésticos y comerciales, donde muchas veces se observa una falta de opciones adaptables y de bajo costo. Para abordar esta problemática, la investigación se desarrollará específicamente dentro del ámbito de los sistemas de seguridad electrónica, enfocándose en la integración de sensores y microcontroladores como el Arduino. Los recursos documentales que se emplearán en este estudio incluyen artículos académicos, tesis, y reportes técnicos, seleccionados en función de su relevancia técnica, cronológica, y aplicabilidad, especialmente aquellos publicados en los últimos cinco años. Este enfoque nos permitirá contextualizar adecuadamente la documentación utilizada, siguiendo criterios específicos de búsqueda que aseguren la relevancia y actualidad de los estudios seleccionados. Además, se tendrá en cuenta las tendencias recientes en la investigación, con la intención de aplicar los resultados obtenidos al entorno local y explorar cómo estas innovaciones pueden transferirse y adaptarse a las necesidades reales de seguridad.

Para fundamentar este proyecto, se han revisado conceptos clave relacionados con la seguridad electrónica, la automatización de sistemas mediante microcontroladores, y la conectividad de dispositivos en redes IoT (Internet de las Cosas). Entre los autores más relevantes en esta área, se encuentran Massimo Banzi y Michael Margolis, quienes son pilares en el desarrollo y popularización de la plataforma Arduino, así como Simon Monk, que ha trabajado extensamente en aplicaciones prácticas de Arduino para sistemas de seguridad. Además, se han revisado estudios recientes, como el realizado por (Gutiérrez, 2021), quien implementó un sistema de vigilancia con Arduino en áreas rurales de Bangladesh, utilizando cámaras y sensores de movimiento, obteniendo resultados significativos en la reducción de incidentes de robo. Otro estudio destacado es el de (Portilla, 2021), en el que se desarrolló un sistema de monitoreo de incendios basado en Arduino, demostrando la eficacia de estos sistemas en la detección temprana de incendios en edificios comerciales. Estos trabajos proporcionan un marco sólido para nuestro proyecto, permitiendo aprovechar tecnologías y metodologías que han demostrado ser efectivas en contextos similares.

El análisis de estos estudios previos nos permite orientar nuestra investigación hacia la creación de un sistema de seguridad basado en Arduino que no solo sea accesible y fácil de

instalar, sino que también incorpore tecnologías emergentes como la comunicación IoT para mejorar la eficiencia y capacidad de respuesta del sistema. Al revisar las soluciones existentes, hemos identificado la oportunidad de integrar mejoras en la detección de intrusos y notificaciones en tiempo real, así como la posibilidad de añadir funcionalidades adicionales como el control remoto desde dispositivos móviles. Esta orientación se fundamenta en la clasificación de ideas previas, adoptando lo más relevante de cada investigación, y añadiendo valor mediante la integración de nuevas tecnologías y métodos que no han sido completamente explorados, como la inclusión de sistemas de energía renovable para aumentar la sostenibilidad del sistema de seguridad. Con esta base, nuestro proyecto tiene el potencial de contribuir de manera significativa al campo de la seguridad electrónica, ofreciendo una solución innovadora y adaptable a las necesidades específicas del contexto ecuatoriano.

Capítulo III: (Metodología)

3.1. Diseño y arquitectura del sistema

Para este proyecto, tenemos una arquitectura que conecta diversos componentes de hardware con un microcontrolador **Arduino**. El propósito es crear un sistema de seguridad que sea capaz de detectar intrusos mediante sensores de movimiento y cámaras de seguridad conectadas a un IP, y una alarma que notifique al usuario la invasión de personas no autorizadas y capturar en fotogramas y guardarlos en la nube.

Tabla 1 *Diseño y arquitectura del sistema*

Aspecto	Descripción
Arquitectura del Sistema	
División de Tareas	Identificación y división de tareas lógicas en módulos o componentes.
Comunicación entre Componentes	Establecimiento de mecanismos eficientes para la transferencia de datos entre módulos.
Gestión de Datos	Diseño de un sistema robusto de gestión de datos, considerando almacenamiento local o en la nube.
Diseño del Sistema	
Hardware y Periféricos	Selección cuidadosa de hardware, sensores, cámaras y actuadores para integrarse coherentemente.
Distribución de Tareas	Posible distribución de tareas entre varios dispositivos para cómputo distribuido.

Aspecto	Descripción
Interfaz de Usuario (UI)	Diseño de una interfaz de usuario intuitiva, ya sea web, móvil o GUI.
Seguridad	Implementación de medidas de seguridad como cifrado de datos y autenticación.
Escalabilidad	Diseño para escalabilidad, considerando el crecimiento del sistema y nuevas funcionalidades.
Mantenimiento y Actualizaciones	Estrategia para el mantenimiento y la aplicación de actualizaciones sin interrupciones.
Resiliencia y Tolerancia a Fallos	Integración de mecanismos para garantizar la resiliencia del sistema ante posibles fallos.

3.2. Especificaciones técnicas

Las especificaciones técnicas de un sistema que permite tener características y capacidades de componentes siendo utilizados dentro del sistema, cada uno de los componentes tiene un papel importante desde el control de entrada como de salida el microcontrolador Arduino siendo el principal todos los componentes se basan en tenerlo robusto tanto en rendimiento como en seguridad llevando a si su mejor función dentro del sistema de detección de intrusos.

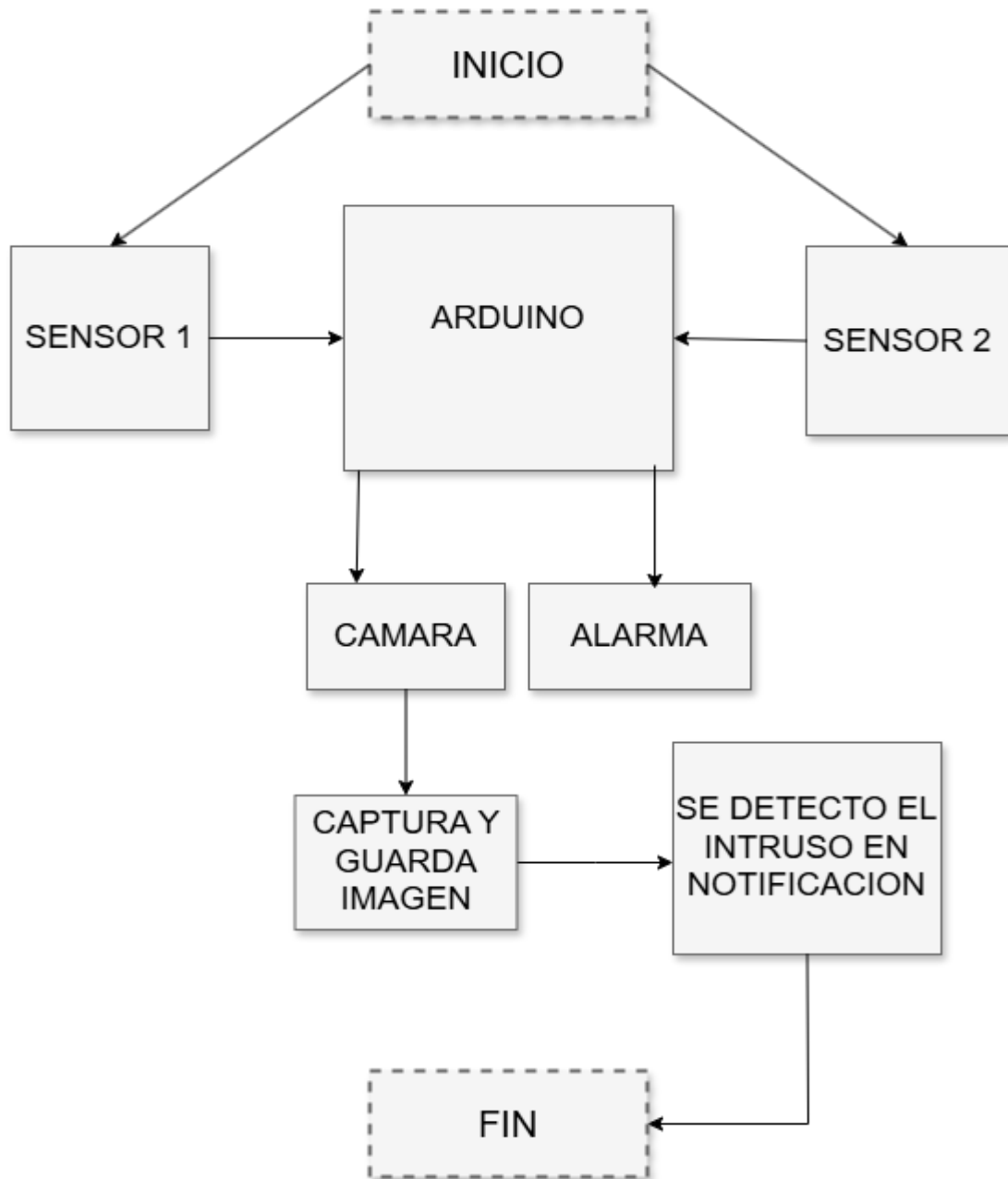
Tabla 2 Especificaciones técnicas

Componente	Rendimiento	Seguridad	Certificaciones
Microcontrolador Arduino	Control de sensores y actuadores con baja latencia.	Protección en el acceso al sistema a través de autenticación.	Certificación CE, RoHS (Arduino UNO).
Sensores de Movimiento (PIR)	Detección rápida de movimiento en tiempo real.	No aplica directamente.	No aplicable.
Cámaras de Seguridad	Captura de imágenes y videos en tiempo real.	Captura imágenes para control de acceso, puede tener encriptación en la transmisión de datos.	Certificación CE, FCC.

Componente	Rendimiento	Seguridad	Certificaciones
Sensores de Puertas/Ventanas	Monitoreo de apertura/cierre de puertas y ventanas.	Monitoreo de aperturas y cierres, protección física de los sensores.	No aplica.
Módulo de Comunicación Wi-Fi (ESP8266/ESP32)	Conexión estable a Wi-Fi para enviar datos de manera continua.	Conexión segura al sistema mediante protocolos Wi-Fi.	Certificación CE, FCC (ESP8266/ESP32).
Sirena de Alarma	Emisión de sonido con alta potencia.	Emite señales acústicas para alertar intrusiones.	Certificación CE.
Pantalla LCD o LED	Muestra información en tiempo real con buena visibilidad.	No aplica.	No aplica.
Controladores de Acceso	Autenticación rápida con métodos como RFID o teclado.	Seguridad en el acceso mediante autenticación (RFID o PIN).	Certificación CE (para módulos RFID).
Módulo RTC (DS3231)	Proporciona hora y fecha exacta para los registros.	Protege la hora y fecha con backups internos.	Certificación RoHS.
Fuente de Alimentación	Proporciona energía continua al sistema.	Sistema protegido contra sobrecarga de energía.	No aplica.

3.3. Diagrama de procesos

Figura 1 Diagrama de procesos



El Sensor PIR establece un campo de monitoreo en el área buscando movimientos. Este al ser detectado se comunica con el módulo principal que es el Arduino.

El desarrollo en este contexto se comunica de manera directa con el Arduino quien controla, el cual tiene una activación inmediata con los periféricos.

Una vez activado el periférico sondea con fotogramas, y empieza a capturar movimientos y objetos que empiecen a tener movimiento.

El proceso concluye con un mensaje tipo alerta, y en la nube se encuentra todos los fotogramas que el periférico obtuvo mediante se realizó el proceso de detección.

El módulo principal obtiene todos los datos relevantes para que pueda tener un funcionamiento adecuado, fue instalado y actualizado para un óptimo desempeño, la programación y la compilación se realiza en lenguaje de Python. dado que este módulo principal Arduino posee un potente procesamiento, y se convierte en un módulo versátil capaz de abordar diversas necesidades del sistema. No obstante, es un módulo que puede tener varios procesos y no tendría problemas en realizar varias funciones.

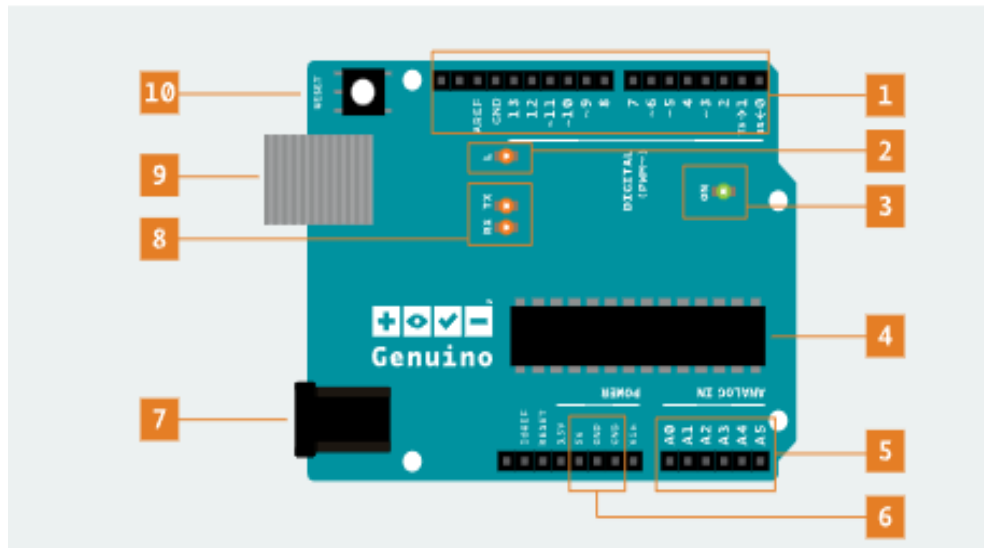
El módulo de entrada y salida está compuesto por componentes de electrónica, ya sea discreta o integrada, con el propósito de adaptar las señales a las características de los puertos de comunicación del microprocesador. En la medida de lo posible, se diseña este módulo de entrada y salida de manera sencilla, con la intención de agrupar la mayor cantidad de funciones en los módulos de control y gestión.

3.4. Configuración de Microcontroladores

3.4.1. Ajustes y configuración del Microcontrolador Arduino

Las placas Arduino detectan el entorno al recibir información de numerosos sensores e influyen en su entorno controlando luces, motores y otros actuadores. Las placas Arduino son la plataforma de desarrollo de microcontroladores que será la base de tus proyectos. Al crear algo, construirás los circuitos e interfaces para la interacción y le indicarás al microcontrolador cómo interactuar con otros componentes. Aquí tienes la anatomía de Arduino UNO. Si la Figura que aparece en la pantalla no llena completamente el monitor o televisor, podemos experimentar desactivando esta opción para ver si se corrige.

Figura 1. Microcontrolador Arduino



1. Pines digitales

Configuración: Estos pines son esenciales para la entrada y salida de señales digitales. Utilizarás `digitalRead()` para leer el estado de los sensores de movimiento (PIR) y de puertas/ventanas, y `digitalWrite()` para controlar actuadores como la sirena de alarma, encender o apagar LEDs, y activar dispositivos como cámaras o el relé para el control de acceso.

Pines PWM (Pulse Width Modulation): Si necesitas controlar dispositivos que requieren señales analógicas, como controlar el brillo de un LED o la velocidad de un motor, usarás `analogWrite()` en estos pines. Esto puede ser útil para hacer que una sirena tenga diferentes niveles de intensidad o para controlar dispositivos como un ventilador si fuera necesario en el sistema.

2. LED del pin 13

Configuración: El LED en el pin 13 está integrado y es muy útil para la depuración. Podrás usarlo para verificar que el programa está corriendo y depurar tu código al mostrar si el sistema está recibiendo señales de los sensores o si el Arduino está respondiendo correctamente. Por ejemplo, cuando detecte movimiento, puedes hacer que el LED parpadee.

```
pinMode (13, OUTPUT); // Configura el pin 13 como salida
digitalWrite (13, HIGH); // Enciende el LED
delay (1000); // Espera 1 segundo
digitalWrite (13, LOW); // Apaga el LED
```

3. El LED de encendido

Configuración: Este LED indica que el Arduino recibe alimentación. No es programable directamente, pero es un buen indicador de que la placa está funcionando y es útil para verificar que el Arduino está alimentado correctamente cuando lo conectas a una fuente externa (fuente de alimentación o batería).

4. Microcontrolador ATmega

Configuración: El microcontrolador ATmega328P es el cerebro de tu Arduino y ejecuta todos los programas que cargues. Asegúrate de que tu código esté optimizado para su capacidad, ya que algunos sensores y componentes pueden requerir más recursos de los disponibles en este microcontrolador.

El ATmega328P controlará los sensores de movimiento, cámaras, y actuadores (sirenas, LEDs, etc.) a medida que tu código se ejecute.

5. Entrada analógica

Configuración: Los pines analógicos permiten leer valores continuos, por ejemplo, para leer valores de sensores de temperatura o luz, si los incluyes en tu proyecto. Sin embargo, en tu configuración actual, probablemente no sea necesario para los sensores PIR o las cámaras, ya que estos usan entradas digitales. Podrías usar estos pines si decides integrar sensores adicionales.

```
int sensorValue = analogRead(A0);    // Lee un valor de un sensor analógico
```

6. Pines GND y 5V

Configuración: Estos pines proporcionan la alimentación para el Arduino y para los componentes externos. Conectarás los pines GND (tierra) a la mayoría de los componentes (sensores PIR, sirenas, etc.) y el pin 5V para alimentar algunos de los módulos, como la pantalla LCD o los sensores de movimiento.

GND: Se conecta a los sensores, sirena, y otros módulos que necesiten tierra.

5V: Alimenta módulos que operen con 5V, como el ESP8266/ESP32 para comunicación Wi-Fi o el sensor PIR.

7. Conector de alimentación.

Configuración: Si no estás usando el puerto USB, puedes alimentar el Arduino usando una fuente externa. Es importante que la fuente tenga un voltaje de entre 7V y 12V para garantizar el funcionamiento estable del Arduino. Esto puede ser útil si estás usando baterías o una fuente de alimentación externa para tu proyecto.

Utiliza este conector si desea alimentar tu sistema de manera autónoma (sin depender de la computadora para la carga del código).

8. LED TX y RX

Configuración: Los LED TX (transmisión) y RX (recepción) indican la comunicación entre el Arduino y el ordenador. Si tu proyecto involucra comunicación en serie (por ejemplo, para transmitir datos entre el Arduino y una aplicación o un sistema de monitoreo), estos LED te ayudarán a verificar si los datos están siendo enviados y recibidos correctamente.

```
Serial.begin(9600);      // Inicia la comunicación serie a 9600 baudios
Serial.println("Sistema activo");  // Envía un mensaje al puerto serial
```

9. Puerto USB

Configuración: El puerto USB permite cargar programas en el Arduino y alimentar el sistema cuando está conectado al ordenador. Para tu proyecto, lo usarás para cargar el código y, posiblemente, para depurar a través de la comunicación serie.

Este puerto será útil para la comunicación entre el Arduino y otros dispositivos o para monitorear los resultados a través de la consola serie.

10. Botón de reinicio

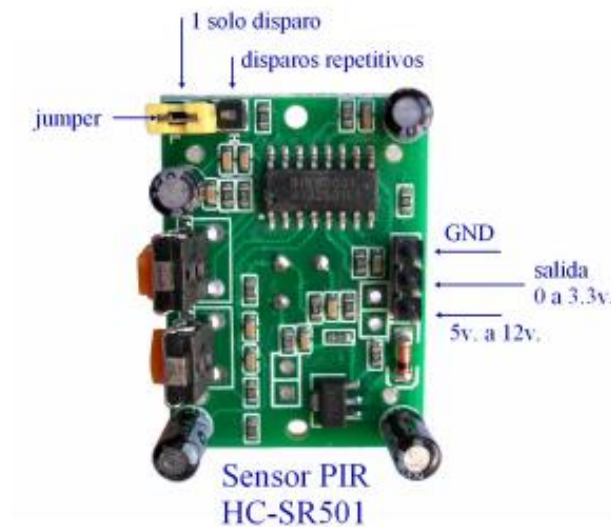
Configuración: El botón de reinicio permite reiniciar el microcontrolador ATmega328P. Si hay un error en el programa o si necesitas reiniciar el sistema, puedes utilizar este botón para reiniciar la ejecución del código sin tener que desconectar y reconectar la alimentación.

Útil si necesitas reiniciar el sistema tras una detección de error o cuando desees reiniciar todo el sistema después de una interrupción.

3.4.2. Ajustes y configuración del Sensor infrarrojo pasivo HC-SR501

En los sensores de movimiento, el sensor PIR consta en realidad de 2 elementos detectores separados, siendo la señal diferencial entre ambos la que permite activar la alarma de movimiento. En el caso del HC-SR501, la señal generada por el sensor ingresa al circuito integrado BISS0001, el cual contiene amplificadores operacionales e interfaces electrónicas adicionales.

Figura 2. Sensor infrarrojo pasivo HC-SR501

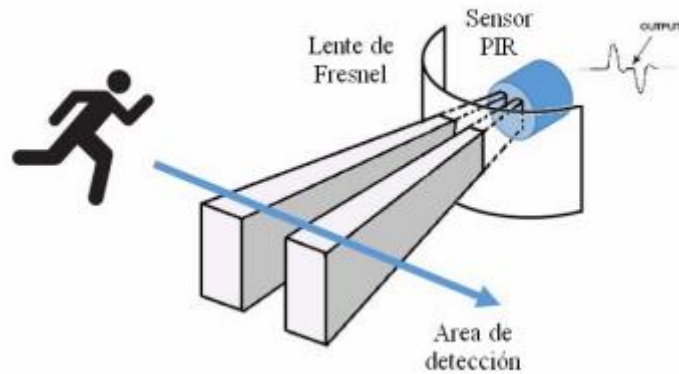


Las funciones y ajustes complementarios del sensor de movimiento son:

1.- **Ajuste de parámetros:** mediante 2 potenciómetros, el usuario puede modificar tanto la sensibilidad como la distancia de detección del PIR.

2.- **Detección automática de luz** (esta función no está disponible al adquirir el sensor de fábrica): por medio de una foto resistencia CdS (Sulfuro de Cadmio), se deshabilita la operación del sensor en caso de que exista suficiente luz visible en el área. Esta función es utilizada en caso de sensores que enciendan lámparas en lugares poco iluminados durante la noche, y especialmente en corredores ó escaleras.

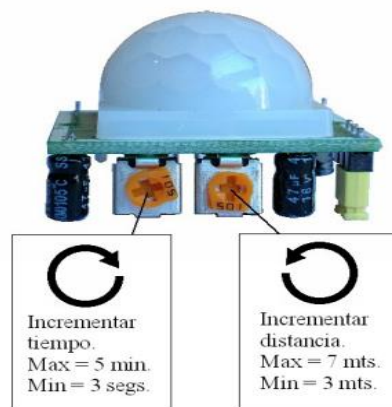
Figura 3. Funcionamiento sensor



3.- El rango de detección de movimiento de los PIR es ajustable y generalmente funcionan con alcances de hasta 7 metros, y con aperturas de 90° a 110°, como se muestra en la figura. El montaje del PIR puede realizarse tanto en piso, muro ó techo, según convenga a la aplicación.

4.- El usuario puede ajustar tanto el tiempo de disparo de la señal de alarma de movimiento, como la distancia de detección. Los potenciómetros correspondientes deben girarse en la dirección mostrada para realizar los ajustes.

Figura 4. Opciones de sensor



El modo de operación a trabajar es:

Disparos repetitivos: en este modo, cada evento detectado genera un nuevo tiempo de activación. Volviendo al ejemplo de tiempo de 60 segundos. Cuando ocurre el primer evento, la

salida se activa. Si transcurridos 30 segundos ocurre un segundo evento, entonces se sumarán 60 segundos al tiempo transcurrido, dando un total de 90 segundos continuos con la salida activa. Y así, cada evento adicional, sumará un tiempo de 60 segundos de activación al tiempo ya transcurrido.

Tiempo de retardo (Delay): Ajusta cuánto tiempo se mantiene activa la salida después de detectar movimiento. Esto es útil si deseas que el sensor mantenga la señal activada durante un cierto período de tiempo después de detectar movimiento.

- En cualquier caso, si la salida regresa a su estado inactivo, habrá un lapso de 3 segundos durante los cuales los nuevos eventos no serán considerados. Pasados esos 3 segundos, el dispositivo regresa a su funcionamiento normal.

3.4.3. Ajustes y conexión de Cámaras de Seguridad Nexxt Solutions

La integración de cámaras de seguridad en el sistema de detección de intrusos es fundamental para monitorear el área y obtener evidencia visual en caso de que se detecte movimiento o apertura de puertas/ventanas. Las cámaras Nexxt Solutions AHIMPFI4U2 V3 proporcionan una solución eficaz para la grabación de video en alta resolución. Este apartado detalla cómo integrar estas cámaras en el sistema utilizando Arduino como plataforma central de control.

Figura 5. Cámaras de Seguridad Nexxt Solutions



La cámara Nexxt Solutions **AHIMPFI4U2 V3** es una cámara de seguridad tipo IP (protocolo de Internet) que se conecta a la red local para enviar y recibir datos. Algunas de sus especificaciones son las siguientes:

Tipo de cámara:

Las cámaras IP son cámaras de seguridad digitales que se conectan a través de redes locales o a Internet. Son ideales para monitoreo en tiempo real, ya que ofrecen transmisión de video en vivo a través de una red, lo que permite acceso remoto desde cualquier lugar con conexión a Internet. Las cámaras IP son fáciles de integrar con sistemas de seguridad basados en Arduino mediante módulos Wi-Fi como el **ESP8266** o **ESP32**, lo que permite una conexión remota eficiente.

Resolución:

Con **4** megapíxeles (MP) de resolución, la cámara proporciona imágenes claras y detalladas, lo que es esencial para **identificación facial** o la **visualización detallada** de áreas sensibles. En la mayoría de las aplicaciones de seguridad, una resolución de 4MP es ideal para monitorear tanto áreas amplias como específicas, asegurando que se capture suficiente detalle incluso en condiciones de poca luz.

Visión nocturna:

Es una característica esencial para grabar en condiciones de poca luz. La cámara cuenta con **LEDs infrarrojos (IR)** que permiten capturar imágenes claras en la oscuridad, lo que asegura que el sistema de seguridad funcione las 24 horas del día, incluso en entornos con poca o ninguna luz ambiental.

Conexión:

La cámara se conecta a la red local a través de Wi-Fi o Ethernet.

1.- Wi-Fi: Ideal para instalaciones donde no es posible cablear una red Ethernet. La conexión Wi-Fi proporciona flexibilidad y facilita la instalación en áreas sin puertos de red cableados.

2.- Ethernet: Proporciona una conexión más estable y rápida, ideal para cámaras que requieren alta calidad de transmisión y grandes anchos de banda (por ejemplo, grabación en tiempo real de video en alta resolución).

Ángulo de visión:

El ángulo de visión de la cámara es crucial para determinar el área cubierta. Las cámaras con un ángulo de visión de 90° o más son adecuadas para cubrir áreas amplias, como entradas o pasillos. Para áreas más específicas o detalladas (como una puerta), una cámara con un ángulo de visión más estrecho puede ser más adecuada.

Funciones adicionales:

1.- Grabación en la nube: Muchas cámaras IP ofrecen grabación en la nube, lo que permite almacenar las grabaciones sin necesidad de un disco duro local. Esta función es útil para tener copias de seguridad de las grabaciones y acceder a ellas desde cualquier lugar.

2. Detección de movimiento: Esta característica es clave para activaciones automáticas. Cuando la cámara detecta movimiento en su campo de visión, puede iniciar la grabación o enviar una alerta al usuario.

3.- Audio bidireccional: Algunas cámaras permiten grabar audio y comunicarte con personas en el área vigilada, lo cual puede ser útil para aplicaciones interactivas o como un elemento disuasorio.

Control remoto:

El acceso remoto a las imágenes es una de las mayores ventajas de las cámaras IP. Con la cámara Nexxt Solutions AHIMPF14U2 V3, puedes acceder a las imágenes a través de una **aplicación móvil** o un **servidor web**.

1.- Aplicación móvil: Muchas cámaras IP ofrecen aplicaciones móviles que permiten acceder a la transmisión en vivo y controlar las cámaras de manera remota.

2.- Servidor web: Si se configura correctamente, puedes acceder a las imágenes a través de un servidor web, lo que te permitirá visualizar las grabaciones desde cualquier computadora o dispositivo conectado a la red.

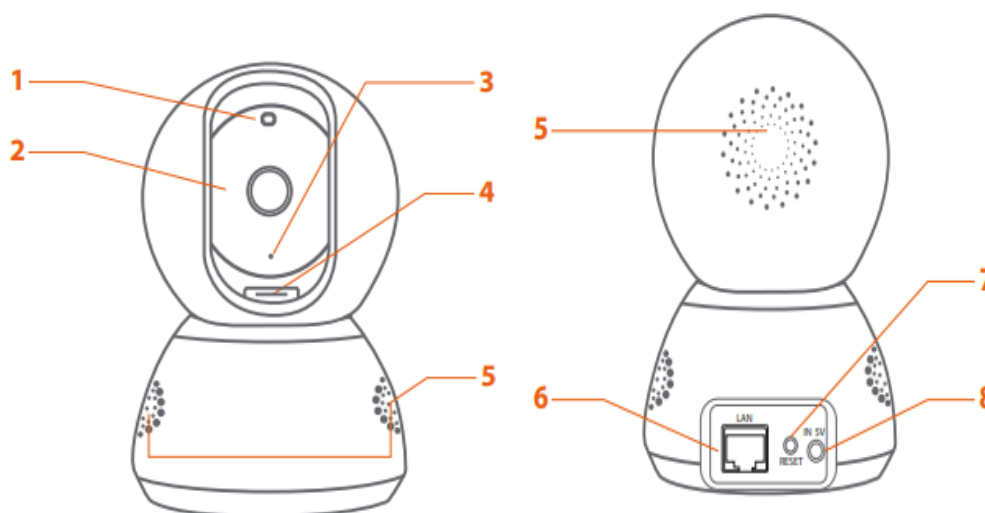
Figura 6. Especificaciones técnicas cámaras

Especificaciones técnicas	
MPN	AHIMPF14U2
Aspectos generales	
Sensor de imagen	CMOS de 1/2,7 pulgada
Tipo de lente	Fijo de 3,6mm
Ángulo de visión	80 grados
Ángulo de inclinación	105 grados
Ángulo de rotación	Horizontal: 335 grados Vertical: 75 grados
Compresión de audio	G.711U/AAC
Audio bidireccional	Micrófono y parlante integrados
Botón de reposición	Si
Frecuencia de imagen	15fps
Modo infrarrojo	Conmutación de doble filtro
Resolución	2K QHD (2304x1296p) 3MP
Captura de imagen estática	Si
Visión nocturna	Hasta 9m
Grabación de video	Tarjeta microSD™ de hasta 128GB
Compresión de video	H.264
Frecuencia inalámbrica	IEEE 802.11N, 2.4GHz (no es compatible con redes Wi-Fi de 5GHz)
Alimentación	
Tensión	100-240VCA a 50/60Hz, 0,25A
Adaptador de corriente	5VCC, 1000mA
Potencia máxima	5W
Características físicas	
Dimensiones	8,1x7,9x11,7cm
Peso	0,2kg
Longitud del cable	1,5m
Condiciones ambientales	
Temperatura de funcionamiento	10°C ~ 60°C
Temperatura de almacenamiento	200°C ~ 70°C
Humedad relativa	10% ~ 90%
Información adicional	
Requisitos del sistema	- Dispositivo móvil con plataforma iOS8 o superior, Android™ 4.1 o superior - Aplicación Nexxt Home - Red Wi-Fi existente
Garantía	Dos años

Configuración: Configura la cámara para grabar en su máxima resolución y ajusta las configuraciones de compresión de video (por ejemplo, H.264 o H.265) para optimizar el almacenamiento sin perder calidad visual.

Ajuste de visibilidad nocturna: Verifica que la función de visión nocturna esté activada en la configuración de la cámara. Asegúrate de que los LEDs infrarrojos no estén obstruidos, ya que pueden reducir la efectividad de la grabación nocturna.

Figura 7. Características físicas cámaras



Descripción:

1. LED infrarrojo: para visión nocturna
2. Lente
3. Micrófono: capta su voz o cualquier otro sonido alrededor de la cámara
4. Ranura para tarjeta microSD™: admite el almacenamiento local a través de una tarjeta microSD de hasta 128GB
5. Parlante: transmite el audio captado por el teléfono a través de la aplicación Nexxt Home
6. Puerto LAN
7. Botón de reposición: mantenga presionado este botón para reiniciar el dispositivo.
8. Conjunto de alimentación: CC5V $\pm$ 10%

Recomendación para la conexión:

Si la señal de **Wi-Fi** es débil o intermitente, opta por la **conexión Ethernet** para asegurar una transmisión estable.

Si se usa Wi-Fi, asegúrate de configurar la cámara para conectarse a la red mediante una **dirección IP estática** para facilitar la conexión remota.

3.4.4. Conexión y Configuración de Sensores de Puertas/Ventanas

Los **sensores de puertas/ventanas** son cruciales para el monitoreo de aperturas y cierres en puntos vulnerables. Estos sensores utilizan un **Reed switch** que detecta la apertura o cierre de puertas o ventanas mediante la variación de la proximidad de un imán.

El sensor **magnético** de puertas/ventanas está compuesto por dos partes:

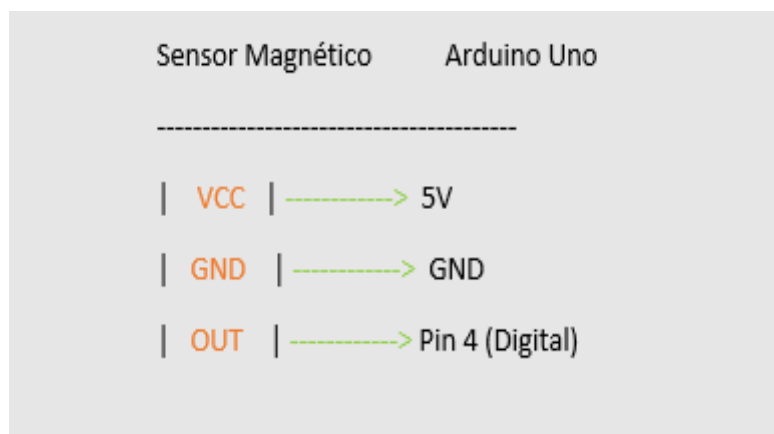
1.- Parte fija: Instalado en el marco de la puerta o ventana.

2.- Parte móvil: Instalado en la puerta o ventana misma.

Alineación del Sensor: El sensor debe estar correctamente alineado con el imán. Si no están alineados adecuadamente, el sensor puede no detectar correctamente la apertura o cierre de la puerta/ventana.

Sensor de detección magnética: cuando la puerta o ventana está cerrada, el imán de la parte móvil mantiene el circuito cerrado, y cuando se abre la puerta o ventana, el imán se aleja, lo que abre el circuito y genera una señal.

Estos sensores magnéticos no requieren ajustes específicos, solo asegurarse de que el imán esté bien alineado con el sensor para así poder ejecutar su acción y cumpla con el requerimiento.



Pin VCC:

El **pin VCC** este trabaja en (**Alimentación**) este tipo de sensor debe ser conectado al **pin de 5V** en la placa de **Arduino**. Una vez conectado suministra la energía al sensor. El pin de 5v que alimenta el sensor de forma adecuada para su funcionamiento. Pueden funcionar con 3.3v, pero para sensores comunes como son el **HC-SR501**, se usa **5V**.

Pin GND:

El **pin GND** este trabaja en (Tierra), el sensor debe conectarse al pin **GND** en la placa de **Arduino**. La referencia común para los circuitos eléctricos y permite completar el circuito de alimentación entre el sensor y el Microcontrolador.

Este pin es crucial para que el sensor y el Arduino tengan una base de referencia común y el sensor funcione correctamente.

Pin OUT:

El **pin OUT** este trabaja en (Salida), el sensor que proporciona el estado de detección (abierto o cerrado). Este pin debe conectarse a un **pin digital** de Arduino, como el **pin 4** (o cualquier otro pin digital disponible). Cambiará de estado (de **LOW** a **HIGH**) dependiendo de si el sensor detecta la presencia o ausencia del imán.

1.- Cuando la puerta o ventana está **cerrada** (el imán está cerca del sensor), la salida será **LOW** (sin señal).

2.- Cuando la puerta o ventana se **abre** (el imán se aleja), la salida será **HIGH** (se activa la señal).

Protección del Sensor: Para evitar daños en el sensor, especialmente si se encuentra en áreas expuestas, se recomienda protegerlo físicamente con una caja resistente o montarlo en lugares donde esté menos expuesto a impactos o cambios bruscos de temperatura.

3.4.5. Ajustes y configuración Modulo de Comunicación WI-FI (ESP8266/ESP32)

El módulo WI-FI permite una comunicación remota del sistema de detección de intrusos, permitiendo que el sistema se conecte a la red Wi-Fi y envíe notificaciones, imágenes o alertas a un servidor, teléfono móvil o correo electrónico.

El ESP8266 es un SoC (System on Chip), con capacidades de 2.4 GHz Wi-Fi (802.11 b / g / n, soporte WPA / WPA2), 16 GPIO de propósito general (entrada / salida), I²C, convertidor analógico-digital (ADC de 10 bits), SPI, I²S, UART y modulación de ancho de pulso (PWM), emplea un CPU RISC de 32 bits basado en el Tensilica Xtensa LX106 funcionando a 80 MHz (o overclocked a 160 MHz). Tiene una memoria ROM de inicio de 64 KB, memoria RAM de instrucciones de 64 KB y 96 KB de RAM de datos. Memoria flash externa de 4MB, pero este último varía entre diferentes versiones de modulo.

Figura 8. *Módulo de Comunicación WI-FI (ESP8266/ESP32)*



ESP8266 es un microcontrolador con WIFI incorporado. Y al igual que muchos microcontroladores podemos reprogramar a nuestra necesidad, por defecto o de fábrica nuestro ESP8266 viene con un firmware de comandos AT, que nos permite manipular el ESP8266 desde otro microcontrolador, nosotros en este tutorial vamos a programar el ESP8266, pero usando el IDE Arduino.

Para la correcta conexión, la placa ESP8266 utiliza un nivel lógico de 3.3V, se conecta todo correctamente para evitar dañar el módulo.

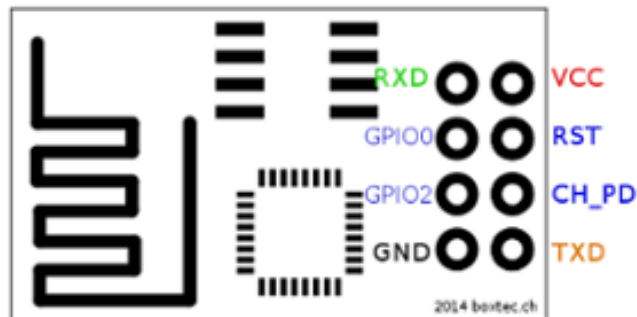
Figura 9. Guía de pines del ESP8266

ESP8266	UARTbee	Arduino Uno
RXD	TXD	RX
GPIO0	GND	GND
GPIO2	-----	
GND	GND	GND
VCC	3.3V	3.3V
RST	-----	
CH_PD	3.3V	3.3V
TXD	RXD	TX

Conecta el **ESP8266** o **ESP32** al Arduino utilizando los pines **TX** y **RX** para la comunicación serie. Si el ESP es autónomo, solo conéctalo a una fuente de alimentación de **3.3V** (si usas ESP8266) o **5V** (si usas ESP32).

Configuración de red: Configura la conexión Wi-Fi utilizando las credenciales de la red (SSID y contraseña) en el código de Arduino.

Figura 10. Configuración red



VCC: Conectar a la fuente de alimentación de 3.3V

GND: Conectar a tierra

TX: Conectar al RX del UART o Arduino

RX: Conectar al TX del UART o Arduino (usar un divisor de voltaje para reducir de 5V a 3.3V si se usa Arduino)

GPIO0: Conectar a tierra (pone al ESP8266 en modo de flas)

3.4.6. Conexión y Configuración de Sirena de Alarma

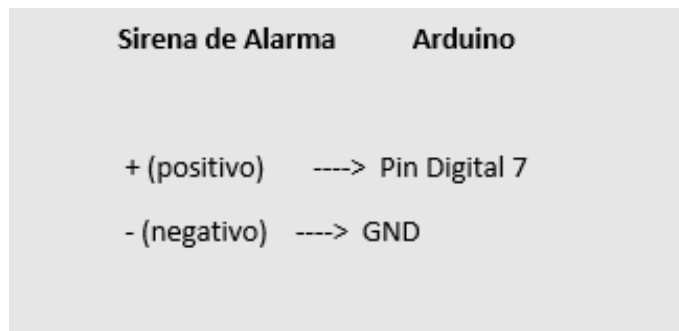
La **sirena de** alarma es un dispositivo de salida que se activa cuando el sistema detecta una intrusión o una apertura no autorizada de puertas/ventanas. Su principal función es emitir un sonido fuerte para alertar a las personas sobre la presencia de un intruso, desalentando el acceso no autorizado.

Pin Positivo:

- 1.- El pin **positivo** de la sirena debe ser conectado a un **pin digital** de Arduino, como el **pin 7**.
- 2.- Este pin se utilizará para enviar una señal **HIGH** que activará la sirena de alarma.

Pin Negativo:

- 1.-El pin **negativo** de la sirena debe ir al pin **GND** de Arduino. Esto completa el circuito y permite que la sirena reciba la señal de control.

**Voltaje:**

Antes de conectar la sirena, asegúrate de que la fuente de alimentación (usualmente 5V o 12V) sea compatible con la sirena. Las sirenas más comunes requieren 12V, pero algunas pequeñas sirenas pueden funcionar con 5V. Verifica las especificaciones de tu sirena.

Corriente:

Si la sirena requiere un alto consumo de corriente, es recomendable utilizar un transistor o un relé para manejar el flujo de corriente, ya que los pines digitales de Arduino no pueden suministrar suficiente corriente para activar una sirena de gran potencia.

Control del Volumen:

Si la sirena permite ajustar el volumen, configura la sirena a un nivel que sea suficientemente audible en el área de monitoreo, pero sin causar molestias innecesarias.

Duración de la Alarma:

En el código de Arduino, puedes modificar la duración de la alarma ajustando el delay () que mantiene la sirena encendida. Para evitar que la alarma suene por un tiempo demasiado largo, establece un valor adecuado (como 5 o 10 segundos).

Desactivación Automática:

En lugar de dejar la sirena encendida indefinidamente, puedes programar una función que apague la sirena después de un tiempo determinado. Además, se puede configurar un botón de "reset" que permita apagar la sirena manualmente después de una falsa alarma o una vez que se confirme que la intrusión ha sido gestionada.

3.4.7. Conexión y Configuración de la Pantalla LCD o LED

La Pantalla LCD 16x2 es un componente comúnmente utilizado para mostrar información en tiempo real sobre el estado de un sistema, como el estado del sistema de seguridad o si se ha detectado movimiento. Este tipo de pantalla es ideal para mostrar mensajes simples y es fácil de integrar con Arduino para proyectos de monitoreo de seguridad.

La pantalla LCD o LED puede usarse para visualizar:

- 1.- El estado del sistema (activado o desactivado).
- 2.- Los mensajes de alerta (por ejemplo, "Movimiento detectado").
- 3.- Información adicional como la hora y fecha del evento (utilizando el módulo RTC DS3231).

Conexión con I2C:

Pantalla LCD (I2C)	Arduino Uno
VCC	-----> 5V
GND	-----> GND
SDA	-----> A4 (SDA)
SCL	-----> A5 (SCL)

La pantalla LCD con I2C tiene dos pines de comunicación, SDA (Serial Data Line) y SCL (Serial Clock Line), que debes conectar a los pines A4 (SDA) y A5 (SCL) de un Arduino Uno (o los pines correspondientes en otras placas de Arduino, por ejemplo, en el ESP8266).

Conectar VCC y GND: Conecta el pin VCC de la pantalla al pin 5V de Arduino y el pin GND al pin GND de Arduino.

3.4.8. Conexión y Configuración de Controladores de Acceso

Para verificar la identidad de las personas que intentan ingresar al área protegida. Dependiendo del tipo de tecnología utilizada, pueden incluir **lectores RFID**, **teclados matriciales** o **sensores biométricos**.

Configuración:

Un lector **RFID** (Identificación por Radiofrecuencia) es un dispositivo que lee las tarjetas o llaveros RFID para verificar la identidad de una persona. El lector suele comunicarse con **Arduino** utilizando el protocolo **SPI** (Serial Peripheral Interface).

Conexión:

Lector RFID	Arduino Uno
<i>SDA</i>	---> <i>Pin SDA (D2)</i>
<i>SCL</i>	---> <i>Pin SCL (D3)</i>
<i>MISO</i>	---> <i>Pin MISO (D4)</i>
<i>MOSI</i>	---> <i>Pin MOSI (D5)</i>
<i>SCK</i>	---> <i>Pin SCK (D6)</i>
<i>RST</i>	---> <i>Pin 9</i>
<i>3.3V</i>	---> <i>3.3V (algunos lectores funcionan con 3.3V)</i>
<i>GND</i>	---> <i>GND</i>

El lector RFID se conecta al Arduino usando los pines SDA, SCL, MISO, MOSI y SCK.

1.- SDA (Serial Data Line) se conecta al pin SDA de Arduino.

2.- SCL (Serial Clock Line) se conecta al pin SCL de Arduino.

3.- MISO (Master in Slave Out), MOSI (Master Out Slave In) y SCK (Serial Clock) se conectan a los pines SPI correspondientes de Arduino.

4.- GND y VCC se conectan a GND y 5V de Arduino, respectivamente.

3.4.9. Conexión y Configuración de la Fuente de alimentación

El sistema de detección de intrusos debe estar alimentado por una fuente de alimentación estable que proporcione suficiente energía para todos los componentes del sistema.

Configuración:

Utiliza una fuente de 5V para alimentar el Arduino y los componentes que funcionan a esta tensión.

Si estás utilizando otros dispositivos que requieren 12V (como algunas sirenas), puedes usar una fuente de 12V y un regulador de voltaje para proporcionar la energía necesaria.

Es importante tener en cuenta el consumo de energía de cada componente. Asegúrate de que la fuente pueda manejar el consumo total de todos los dispositivos conectados.

Conexión:

Fuente de Alimentación	Arduino
12V (o 5V)	---> Vin (o 5V)
GND	---> GND

Para un sistema estable, hay que asegurar de que la fuente de 5V sea capaz de alimentar adecuadamente el Arduino y los sensores.

Si se usa dispositivos de 12V como la sirena, un regulador de voltaje o un relé es necesario para convertir los 12V a 5V para Arduino.

Capítulo IV: Implementación

La propuesta de implementación de un sistema de seguridad domiciliario utilizando Arduino se fundamenta en una integración de conceptos teóricos y tecnológicos, incluyendo electrónica, automatización, y comunicación inalámbrica. Cada una de estas disciplinas juega un papel crucial en el desarrollo del sistema, contribuyendo a su funcionalidad y eficacia.

Electrónica es la base sobre la cual se construye el sistema de seguridad. Esta área proporciona los conocimientos necesarios para el diseño e integración de componentes electrónicos esenciales, como sensores y actuadores. Según Hagedorn et al. (2019), los sensores permiten la detección precisa de eventos, como la apertura de una puerta o el movimiento en una habitación. Los actuadores, por su parte, ejecutan las acciones correspondientes, como encender luces o activar alarmas. La combinación de estos componentes permite al sistema responder de manera efectiva a las condiciones del entorno, garantizando una respuesta oportuna ante posibles amenazas.

La automatización es otro aspecto clave que complementa la electrónica. Este concepto se refiere a la capacidad de programar el sistema para que opere de manera autónoma y eficiente. Al utilizar el entorno de desarrollo Arduino, se puede escribir código que define cómo deben reaccionar los componentes del sistema a diferentes señales y eventos. Según EASA (2021), la automatización reduce la necesidad de intervención humana constante, minimizando errores y mejorando la precisión del sistema. Este enfoque permite que el sistema de seguridad funcione de manera continua y confiable, ajustando sus respuestas de acuerdo con los eventos detectados.

El Internet de las Cosas (IoT) juega un papel importante en la conectividad del sistema. Esta tecnología permite que los dispositivos se comuniquen y se controlen de manera remota a través de una red WiFi. El uso del módulo ESP32-CAM, como se menciona en la investigación de EASA (2023), permite que el sistema no solo recopile datos y ejecute acciones, sino que también transmita información en tiempo real. Esto proporciona a los usuarios la capacidad de monitorear y controlar el sistema desde cualquier lugar, mejorando la flexibilidad y la accesibilidad del sistema de seguridad.

La programación en el IDE de Arduino es esencial para la implementación del sistema, este entorno de desarrollo permite la configuración detallada del sistema, definiendo cómo se procesan los datos de los sensores y cómo se activan los actuadores. De acuerdo con Hagedorn et al. (2019), una programación adecuada asegura que todos los componentes del sistema trabajen en conjunto de manera armoniosa. Esta etapa es fundamental para la funcionalidad

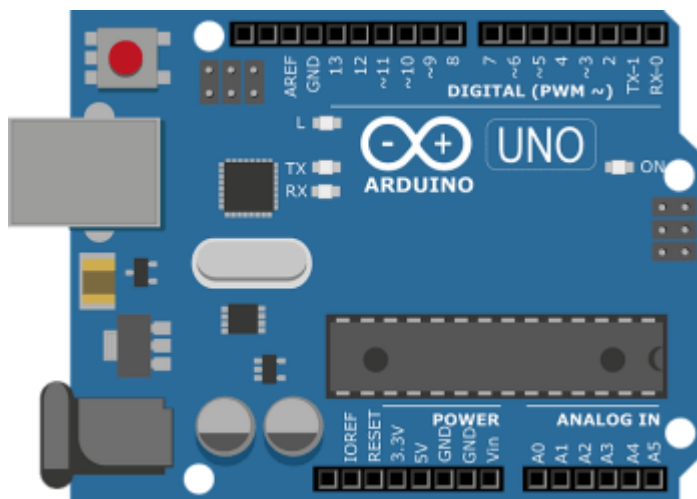
del sistema, ya que permite personalizar y ajustar el comportamiento del sistema según las necesidades específicas del usuario.

Aunque el proyecto no utiliza tecnologías avanzadas como la visión por computador, el módulo ESP32-CAM ofrece capacidades básicas de captura de imágenes. Esta funcionalidad adicional, como se destaca en la investigación de EASA (2023), proporciona una capa extra de seguridad al permitir la visualización remota del entorno, ofreciendo a los usuarios la posibilidad de verificar la situación en su hogar desde cualquier ubicación.

4.1. Arduino

Arduino es una plataforma de hardware de código abierto que se caracteriza por su tarjeta electrónica sencilla, la cual incluye una placa con puertos de entrada y salida, tanto analógicos como digitales. En la figura 1 se muestra a Arduino, que opera en un entorno de desarrollo que emplea un lenguaje de programación propio, lo que facilita su uso en diversos proyectos. El microcontrolador Atmega328, fabricado por ATMEL, actúa como el componente central de Arduino, siendo un circuito integrado de bajo costo que permite la creación de una amplia variedad de diseños (González, 2014).

Figura 11. *Arduino*



Nota: Arduino es una plataforma de hardware abierta con un microcontrolador que permite crear proyectos electrónicos de bajo costo. Fuente: (Chico, 2022)

La estructura principal de Arduino se compone de una placa que alberga los controladores necesarios para gestionar otros componentes y circuitos conectados. Este dispositivo requiere un lenguaje de programación para ser configurado según las necesidades del usuario, lo que lo convierte en una herramienta integral tanto para la creación como para el aprendizaje en el

diseño de sistemas electrónicos automatizados. Su accesibilidad y versatilidad hacen que Arduino sea una opción popular para quienes buscan desarrollar proyectos tecnológicos de manera eficiente y efectiva (Martínez, 2017).

Funciones y uso

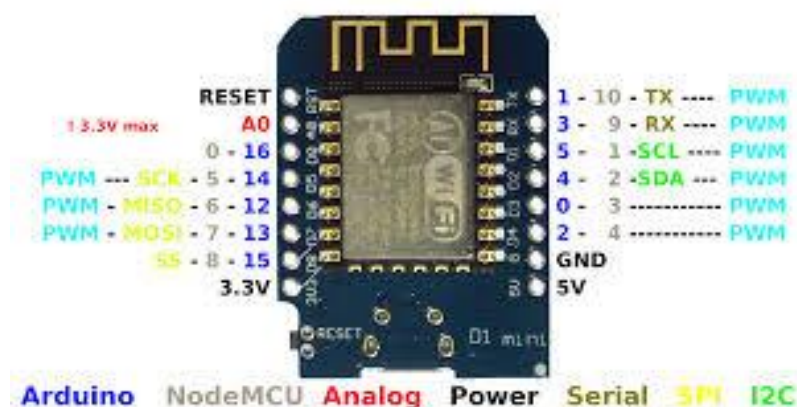
El funcionamiento de Arduino en la seguridad del hogar se basa en la combinación de sensores, actuadores y módulos de comunicación, creando un sistema automatizado que supervisa y reacciona ante posibles amenazas o emergencias (Salvador, 2022). Este sistema emplea sensores magnéticos de contacto para detectar aperturas no autorizadas en puertas y ventanas, y sensores de movimiento PIR para vigilar el interior de la casa. Si se identifica una irregularidad, el Arduino puede activar una alarma y enviar alertas al propietario mediante un módulo Wi-Fi, como el ESP8266 o ESP32.

El control de acceso se gestiona a través de un módulo RFID o un teclado matricial, permitiendo que los usuarios se autenticuen para ingresar a la vivienda. Una pantalla LCD u OLED muestra el estado del sistema, mientras que un módulo RTC registra eventos importantes. La alimentación del sistema es respaldada por una batería externa, asegurando su funcionamiento continuo incluso durante cortes de energía. Este enfoque convierte a Arduino en el núcleo de un sistema de seguridad doméstico, capaz de operar de manera eficiente y confiable en la supervisión y el control remoto de la seguridad del hogar.

4.1.1. Entradas y salidas

Los puertos de entrada y salida del Arduino se programan de manera autónoma; en la figura 2 se muestra cómo la transmisión de la información o la ejecución de las instrucciones se lleva a cabo directamente a través de la interfaz gráfica utilizada para la programación del dispositivo Arduino, lo que requiere la identificación de los componentes relevantes.

Figura 12. Entradas y salidas



Nota: La figura muestra un módulo NodeMCU, un microcontrolador basado en ESP8266 con pines de entrada y salida. Fuente: (Arce, 2023)

En el desarrollo del sistema domótico y de entretenimiento, se empleó la versión Arduino UNO, la cual cuenta con 14 pines digitales y 6 pines analógicos. Los pines analógicos se utilizan exclusivamente como entradas, mientras que los pines digitales pueden configurarse tanto como entradas como salidas. Además, estos pines digitales pueden ser ajustados para funcionar en modo de Modulación por Ancho de Pulso (PWM).

El Arduino UNO también dispone de un bus serie que facilita la comunicación con el dispositivo. Este bus serie está conectado al puerto USB, permitiendo que el Arduino se conecte directamente a la computadora mediante un cable USB

4.1.2. Electrónica y Automatización

La electrónica es fundamental para la construcción de cualquier sistema de seguridad basado en hardware. En este contexto, el uso de componentes como sensores, actuadores y módulos de comunicación es crucial. La automatización, por su parte, permite la implementación de sistemas que pueden operar de manera autónoma o con mínima intervención humana. La plataforma Arduino, al ser una herramienta de código abierto, facilita la integración de estos componentes y la programación del sistema para que realice tareas específicas, como la activación de alarmas o el control de dispositivos a distancia.

4.1.3. Internet de las Cosas (IoT)

El concepto de IoT se refiere a la interconexión de dispositivos a través de internet, permitiendo la comunicación y el intercambio de datos entre ellos. En el caso del sistema de seguridad, el uso de IoT permite que el sistema sea accesible desde cualquier ubicación, proporcionando un control remoto sobre la seguridad del hogar. El módulo ESP32-CAM, utilizado en el proyecto, es un ejemplo de dispositivo IoT que combina capacidades de cámara con conectividad WiFi, facilitando la videovigilancia y el acceso a imágenes en tiempo real desde cualquier parte del mundo.

4.1.4. Comunicaciones inalámbricas

Las comunicaciones inalámbricas son esenciales para el funcionamiento del sistema de seguridad, permitiendo la transmisión de datos entre el sistema de monitoreo y el usuario final sin necesidad de cables físicos. El módulo ESP32-CAM, por ejemplo, utiliza WiFi para enviar imágenes y datos a través de internet, lo que permite al usuario verificar la seguridad de su hogar desde un teléfono inteligente u otro dispositivo conectado.

4.1.5. Programación y desarrollo de software

La programación es el núcleo del sistema de seguridad, ya que permite la configuración y el control de los dispositivos conectados. El entorno de desarrollo Arduino IDE proporciona las herramientas necesarias para escribir y cargar el código en la placa Arduino y otros módulos. Este software es compatible con diversos lenguajes de programación y librerías que facilitan la integración de hardware y software, permitiendo la creación de aplicaciones personalizadas para el sistema de seguridad.

4.2. Descripción del sistema

El sistema de seguridad que se propone implementar contará con una estructura robusta y componentes de alta tecnología que cumplen con los requisitos del proyecto. A continuación, se enumeran los dispositivos utilizados para el desarrollo de este sistema como se muestra en la tabla 1:

Tabla 3 Descripción de la propuesta

DISPOSITIVO	MODELO	FUNCION
Microcontrolador Arduino	Arduino Uno (o similar)	Actúa como el cerebro del sistema, controlando y gestionando las entradas y salidas de los sensores y actuadores conectados. Programa las instrucciones para la toma de decisiones en tiempo real.
Sensores de Movimiento (PIR)	Sensor PIR HC-SR501	Detecta la presencia de personas en áreas monitoreadas, activando alarmas o enviando notificaciones si se detecta movimiento no autorizado.
Cámaras de Seguridad	Cámara IP (con soporte para OpenCV)	Captura imágenes y videos en tiempo real, permitiendo la identificación visual de eventos sospechosos y la activación de grabaciones o alertas automáticas.
Sensores de Puertas/Ventanas	Sensores magnéticos de contacto	Monitorea la apertura o cierre de puertas y ventanas, disparando alarmas o notificando al usuario si se detecta una intrusión.
Módulo de Comunicación Wi-Fi	ESP8266 o ESP32	Permite la conectividad del sistema a una red Wi-Fi, facilitando la comunicación con la aplicación móvil y la nube para el control remoto y las notificaciones.
Sirena de Alarma	Sirena piezoeléctrica	Emite un sonido de alta potencia cuando se detecta una intrusión, actuando como elemento disuasorio.
Pantalla LCD o LED	Pantalla LCD 16x2 o Pantalla OLED	Controla dispositivos como luces, cerraduras eléctricas, o cualquier otro actuador que pueda ser activado automáticamente en respuesta a un evento.

Controladores de Acceso (Teclado, RFID)	Módulo RFID-RC522 o teclado matricial	Permite la autenticación de usuarios para armar o desarmar el sistema de seguridad.
Módulo RTC (Real-Time Clock)	DS3231	Proporciona la hora y fecha exacta para registrar eventos y establecer programaciones automáticas en el sistema.
Fuente de Alimentación	Adaptador de corriente 9V 1A o Power Bank	Proporciona energía constante al sistema, asegurando su funcionamiento continuo. Se puede utilizar una batería de respaldo en caso de cortes de energía.
Buzze	Buzzer piezoeléctrico	Proporciona señales acústicas para notificaciones breves o confirmación de acciones del sistema.

Fuente: (Arce, 2023)

Panel de Control Centralizado

Un panel de control centralizado es el cerebro del sistema de seguridad. Diseñado para gestionar todos los dispositivos conectados, como sensores de movimiento, cámaras y alarmas, este panel se comunica con los otros componentes a través de protocolos seguros, garantizando la operación coordinada y eficaz del sistema.

Sensores de Movimiento PIR

Los sensores de movimiento PIR (infrarrojos pasivos) son elementos esenciales para la detección de intrusos. Estos dispositivos identifican variaciones en los niveles de infrarrojos emitidos por objetos en movimiento, lo que provoca la activación automática del sistema de alarma. Son particularmente efectivos cuando se colocan en puntos estratégicos de la casa, lo que permite optimizar la cobertura de seguridad. En la figura 3 se muestran los sensores de movimiento PIR.

Figura 13. *Sensores de Movimiento PIR*



Nota: En la figura se indica la ubicación y el rango de detección del sensor de movimiento PIR.

Fuente: (Chico, 2022)

Cámaras de Seguridad IP

Las cámaras IP son fundamentales para la vigilancia continua del hogar. Con su conectividad a Internet, permiten el monitoreo en tiempo real desde cualquier ubicación. Además, están equipadas con funciones como grabación en alta resolución, visión nocturna y detección de movimiento, garantizando un monitoreo constante y eficiente. En la figura 4 se muestran las cámaras de seguridad IP.

Figura 14. Cámaras de Seguridad IP



Fuente: En la figura se muestra la cámara de seguridad IP, destacando su capacidad para transmitir video en tiempo real a través de la red. Fuente: (Salvador, 2022)

Sensores Magnéticos de Contacto

Estos sensores detectan variaciones en el campo magnético y se activan cuando un imán se acerca. Son comúnmente utilizados en sistemas de seguridad y automatización, como en puertas o ventanas para detectar su apertura o cierre. Su simplicidad y fiabilidad los hacen ideales para aplicaciones de seguridad en hogares y oficinas. En la figura 5 se muestran los sensores magnéticos de contacto.

Figura 15. Sensores Magnéticos de Contacto

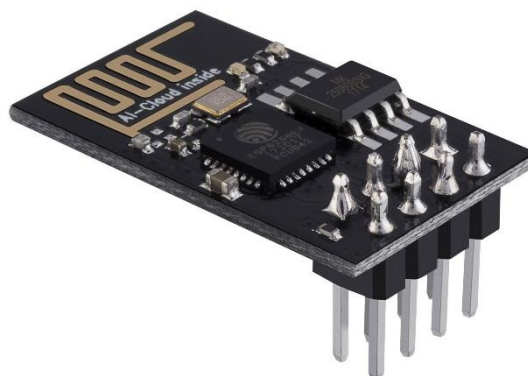


Nota: Los sensores magnéticos de contacto detectan la apertura de puertas o ventanas al registrar cambios en el campo magnético cuando se separan del imán asociado. Fuente: (Rodríguez, 2018)

Módulo de comunicación wifi ESP8266 o ESP32

Ambos son microcontroladores ampliamente utilizados en el desarrollo de proyectos IoT. El ESP8266 destaca por su bajo costo y conectividad Wi-Fi, facilitando la integración de dispositivos a la red de manera sencilla. Por su parte, el ESP32, que también ofrece Wi-Fi y Bluetooth, proporciona mayor capacidad de procesamiento y un mayor número de pines GPIO, lo que lo hace más adecuado para aplicaciones complejas. Además, el ESP32 incorpora mejoras en la gestión de energía, lo que lo convierte en una opción ideal para dispositivos que requieren eficiencia energética. En la figura 6 se muestran el módulo de comunicación WiFi ESP8266 y el ESP32.

Figura 16. Módulo de comunicación wifi ESP8266 o ESP32



Nota: El módulo de comunicación WiFi ESP8266 o ESP32 permite la conexión inalámbrica de dispositivos a redes WiFi, facilitando la comunicación y el control remoto. Fuente: (Vega, 2021)

Sirena Piezoeléctrica

Este dispositivo emite un sonido fuerte cuando se aplica una corriente eléctrica, comúnmente utilizado en alarmas de seguridad. La tecnología piezoeléctrica permite generar un sonido intenso con un consumo de energía relativamente bajo, lo que es crucial en sistemas de seguridad doméstica que funcionan con baterías. En la figura 7 se muestra la sirena piezoeléctrica.

Figura 17. Sirena Piezoeléctrica



Nota: La sirena piezoeléctrica emite un sonido fuerte y agudo al convertir señales eléctricas en vibraciones acústicas, utilizada comúnmente para alertas y sistemas de alarma. Fuente: (Gómez, 2022)

Pantalla LCD 16x2 o Pantalla OLED

La pantalla LCD 16x2 es una opción tradicional para mostrar información en proyectos electrónicos, con capacidad para mostrar dos líneas de 16 caracteres. En la figura 8 se presentan la pantalla LCD 16x2 y la pantalla OLED, que proporcionan un mejor contraste y menor consumo de energía, además de ser visibles incluso en condiciones de alta luminosidad, lo que las hace más adecuadas para dispositivos portátiles o de bajo consumo.

Figura 18. Pantalla LCD 16x2 o Pantalla OLED



Nota: La pantalla LCD 16x2 o la pantalla OLED muestran información visualmente, con la LCD ofreciendo dos líneas de texto y la OLED proporcionando una mayor calidad de Figura y contraste. Fuente: (Vega, 2021)

Módulo RFID-RC522 o Teclado Matricial

El módulo RFID-RC522 permite la lectura de tarjetas RFID, comúnmente usado en sistemas de acceso. En la figura 9 se muestran el módulo RFID-RC522 y el teclado matricial, que ofrece una forma sencilla de entrada de datos y se utiliza frecuentemente en sistemas de autenticación por código PIN.

Figura 19. Módulo RFID-RC522 o Teclado Matricial

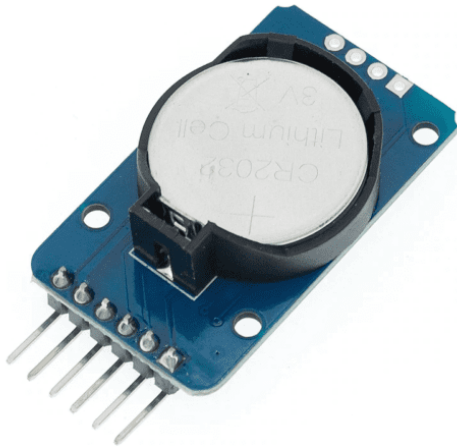


Nota: El módulo RFID-RC522 permite la lectura de tarjetas y etiquetas RFID, mientras que el teclado matricial facilita la entrada de datos mediante una cuadrícula de botones. Fuente: (Portilla, 2021)

Módulo RTC DS3231

Este es un reloj en tiempo real (RTC) de alta precisión, que incorpora un oscilador de cristal compensado térmicamente para garantizar una medición exacta del tiempo, incluso bajo condiciones de temperatura variable. En la figura 10 se muestra el módulo RTC DS3231, comúnmente utilizado en dispositivos que necesitan un seguimiento temporal continuo, como los sistemas de registro de datos.

Figura 20. Módulo RTC DS3231



Nota: El módulo RTC DS3231 proporciona una medición precisa del tiempo y la fecha con alta estabilidad y bajo consumo de energía. Fuente: (Portilla, 2021)

Adaptador de Corriente 9V 1A

Estos dispositivos proporcionan la energía necesaria para proyectos electrónicos portátiles o que requieran un suministro constante y fiable. En la figura 11 se muestra el adaptador de corriente de 9V 1A, ideal para alimentar microcontroladores y periféricos, mientras que un power bank ofrece la flexibilidad de usar dispositivos sin estar conectados a la red eléctrica.

Figura 21. Adaptador de Corriente 9V 1A



Nota: El adaptador de corriente 9V 1A proporciona una alimentación estable de 9 voltios y 1 amperio para dispositivos electrónicos. Fuente: (Villegas, 2021)

Buzzer Piezoeléctrico

Similar a la sirena piezoeléctrica, este dispositivo produce un sonido cuando se le aplica corriente eléctrica, pero en menor escala, lo que lo hace adecuado para alarmas de menor

intensidad o para notificaciones sonoras en dispositivos portátiles. En la figura 12 se muestra el buzzer piezoeléctrico.

Figura 22. Buzzer Piezoeléctrico



Nota: El buzzer piezoeléctrico emite tonos audibles al convertir señales eléctricas en vibraciones acústicas. Fuente: (Arce, 2023)

Sistemas de Alarma

El sistema de alarma está diseñado para alertar a los residentes y a las autoridades en caso de una violación de seguridad. Este sistema se activa mediante señales recibidas de los sensores de movimiento o cámaras, y puede emitir alertas audibles, así como notificaciones a dispositivos móviles.

Automatización de Iluminación

El sistema de iluminación automatizado se integra con el sistema de seguridad para simular la presencia de personas en el hogar, disuadiendo posibles intrusos. Este sistema permite la programación de luces para encenderse y apagarse en horarios específicos, y también se activa automáticamente en respuesta a eventos detectados por los sensores.

4.2.1. Descripción de la propuesta

Se propone la implementación de un sistema de seguridad automatizado que abarque desde la vigilancia hasta la respuesta ante intrusos. Este sistema será implementado mediante un conjunto de equipos que interactuarán de manera constante para lograr su cometido.

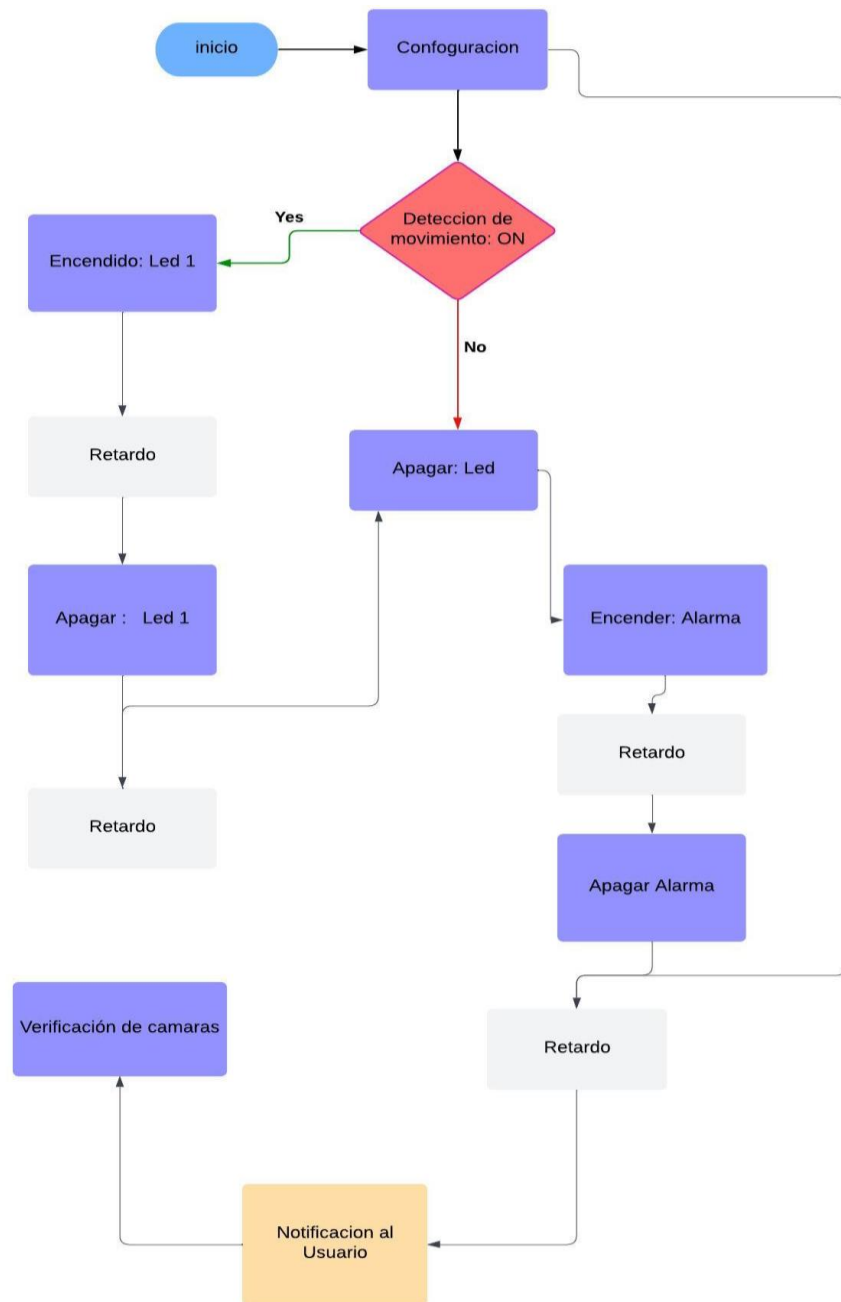
Los dispositivos utilizados para este sistema fueron seleccionados tras una exhaustiva investigación, basada en un presupuesto estipulado por la empresa. Cabe destacar que cada uno de los equipos implementados cumple con las funciones requeridas, desarrolladas durante la

etapa de diseño. A continuación, se detallará de manera más específica el proceso de diseño, implementación y pruebas.

a. Estructura General

El proyecto está compuesto por un panel de control centralizado, que actúa como el núcleo del sistema, procesando y transmitiendo toda la información entre los diferentes dispositivos. El panel de control deberá ser lo más intuitivo posible para evitar errores durante la operación del sistema. A continuación, se presenta el diagrama de flujo del sistema en la figura 13, en su estado automático y semiautomático.

Figura 2. Estructura general

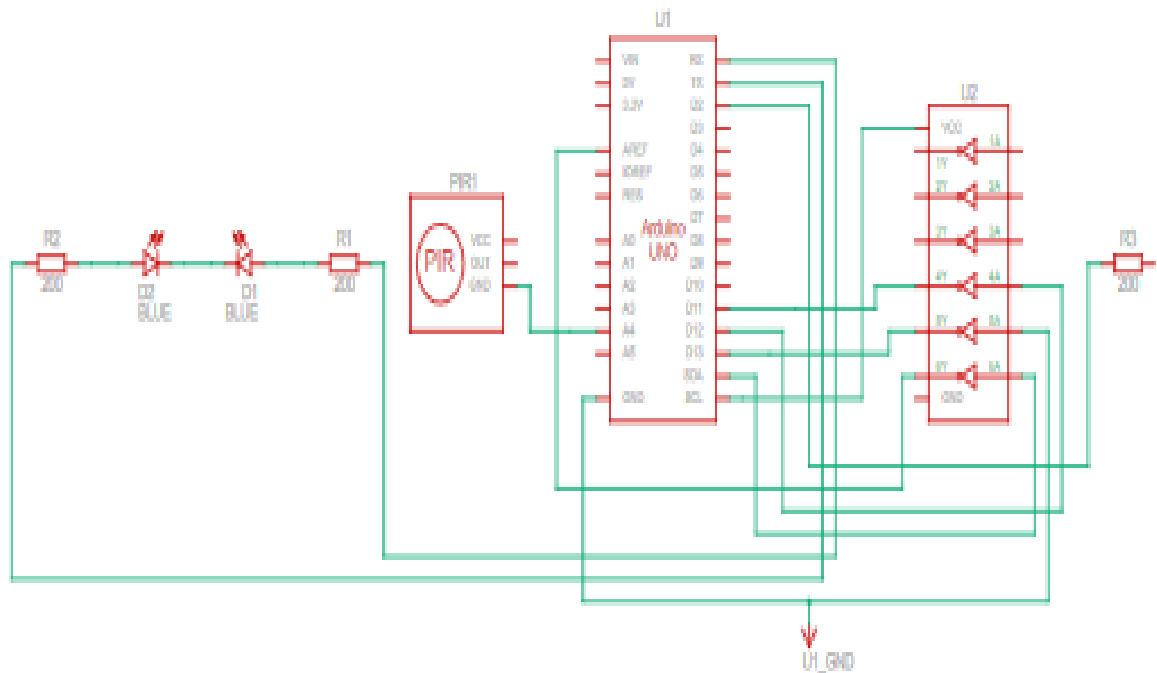


Nota: El proyecto incluye un panel de control centralizado, que gestiona y comunica la información entre los dispositivos, y a continuación se presenta el diagrama de flujo del sistema en sus modos automático y semiautomático. Elaboración propia

Topología

En la figura 14 se muestra la topología del sistema, que detalla la disposición y conexión de sus componentes.

Figura 3. *Tipología del sistema*



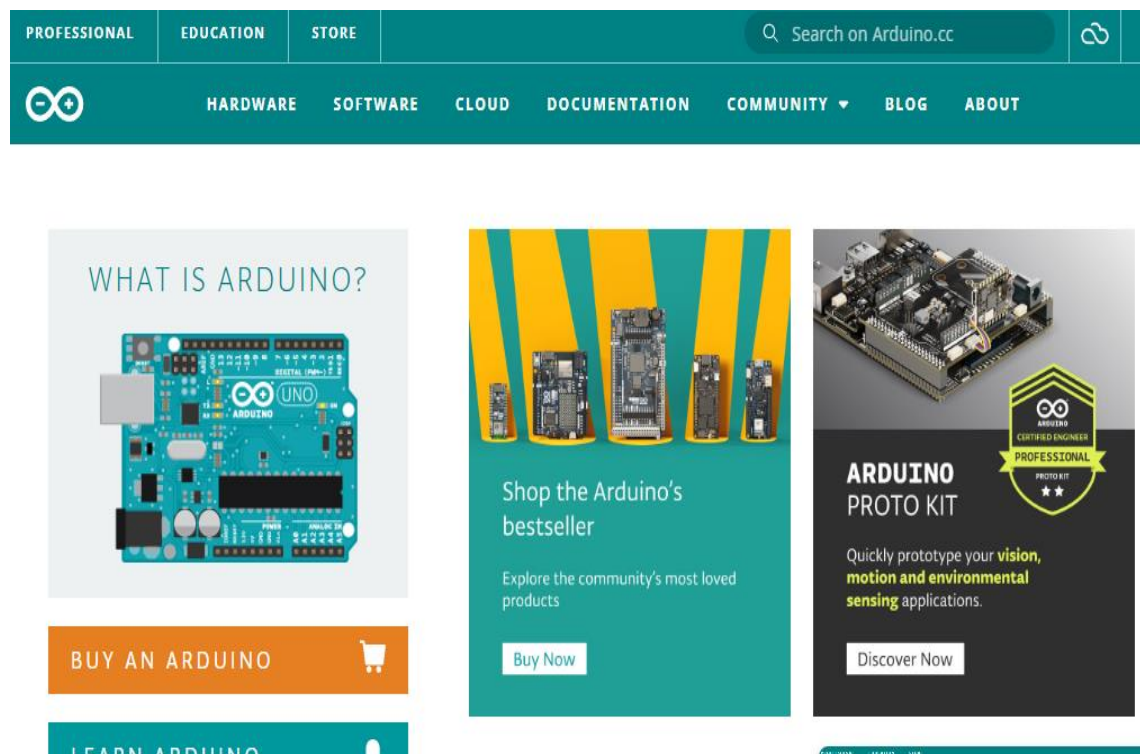
Nota: La topología del sistema se refiere a la clasificación y organización de sus componentes y funciones según su estructura y operación. Elaboración propia

4.3. Descarga E Instalación de ARDUINO IDE

Para la utilización del microcontrolador en este presente proyecto se va a utilizar herramientas y documentación para poder tener un sistema bien empleado y pueda entregar un control estable y sin errores para su optimación en el transcurso que se agregaran los demás componentes.

Ir a la página oficial de Arduino: [Arduino_IDE](#).

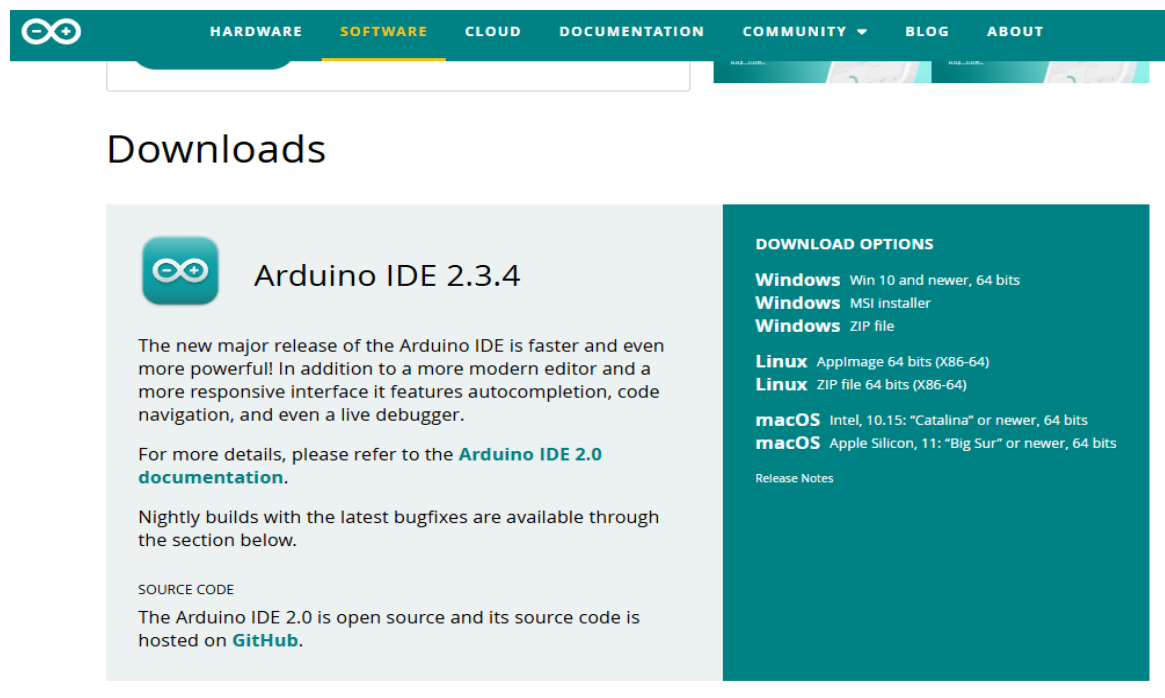
Figura 23. Descarga Arduino



Una vez dentro de la página oficial ve a Documentación ahí encuentras el software y el modelo que buscas para la implementación del proyecto en este punto en la documentación se busca primero descargar el software, en el apartado se encuentra la última versión la cual es ARDUINO IDE 2.3.4 esta última versión hace referencia a las mejoras y optimizaciones que van quizá copándose por el uso de las mismas en generaciones nuevas y así ofreciendo una mejor renderización es sus microcontroladores, hay disponibilidad de descargar para varios sistemas Operativos como son:

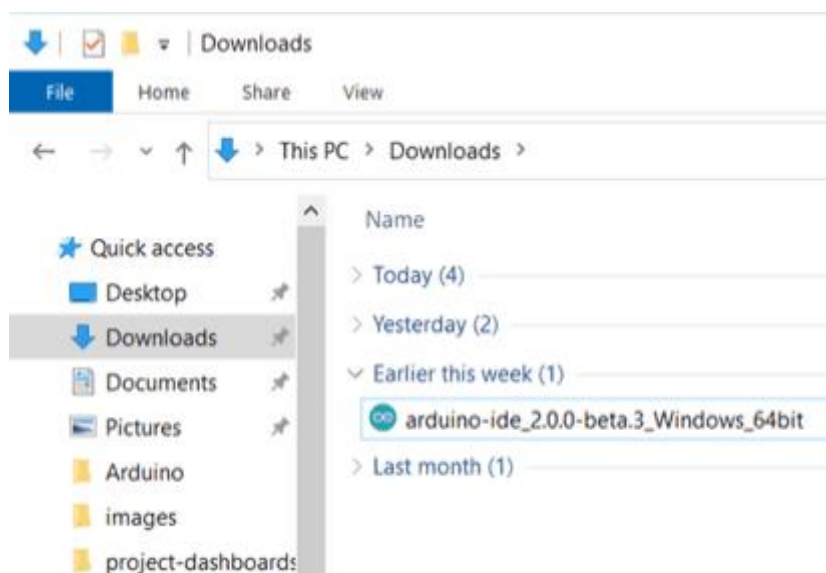
- Windows 64 bits
- Linux x86-64
- macOS 64 bits

Figura 24. Opciones de descarga Arduino



Para instalar Arduino IDE en una computadora con Windows, simplemente ejecute el archivo descargado de la página del software.

Figura 25. Instalación Arduino



En la documentación de [ARDUINO IDE](#) puedes encontrar el paso a paso dependiendo el sistema operativo al cual deseas instalar el software a continuación detallo la instalación en un maquina con sistema operativo Windows.

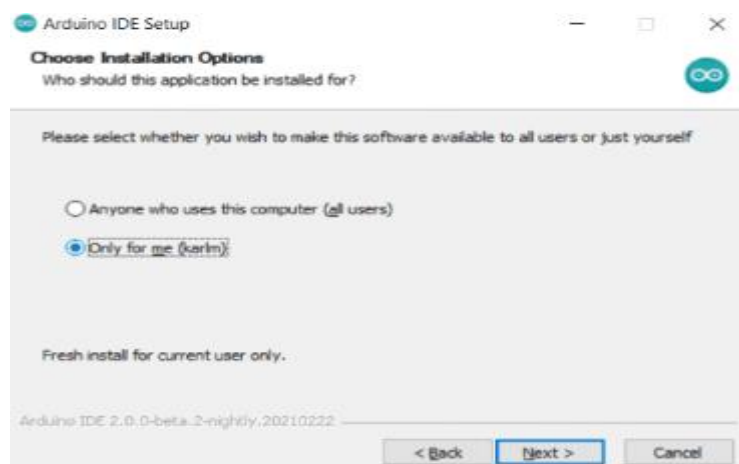
Para continuar con la instalación presiona I Agree

Figura 26. Proceso de instalación Arduino



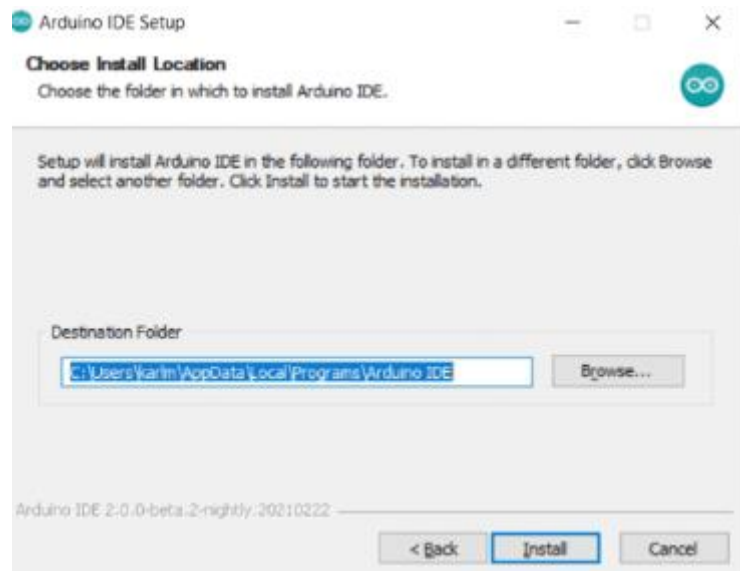
Después tendrás una vista con dos opciones la cual deberás seleccionar la opción z y luego presionar Next

Figura 27. Proceso de instalación Arduino 1



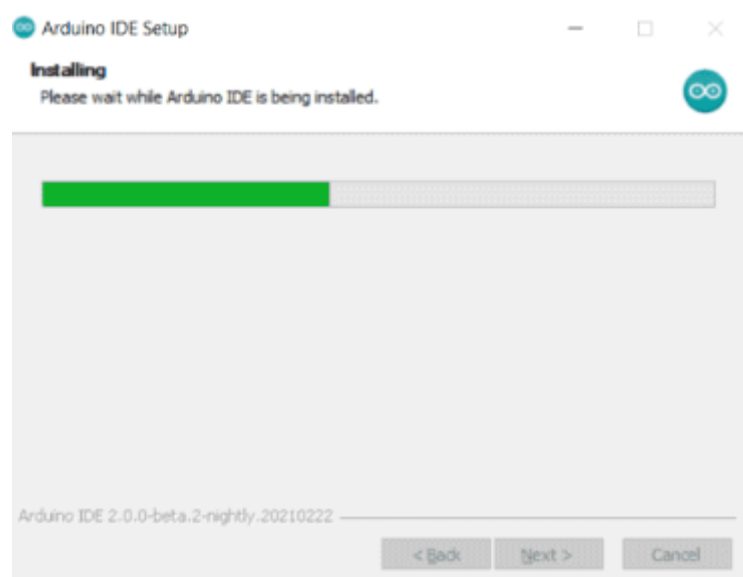
Te muestra el destino en donde será instalado y que carpeta contendrá tu programa aquí se deja por defecto la ubicación del archivo y presiona Install aquí es donde procede a instalar el software que se va a utilizar en este proyecto.

Figura 28. *Proceso de instalación Arduino 2*



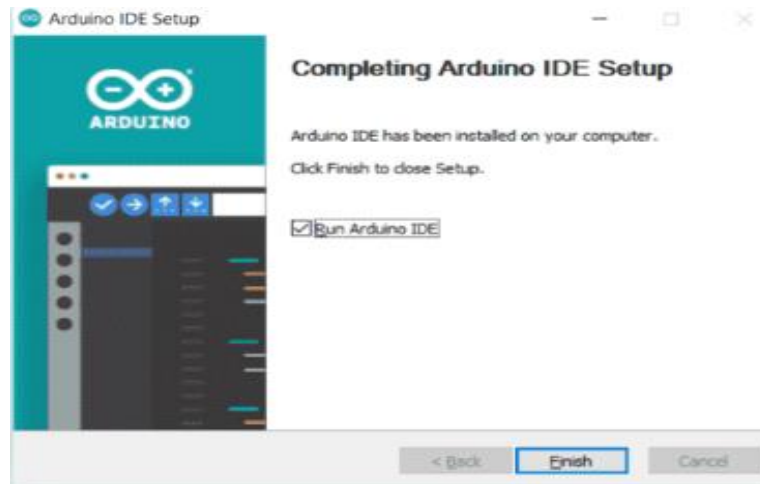
Empieza su instalación al para el tema de tiempo de instalación todo tendrá que ver tu red, se demora al menos unos 5 minutos en procesar la instalación.

Figura 29. *Proceso de instalación Arduino 3*



Se completo la descarga de ARDUINO IDE SETUP, antes de finalizar y ejecutar el programa asegúrate de que se haya instalado correctamente, después de una breve verificación puedes presionar Finalizar y veras como se despliega el programa y ya podrías usarlo.

Figura 30. Proceso de instalación Arduino 4

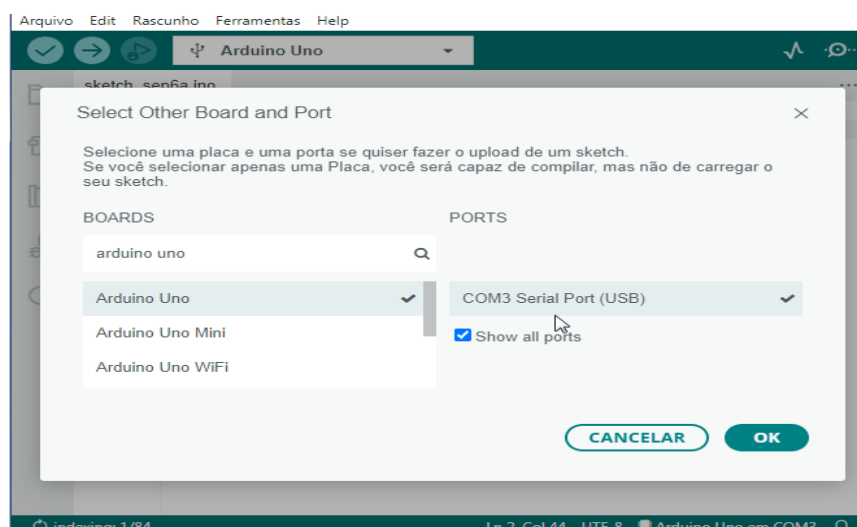


El programa no tubo complicaciones y se abre y ahora se procede a la configuración del IDE de Arduino.

- 1.- Usa el cable USB para conectar tu placa Arduino a la computadora
- 2.- Asegúrate de que el LED de alimentación en la placa este encendido

IDE:

Figura 31. Proceso de instalación52

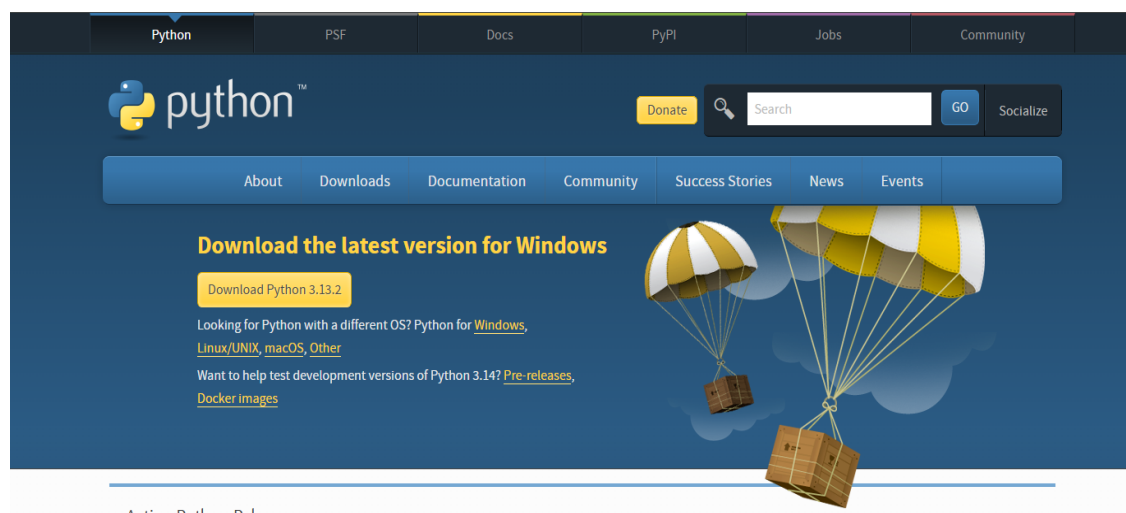


En el buscador se ingresó la palabra clave Arduino Uno encontró, en BOARD se selecciona el que se va a utilizar y en PORTS se presiona para que marque con un visto el USB en la cual está ya conectado.

4.4. Descarga E Instalación de PYTHON

Python es un lenguaje de alto nivel que permite desarrollar rápidamente el código necesario para manejar la **comunicación serial** entre **Arduino** y el sistema de seguridad. Esto es esencial para obtener datos de sensores (como RFID y RTC) y controlar componentes como **sirenas** o **pantallas LCD**.

Figura 32. *Instalación Phyton*



Agregar Python al PATH hace que el sistema operativo reconozca Python como un comando ejecutable desde cualquier terminal o línea de comandos. Esto significa que puedes usar Python y pip (el gestor de paquetes de Python) desde cualquier lugar sin tener que escribir la ruta completa al ejecutable de Python.

Figura 33. Proceso de instalación Python



Facilita la ejecución de comandos como `python --version` o `pip install` desde cualquier directorio en tu computadora, lo cual es esencial para trabajar de manera fluida con Python y sus bibliotecas.

4.5. Descarga E Instalación de Visual Studio Code

Visual Studio Code (VSCode) es un entorno de desarrollo ligero y poderoso que ofrece muchas ventajas para el desarrollo de proyectos en Python y Arduino.

Puedes instalar extensiones para añadir funcionalidades, como la librería de Python para IntelliSense (autocompletado y sugerencias de código) y la extensión de Arduino para cargar y depurar tus programas de Arduino.

VSCode ofrece integración directa con Git, lo que te permite controlar versiones de tu proyecto, realizar commits, y subir los cambios a repositorios de GitHub o cualquier otro servicio de control de versiones.

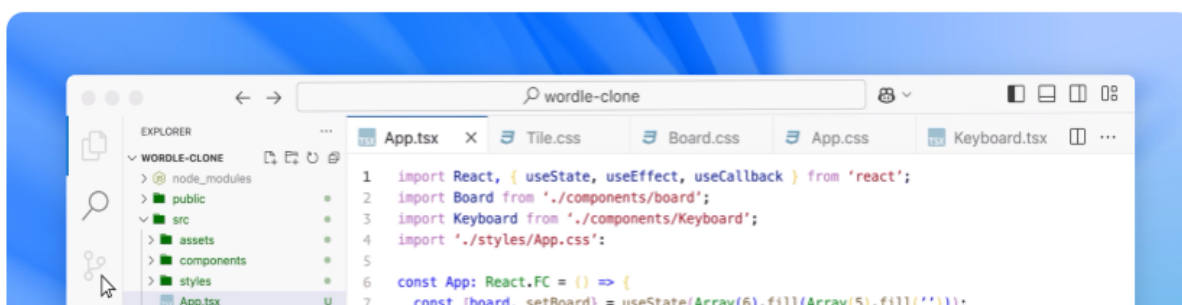
Figura 34. Proceso de instalación Visual Code

Tu editor de código. Redefinido con IA.

Descargar para Windows

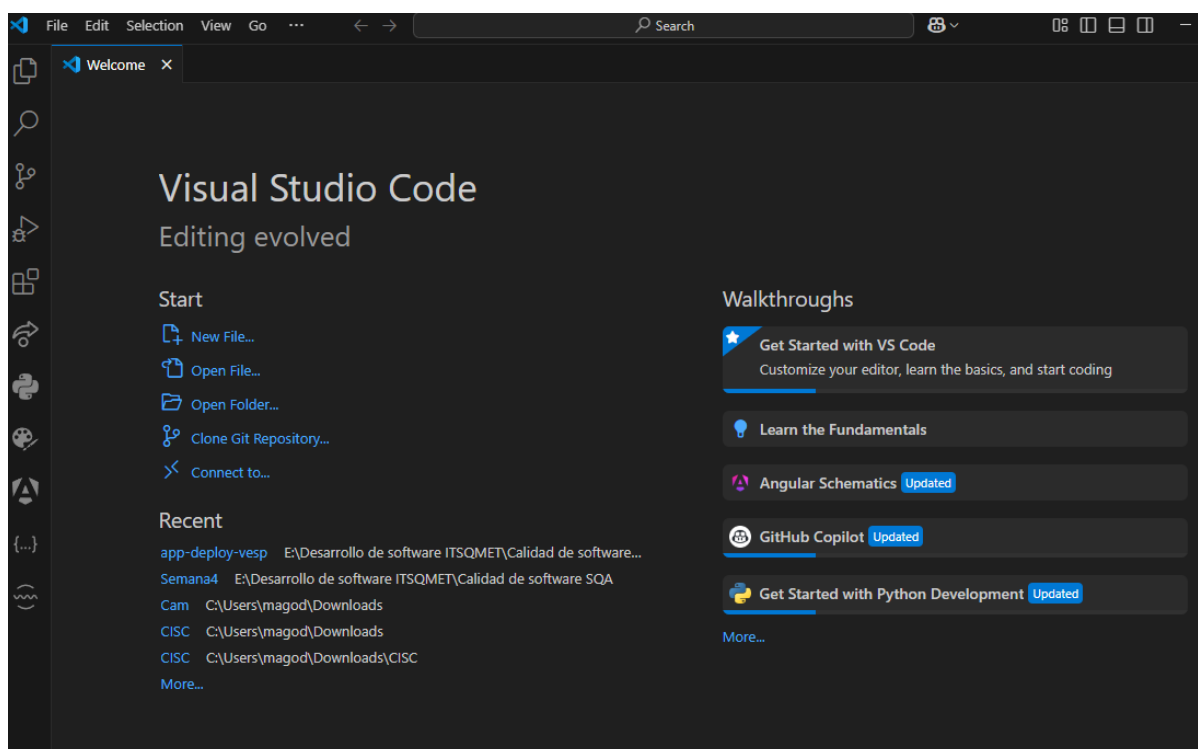
Obtenga Copilot gratis

[Web](#), [edición Insiders](#) u [otras plataformas](#)



Funciona en sistemas operativos **Windows**, **Linux** y **macOS**, lo que lo hace accesible en diferentes plataformas sin perder la funcionalidad.

Figura 35. Proceso de instalación Visual Code 1



Siguientes pasos Después de Instalar Visual Studio Code

Una vez que hayas instalado Visual Studio Code, estos son los pasos para configurar tu entorno de desarrollo y comenzar a trabajar en el proyecto:

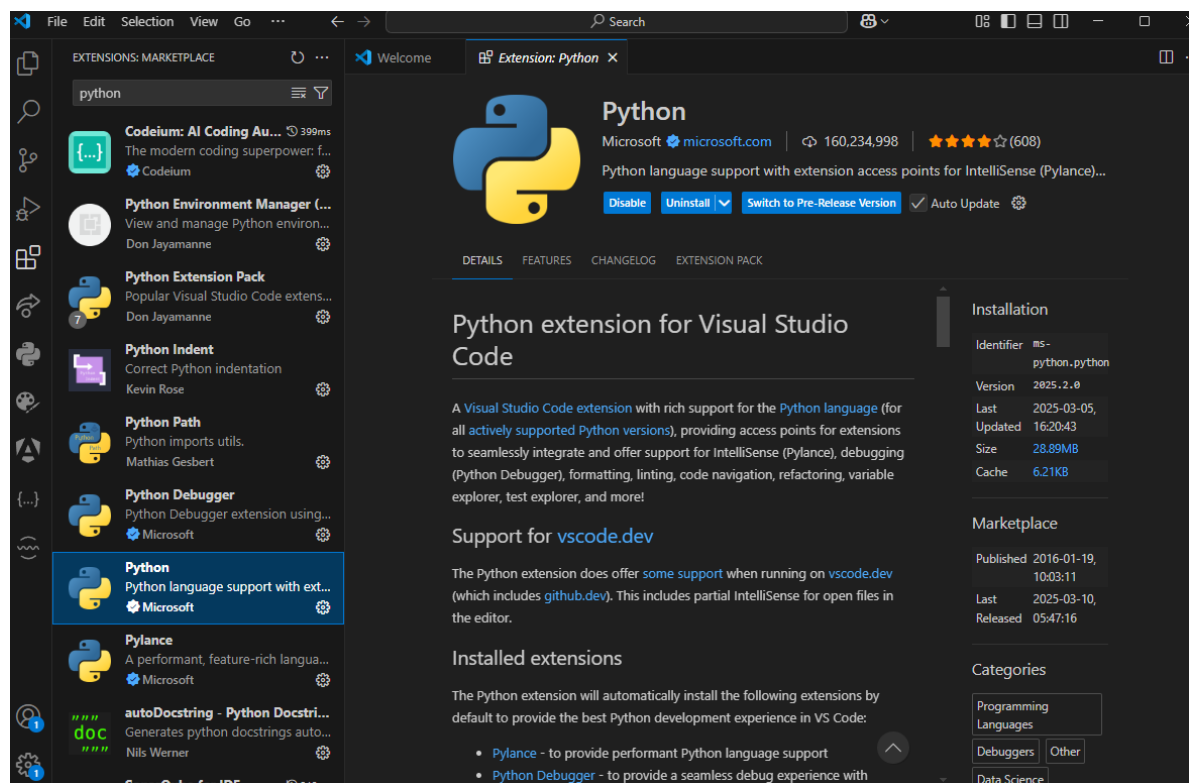
Instalación de las Extensiones Necesarias

Extensión de Python

- 1.- Haz clic en el ícono de extensiones en la barra lateral izquierda o usa el atajo Ctrl+Shift+X.
- 2.- En el campo de búsqueda, escribe Python.
- 3.- Instala la extensión Python desarrollada por Microsoft.

Esta extensión te proporcionará características como autocompletado, depuración y ejecución de código de manera sencilla.

Figura 36. Proceso de instalación vinculación Visual Code y Phyton

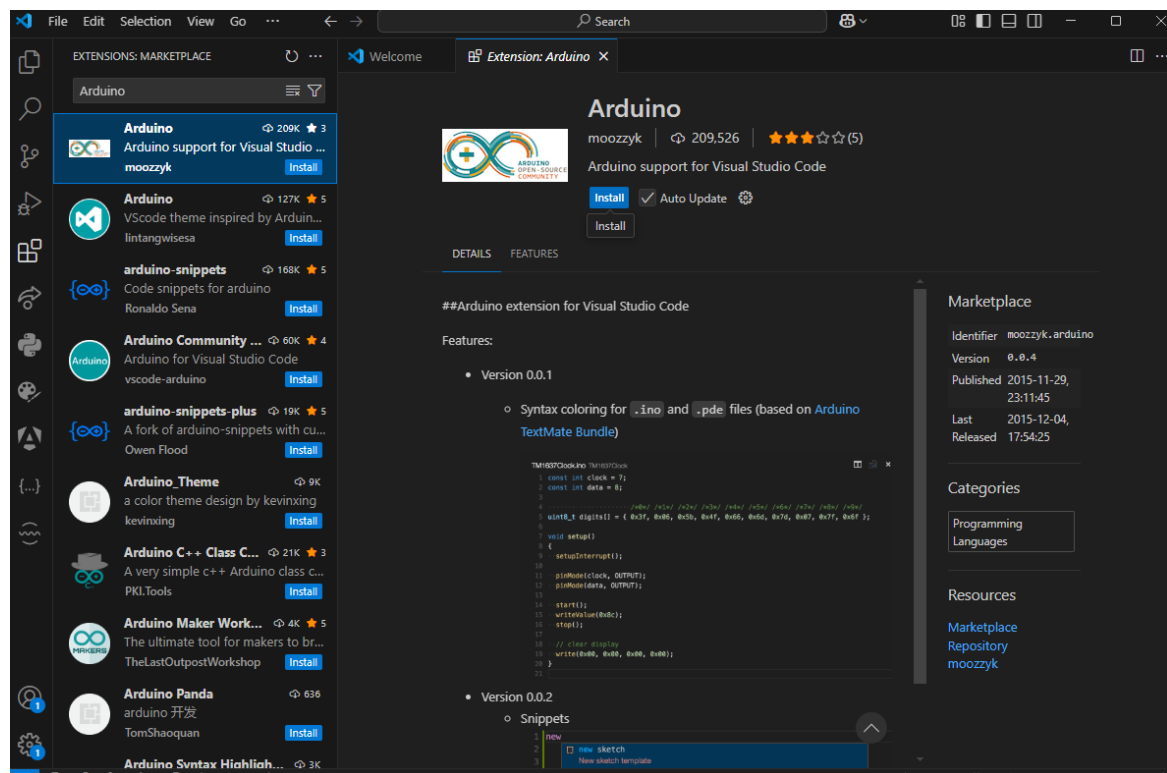


Extensión para Arduino

- 1.- En el mismo menú de extensiones, busca **Arduino**.
- 2.- Instala la extensión **Arduino for Visual Studio Code**.

Esta extensión te permitirá programar y cargar tus programas en el **Arduino** desde VSCode.

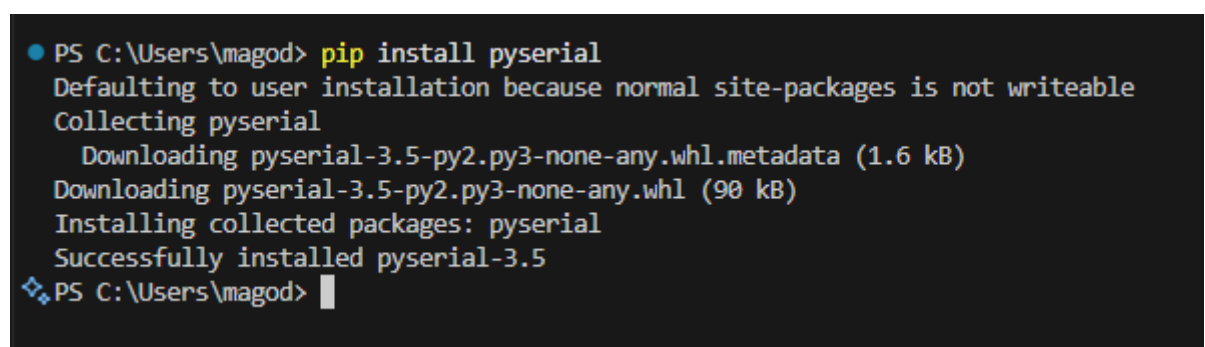
Figura 37. Visual Code y Python 1



Instalar el Soporte de Serial

Para la comunicación **serial** entre **Arduino** y **Python**, asegúrate de tener la librería **PySerial** instalada:

Figura 38. Soporte de Serial

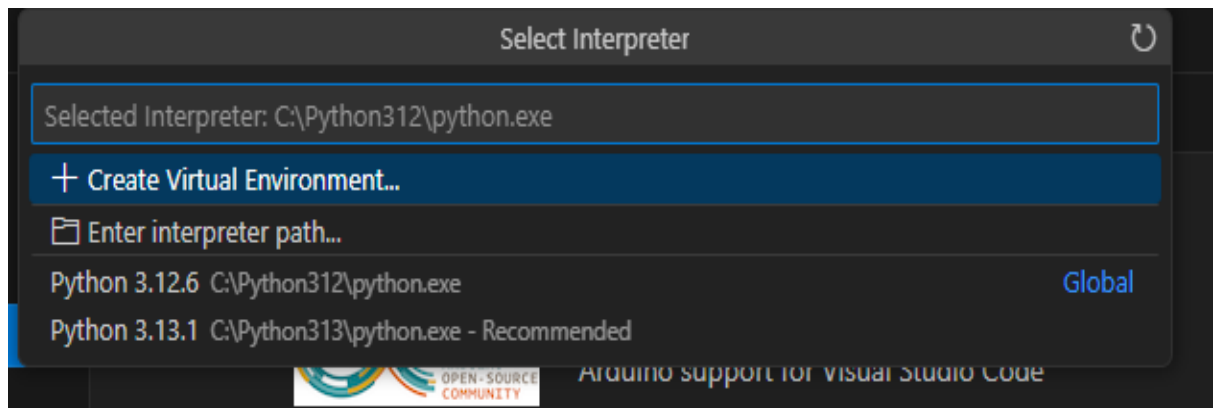


Configuración de Python

Después de instalar la extensión de Python, selecciona el intérprete de Python adecuado:

- 1.- Abre el Command Palette con Ctrl+Shift+P.
- 2.- Escribe Python: Select Interpreter.
- 3.- Selecciona la versión de Python que quieres usar (asegúrate de que esté correctamente instalada y configurada).

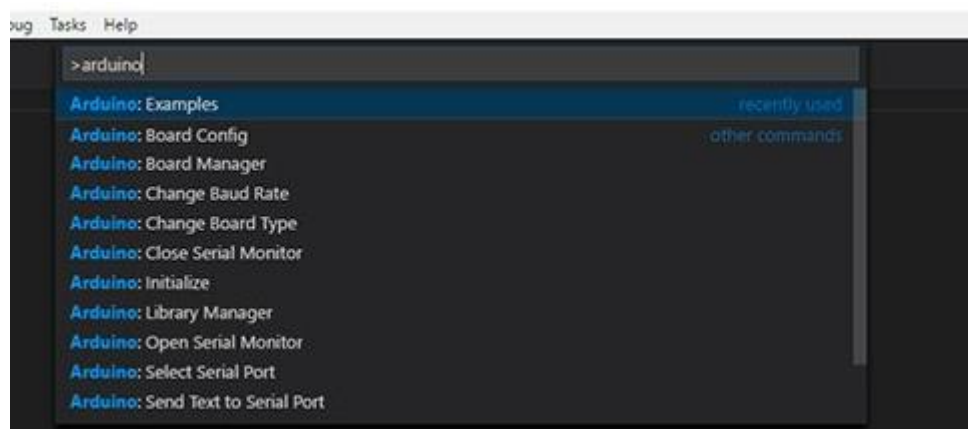
Figura 39. Configuración Python



Configuración de Arduino en VSCode

- 1.- Asegúrate de que tienes instalado el Arduino IDE en tu computadora.
- 2.- En VSCode, abre el Command Palette con Ctrl+Shift+P.
- 3.- Escribe Arduino: Select Board y selecciona tu modelo de placa Arduino (como Arduino Uno, Arduino Mega, etc.).

Figura 40. Configuración de Arduino en VSCode



4.6. Instalación de librerías

Una **librería** en programación es un conjunto de funciones, clases y recursos predefinidos que se pueden utilizar para realizar tareas específicas sin necesidad de escribir el código desde cero. Las librerías permiten a los desarrolladores reutilizar código, lo que ahorra tiempo y esfuerzo, y facilita la implementación de funcionalidades complejas.

OpenCV

Es una librería de visión por computadora que proporciona herramientas para el procesamiento de imágenes, videos y detección de objetos (como rostros, cuerpos, etc.).

Comandos de instalación es:

Figura 41. *Pip Install opencv-python*

```
PS C:\Users\magod> pip install opencv-python
Defaulting to user installation because normal site-packages is not writeable
Collecting opencv-python
  Using cached opencv_python-4.11.0.86-cp37-abi3-win_amd64.whl.metadata (20 kB)
Requirement already satisfied: numpy>=1.21.2 in c:\users\magod\appdata\roaming\python\python313\site-packages (from opencv-python) (2.2.2)
Using cached opencv_python-4.11.0.86-cp37-abi3-win_amd64.whl (39.5 MB)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.11.0.86
PS C:\Users\magod>
```

Figura 42. *Pip install opencv-python-headless*

```
PS C:\Users\magod> pip install opencv-python-headless
Defaulting to user installation because normal site-packages is not writeable
Collecting opencv-python-headless
  Downloading opencv_python_headless-4.11.0.86-cp37-abi3-win_amd64.whl.metadata (20 kB)
Requirement already satisfied: numpy>=1.21.2 in c:\users\magod\appdata\roaming\python\python313\site-packages (from opencv-python-headless) (2.2.2)
Downloading opencv_python_headless-4.11.0.86-cp37-abi3-win_amd64.whl (39.4 MB)
  39.4/39.4 MB 5.7 MB/s eta 0:00:00
Installing collected packages: opencv-python-headless
Successfully installed opencv-python-headless-4.11.0.86
PS C:\Users\magod>
```

NumPy

Es una librería para trabajar con arrays multidimensionales y matrices, y es fundamental para el procesamiento de imágenes en OpenCV.

Comando de instalación:

Figura 43. *Pip install numpy*

```
PS C:\Users\magod> pip install numpy
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\users\magod\appdata\roaming\python\python313\site-packages (2.2.2)
PS C:\Users\magod>
```

Pygame

La librería pygame se usa para reproducir sonidos, como en tu caso cuando se detecta un rostro o intrusión.

Comando de instalación:

Figura 44. *Pip install pygame*

```
PS C:\Users\magod> pip install pygame
Defaulting to user installation because normal site-packages is not writeable
Collecting pygame
  Downloading pygame-2.6.1-cp313-cp313-win_amd64.whl.metadata (13 kB)
  Downloading pygame-2.6.1-cp313-cp313-win_amd64.whl (10.6 MB)
    10.6/10.6 MB 5.3 MB/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.6.1
PS C:\Users\magod>
```

OS

La librería os se utiliza para interactuar con el sistema operativo, como para la manipulación de archivos y carpetas. Esta librería viene preinstalada con Python, por lo que no necesitas instalarla.

Datetime

La librería datetime se utiliza para manejar fechas y horas, lo cual es útil cuando necesitas guardar registros de los eventos de detección. Esta librería también viene preinstalada con Python.

Pillow

Con esta librería se puede hacer alguna manipulación avanzada de imágenes, como abrir o guardar imágenes en diferentes formatos, puedes necesitar Pillow, una librería de imágenes.

Comando de instalación:

Figura 45. *Pip install pillow*

```
PS C:\Users\magod> pip install pillow
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pillow in c:\users\magod\appdata\roaming\python\python313\site-packages (11.1.0)
PS C:\Users\magod>
```

Pyserial

Con esta librería Python puede hacer conexión con Arduino y poder trabajar o modificarse entre sí.

Comando de Instalación:

Figura 46. *Pip install pyserial requests*

```
PS C:\Users\magod\Downloads\ArduinoIntrusion> pip install pyserial requests
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pyserial in c:\users\magod\appdata\roaming\python\python313\site-packages (3.5)
Requirement already satisfied: requests in c:\users\magod\appdata\roaming\python\python313\site-packages (2.32.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\magod\appdata\roaming\python\python313\site-packages (from requests) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\magod\appdata\roaming\python\python313\site-packages (from requests) (2.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\magod\appdata\roaming\python\python313\site-packages (from requests) (2.3.0)
Requirement already satisfied: certifi<2017.4.17 in c:\users\magod\appdata\roaming\python\python313\site-packages (from requests) (2025.1.31)
PS C:\Users\magod\Downloads\ArduinoIntrusion>
```

4.7. Clasificadores en Cascada (Haar Cascade) de OpenCV

Son modelos preentrenados utilizados por OpenCV para la detección de objetos en imágenes o videos. Estos clasificadores se basan en el algoritmo de cascada de Haar, que utiliza características visuales simples para identificar objetos en imágenes, como rostros, cuerpos o partes del cuerpo.

Un clasificador en cascada es un modelo de aprendizaje automático entrenado para reconocer patrones en imágenes. En el caso de OpenCV, los clasificadores en cascada se entrenan con un algoritmo llamado Haar features que utiliza características simples como

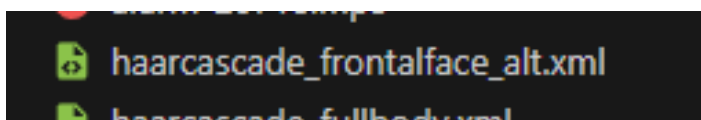
bordes, cambios en intensidad de píxeles, y otras propiedades visuales de la Figura para identificar objetos específicos.

Los clasificadores de Haar son eficientes y rápidos, lo que los hace ideales para detectar objetos en tiempo real en aplicaciones como la detección de rostros, seguimiento de personas o incluso la seguridad.

Detección de Rostros

Este clasificador en cascada está diseñado para detectar rostros humanos en imágenes y videos. El archivo `haarcascade_frontalface_alt.xml` es un modelo preentrenado de OpenCV que se utiliza para detectar los rostros de las personas desde una vista frontal.

El algoritmo utiliza una serie de características Haar para identificar los contornos de un rostro, como la diferencia entre la zona alrededor de los ojos y la nariz. Detecta patrones visuales que corresponden a un rostro humano.

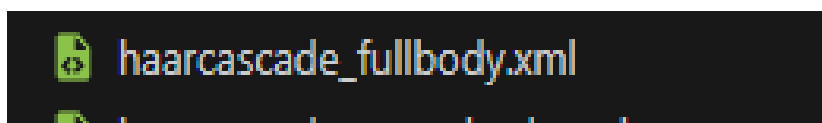


Detección de Cuerpos Completos

Este clasificador en cascada está diseñado para detectar cuerpos completos en una Figura o video. Es útil para la detección de personas en general, especialmente cuando las personas están de pie o caminando.

Similar a la detección de rostros, el clasificador en cascada de cuerpo completo utiliza las características de Haar para identificar la forma general de un cuerpo humano, diferenciando entre el cuerpo y el fondo o el entorno.

Detectar el cuerpo completo es importante cuando se trata de monitorear áreas o entradas donde se necesita verificar la presencia de personas. El sistema puede detectar cuando alguien entra en una zona protegida, permitiendo al sistema actuar (por ejemplo, activar una sirena o enviar una alerta).

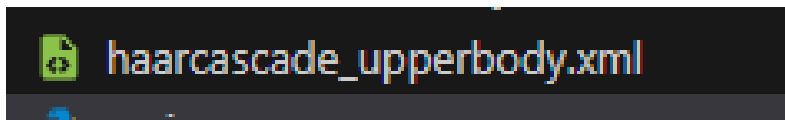


Detección de la Parte Superior del Cuerpo

El clasificador haarcascade_upperbody.xml está diseñado específicamente para detectar la parte superior del cuerpo humano, lo que incluye el torso, brazos, y otras áreas superiores del cuerpo.

Este clasificador utiliza características Haar entrenadas para identificar solo la parte superior del cuerpo, lo que es útil para detectar personas que podrían estar parcialmente cubiertas o cuando no se capturan las piernas.

Detectar solo la parte superior del cuerpo es útil cuando se quiere identificar personas en entornos restringidos donde la visibilidad de las piernas puede ser limitada. Además, puede ser útil para aplicaciones que se centran en identificar movimiento o acciones específicas (por ejemplo, detección de presencia sin capturar toda la figura de la persona).



4.8. Codificación Inicial del Arduino

Librerías Necesarias

Primero, debemos incluir las librerías necesarias para manejar todos los dispositivos conectados:

Figura 47. *Librerías Necesarias Arduino*

```
Arduino.ino X
Arduino.ino
1  #include <SPI.h>           // Librería para SPI (para RFID)
2  #include <Wire.h>          // Librería para I2C (pantalla LCD y RTC)
3  #include <MFRC522.h>       // Librería para el lector RFID
4  #include <LiquidCrystal_I2C.h> // Librería para el LCD I2C
5  #include <RTClib.h>        // Librería para el RTC DS3231
6  #include <Keypad.h>        // Librería para el teclado matricial
7  |
```

Definir los Pines

Aquí configuramos los pines a los que están conectados los dispositivos:

Figura 48. *Definición Pines*

```
7
8 // Pines para el sensor PIR
9 const int pirPin = 8;
10
11 // Pines para el lector RFID
12 #define SS_PIN 10
13 #define RST_PIN 9
14 MFRC522 mfrc522(SS_PIN, RST_PIN); // Crear objeto RFID
15
16 // Pines para el teclado matricial (si se usa)
17 const byte ROW_NUM = 4; // Número de filas del teclado
18 const byte COLUMN_NUM = 4; // Número de columnas del teclado
19 char keys[ROW_NUM][COLUMN_NUM] = {
20     {'1','2','3','A'},
21     {'4','5','6','B'},
22     {'7','8','9','C'},
23     {'*','0','#','D'}
24 };
25 byte pin_rows[ROW_NUM] = {7, 6, 5, 4};
26 byte pin_column[COLUMN_NUM] = {3, 2, 1, 0};
27 Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_column, ROW_NUM, COLUMN_NUM);
28
29 // Pines para la sirena y los LEDs
30 const int speakerPin = 9;
31 const int ledRedPin = 10;
32 const int ledBluePin = 11;
33
34 // Pines para el RTC
35 RTC_DS3231 rtc;
36 LiquidCrystal_I2C lcd(0x27, 16, 2); // Configurar LCD con I2C
37 |
```

Configuración del Setup

La función **setup ()** se ejecuta una sola vez al iniciar el sistema. Aquí, configuramos los pines y comenzamos con la inicialización de los dispositivos:

Figura 49. *Configuración del Setup*

```

37
38 void setup() {
39     // Iniciar comunicación serie
40     Serial.begin(9600);
41
42     // Iniciar el LCD
43     lcd.begin();
44     lcd.print("Sistema de seguridad");
45
46     // Configuración de pines
47     pinMode(pirPin, INPUT);      // Sensor PIR como entrada
48     pinMode(speakerPin, OUTPUT); // Buzzer como salida
49     pinMode(ledRedPin, OUTPUT);  // LED rojo como salida
50     pinMode(ledBluePin, OUTPUT); // LED azul como salida
51
52     // Iniciar el RFID
53     SPI.begin();
54     mfrc522.PCD_Init();
55
56     // Iniciar el RTC
57     if (!rtc.begin()) {
58         lcd.print("No se encuentra RTC");
59         while (1);
60     }
61
62     lcd.clear();
63     lcd.print("Sistema listo");
64 }
65

```

Función loop

La función `loop()` se ejecuta continuamente. Aquí vamos a leer los sensores y realizar las acciones correspondientes.

Figura 50. *Función loop*

```

65
66 void loop() {
67     // Leer el estado del sensor PIR
68     if (digitalRead(pirPin) == HIGH) {
69         // Si se detecta movimiento
70         lcd.clear();
71         lcd.print("Movimiento detectado!");
72         digitalWrite(ledRedPin, HIGH); // Enciende LED rojo
73         digitalWrite(speakerPin, HIGH); // Enciende el buzzer
74     } else {
75         // Si no se detecta movimiento
76         digitalWrite(ledRedPin, LOW); // Apaga LED rojo
77         digitalWrite(speakerPin, LOW); // Apaga el buzzer
78     }
79
80     // Detectar tarjetas RFID
81     if (mfrc522.PICC_IsNewCardPresent()) {
82         if (mfrc522.PICC_ReadCardSerial()) {
83             String rfidData = "";
84             for (byte i = 0; i < mfrc522.uid.size; i++) {
85                 rfidData += String(mfrc522.uid.uidByte[i], HEX);
86             }
87             lcd.clear();
88             lcd.print("RFID detectado: ");
89             lcd.setCursor(0, 1);
90             lcd.print(rfidData); // Muestra el UID de la tarjeta RFID
91         }
92     }
93
94     // Leer la hora del RTC
95     DateTime now = rtc.now();
96     lcd.setCursor(0, 1);
97     lcd.print(now.hour(), DEC);
98     lcd.print(':');
99     lcd.print(now.minute(), DEC);
100    lcd.print(':');
101    lcd.print(now.second(), DEC);
102
103    delay(500); // Retardo de medio segundo antes de volver a leer
104 }
105

```

Configuración de la Conexión Wi-Fi

A continuación, configura la conexión a tu red Wi-Fi. En el **setup()** de tu código de Arduino, establece la conexión a la red Wi-Fi utilizando el nombre de la red (SSID) y la contraseña.

Figura 51. Configuración de la Conexión Wi-Fi

```
Arduino.ino X
Arduino.ino
1  #include <ESP8266WiFi.h> // Para ESP8266
2  // o
3  #include <WiFi.h>        // Para ESP32
4
5  // Datos de la red Wi-Fi
6  const char* ssid = "TuRedWiFi";      // Nombre de la red Wi-Fi
7  const char* password = "TuContraseña"; // Contraseña de la red Wi-Fi
8
9  void setup() {
10     Serial.begin(9600);
11
12     // Conexión Wi-Fi
13     WiFi.begin(ssid, password); // Iniciar conexión Wi-Fi
14
15     // Espera hasta que se establezca la conexión
16     while (WiFi.status() != WL_CONNECTED) {
17         delay(500);
18         Serial.print(".");
19     }
20
21     // Una vez conectado
22     Serial.println("Conexión Wi-Fi exitosa!");
23     Serial.print("IP del dispositivo: ");
24     Serial.println(WiFi.localIP());
25 }
26
```

Enviar Datos a un Servidor Local o Nube

Una vez que el ESP8266/ESP32 esté conectado a Wi-Fi, puedes enviar los datos de los sensores a un servidor **Local o nube**.

Figura 52. Envío de datos

```
Arduino.ino X
Arduino.ino
1  #include <ESP8266HTTPClient.h> // Para ESP8266
2  // o
3  #include <HTTPClient.h>       // Para ESP32
4
5  void sendDataToServer(String data) {
6      if (WiFi.status() == WL_CONNECTED) { // Verifica si está conectado a Wi-Fi
7          HTTPClient http;
8
9          http.addHeader("Content-Type", "application/json"); // Tipo de contenido
10
11         // Crea un objeto JSON para enviar los datos
12         String jsonData = "{\"sensor\": \"movimiento\", \"estado\": \"\" + data + \"\"}";
13
14         int httpResponseCode = http.POST(jsonData); // Enviar los datos
15
16         if (httpResponseCode > 0) {
17             Serial.println("Datos enviados correctamente");
18         } else {
19             Serial.println("Error en el envío de datos");
20         }
21
22         http.end(); // Cerrar la conexión HTTP
23     } else {
24         Serial.println("No conectado a Wi-Fi");
25     }
26 }
27 |
```

Lectura del Sensor de Puerta/Ventana

El código para leer el estado del **sensor de puerta/ventana** es simple. Si el sensor está cerrado, el **OUT** tendrá una señal baja (**LOW**), y si está abierto, tendrá una señal alta (**HIGH**).

Figura 53. Lectura del Sensor de Puerta/Ventana

```
Arduino.ino X
Arduino.ino
1  const int doorSensorPin = 2; // Pin del sensor de puerta/ventana
2
3  void setup() {
4      pinMode(doorSensorPin, INPUT); // Configurar el pin como entrada
5      Serial.begin(9600);
6  }
7
8  void loop() {
9      int sensorState = digitalRead(doorSensorPin); // Leer el estado del sensor
10
11     if (sensorState == HIGH) {
12         // La puerta/ventana está abierta
13         Serial.println("Puerta/Ventana abierta");
14     } else {
15         // La puerta/ventana está cerrada
16         Serial.println("Puerta/Ventana cerrada");
17     }
18     delay(500); // Retardo de medio segundo
19 }
20
```

Integración con el Sistema de Seguridad

Ahora, vamos a integrar los sensores de **movimiento** y **puerta/ventana** con el sistema de seguridad. Cuando se detecte movimiento o una puerta/ventana abierta, el sistema activará la **sirena** y enviará un mensaje a un servidor (si tienes configurada la comunicación Wi-Fi).

- 1.- El ESP8266/**ESP32** se conecta a una red Wi-Fi.
- 2.- Si se detecta **movimiento** o la **puerta/ventana** está abierta, se activa la **sirena** y el **LED rojo**.
- 3.- Los datos se envían a un servidor o API en **JSON** para notificar sobre la alerta.
- 4.- Muestra el estado del sistema en la pantalla **LCD**.

Figura 54. Integración con el Sistema de Seguridad

```
Arduino.ino X
Arduino.ino
1  #include <ESP8266WiFi.h> // o #include <WiFi.h> para ESP32
2  #include <Wire.h>
3  #include <MFRC522.h>
4  #include <LiquidCrystal_I2C.h>
5  #include <RTCLib.h>
6  #include <Keypad.h>
7
8  const char* ssid = "PPAB"; // Nombre de la red Wi-Fi
9  const char* password = "1995BAPP"; // Contraseña de la red Wi-Fi
10
11  const int pirPin = 8; // Sensor PIR
12  const int doorSensorPin = 2; // Sensor de puerta/ventana
13  const int speakerPin = 9; // Buzzer (sirena)
14  const int ledRedPin = 10; // LED rojo
15  const int ledBluePin = 11; // LED azul
16
17  LiquidCrystal_I2C lcd(0x27, 16, 2); // LCD I2C
18
19  void setup() {
20    Serial.begin(9600);
21    pinMode(pirPin, INPUT);
22    pinMode(doorSensorPin, INPUT);
23    pinMode(speakerPin, OUTPUT);
24    pinMode(ledRedPin, OUTPUT);
25    pinMode(ledBluePin, OUTPUT);
26
27    // Conexión Wi-Fi
28    WiFi.begin(ssid, password);
29    while (WiFi.status() != WL_CONNECTED) {
30      delay(500);
31      Serial.print(".");
32    }
33    Serial.println("Conectado a Wi-Fi!");
34    lcd.begin();
35  }
36
```

```

37 void loop() {
38     int pirState = digitalRead(pirPin); // Leer sensor PIR
39     int doorState = digitalRead(doorSensorPin); // Leer sensor de puerta/ventana
40
41     if (pirState == HIGH || doorState == HIGH) {
42         // Si se detecta movimiento o la puerta/ventana está abierta
43         digitalWrite(speakerPin, HIGH); // Activar sirena
44         digitalWrite(ledRedPin, HIGH); // Enciende LED rojo
45
46         sendDataToServer("ALERTA: Movimiento o puerta abierta");
47
48         lcd.clear();
49         lcd.print("Alerta! Movimiento");
50         lcd.setCursor(0, 1);
51         lcd.print("o puerta abierta");
52     } else {
53         digitalWrite(speakerPin, LOW); // Desactivar sirena
54         digitalWrite(ledRedPin, LOW); // Apagar LED rojo
55         digitalWrite(ledBluePin, HIGH); // Enciende LED azul
56
57         lcd.clear();
58         lcd.print("Sistema desactivado");
59     }
60
61     delay(500);
62 }
63

```

```

63
64 // Función para enviar datos al servidor
65 void sendDataToServer(String message) {
66     if (WiFi.status() == WL_CONNECTED) {
67         HTTPClient http;
68         http.addHeader("Content-Type", "application/json");
69
70         String jsonData = "{\"alerta\": \"" + message + "\"}";
71         int httpResponseCode = http.POST(jsonData);
72
73         if (httpResponseCode > 0) {
74             Serial.println("Datos enviados correctamente");
75         } else {
76             Serial.println("Error en el envío de datos");
77         }
78
79         http.end();
80     }
81 }
82

```

4.9. Conexión entre Python y Arduino

Al agregar la librería PySerial para leer los datos enviados por Arduino a través del puerto serial. Suponiendo que Arduino está enviando información sobre el estado de los sensores (movimiento, puerta abierta, etc.), leeremos estos datos en Python.

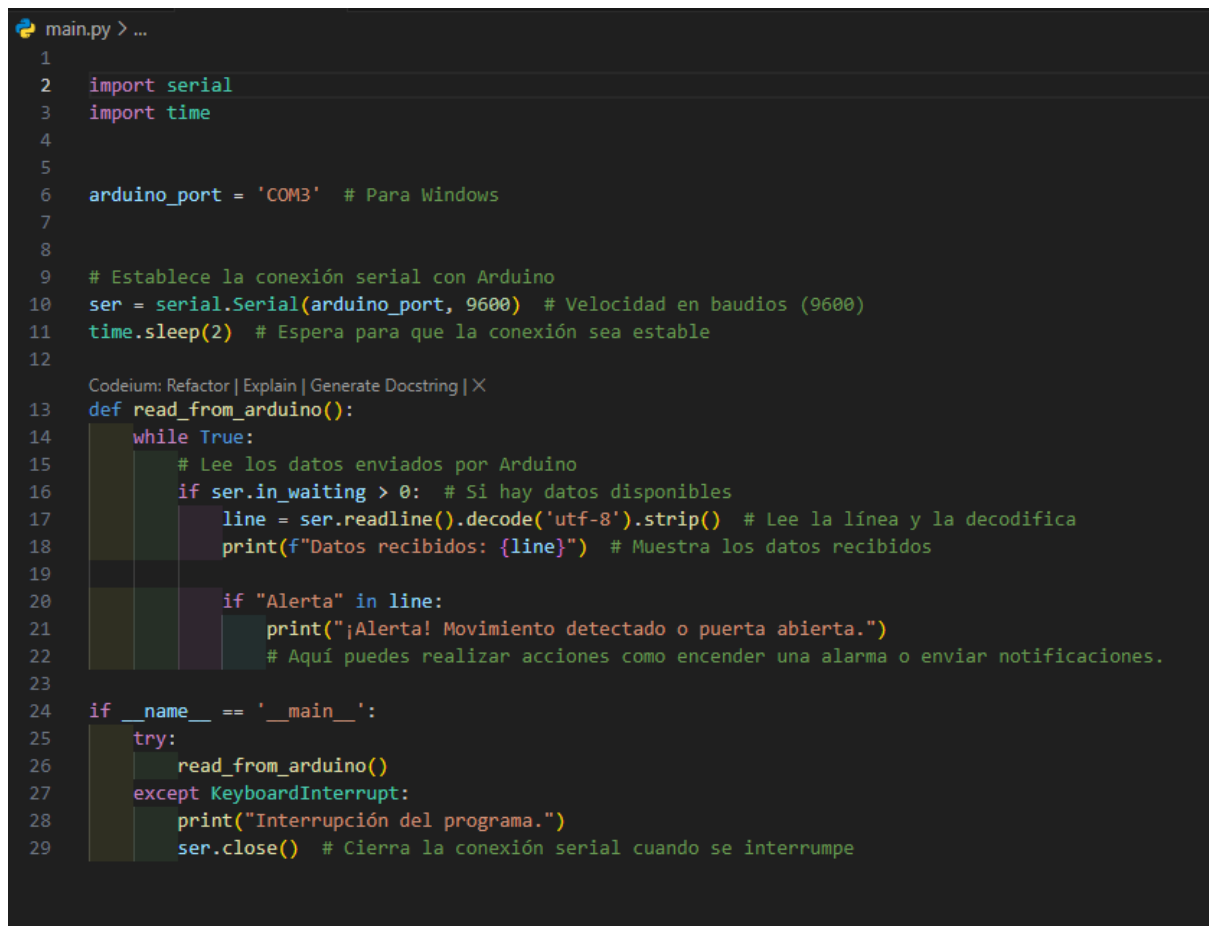
1.- Serial(arduino_port, 9600): Establece la conexión serial a través del puerto especificado con una tasa de 9600 baudios.

2.- Ser.readline(): Lee una línea de datos enviada por Arduino.

3.- Decode('utf-8'): Convierte los datos de bytes a un formato legible (cadena de texto).

4.-Ser.close(): Cierra la conexión serial cuando se interrumpe el programa.

Figura 55. Conexión entre Python y Arduino



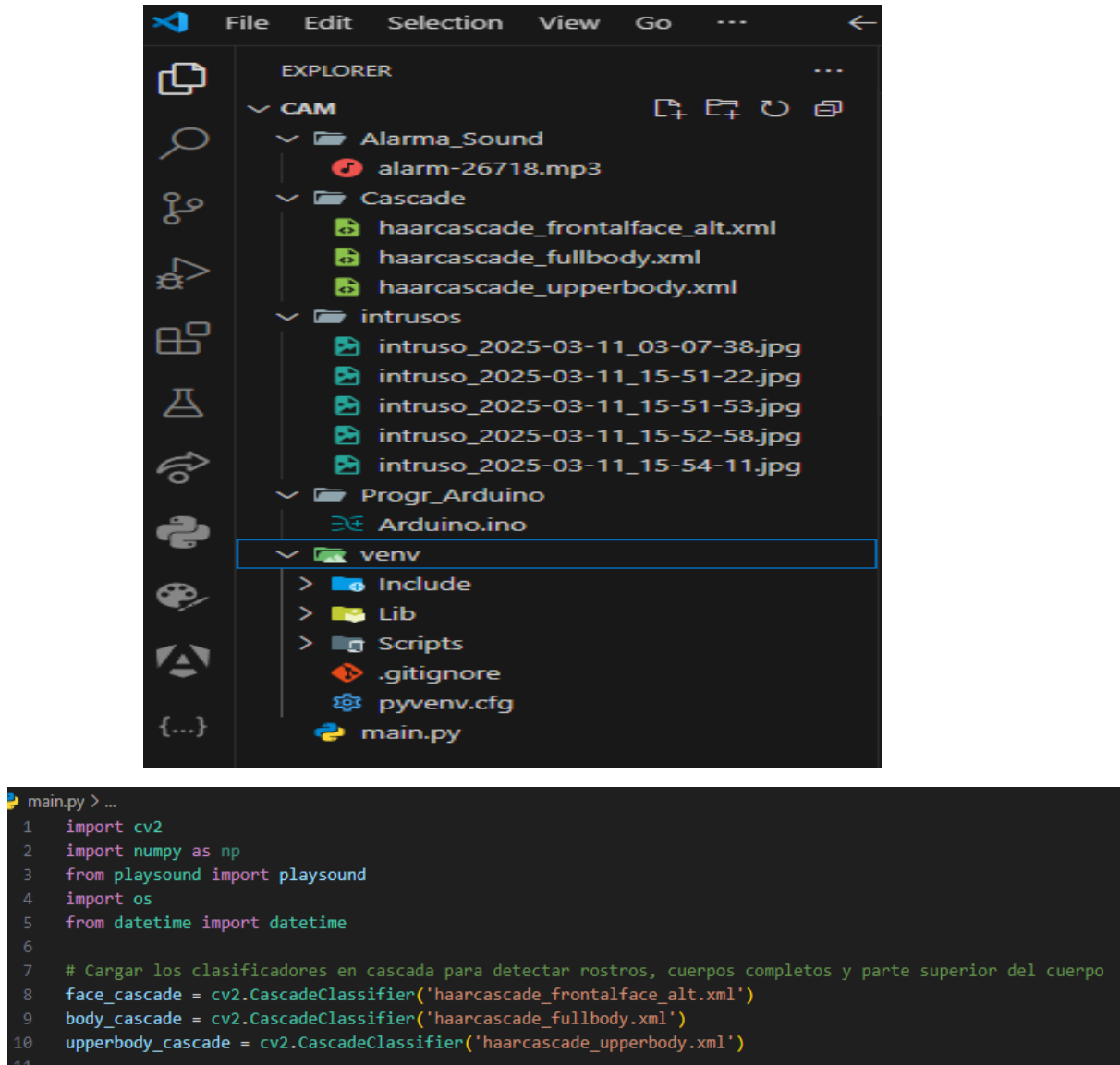
```
main.py > ...
1
2 import serial
3 import time
4
5
6 arduino_port = 'COM3' # Para Windows
7
8
9 # Establece la conexión serial con Arduino
10 ser = serial.Serial(arduino_port, 9600) # Velocidad en baudios (9600)
11 time.sleep(2) # Espera para que la conexión sea estable
12
Codeium: Refactor | Explain | Generate Docstring | X
13 def read_from_arduino():
14     while True:
15         # Lee los datos enviados por Arduino
16         if ser.in_waiting > 0: # Si hay datos disponibles
17             line = ser.readline().decode('utf-8').strip() # Lee la línea y la decodifica
18             print(f"Datos recibidos: {line}") # Muestra los datos recibidos
19
20         if "Alerta" in line:
21             print("¡Alerta! Movimiento detectado o puerta abierta.")
22             # Aquí puedes realizar acciones como encender una alarma o enviar notificaciones.
23
24 if __name__ == '__main__':
25     try:
26         read_from_arduino()
27     except KeyboardInterrupt:
28         print("Interrupción del programa.")
29         ser.close() # Cierra la conexión serial cuando se interrumpe
```

4.10. Despliegue del Aplicativo

Una vez que esta la programación completa del Arduino con sus componentes como son los sensores, alarmas pantalla de alertas y alarma, el siguiente paso es desplegar el proyecto a

Python para poder ejecutarlo para emprender el despliegue o ejecución del proyecto se empieza importando las instalaciones y emplear los clasificadores.

Figura 56. *Despliegue del Aplicativo*



En el campo Explorer están las carpetas en las cuales se encuentra distribuido el proyecto, en la carpeta Alarma_Sound está la alarma la cual será desplegada una vez los sensores capturen algo y le envíen la advertencia al Arduino y se activen, la Carpeta Cascade que están los indicadores, la carpeta Intrusos contiene lo que la cámara va capturando y enviando a la nube, la carpeta Progr_Arduino está la programación del Arduino, se está desplegando VENV para tener mejores resultados en las librerías y las importaciones, y el main.py el cual hace compilar el programa completo.

Figura 57. *Alarma Sound*

Adicional una función en la cual se despliegue un mensaje aparte ya que los primeros mensajes ingresan a los LED en este caso se imprime en la consola un mensaje de alerta igual capturando la hora y el fotograma exacto de la incursión.

Figura 58. *Mensaje de alerta*

```
# Función para mostrar el mensaje de alerta y guardar el fotograma
Codeium: Refactor | Explain | Generate Docstring | X
def termina(frame):
    print("Alerta!!! Alerta!!! Intruso se ha detectado un intruso")
    cap.release()
    cv2.destroyAllWindows()

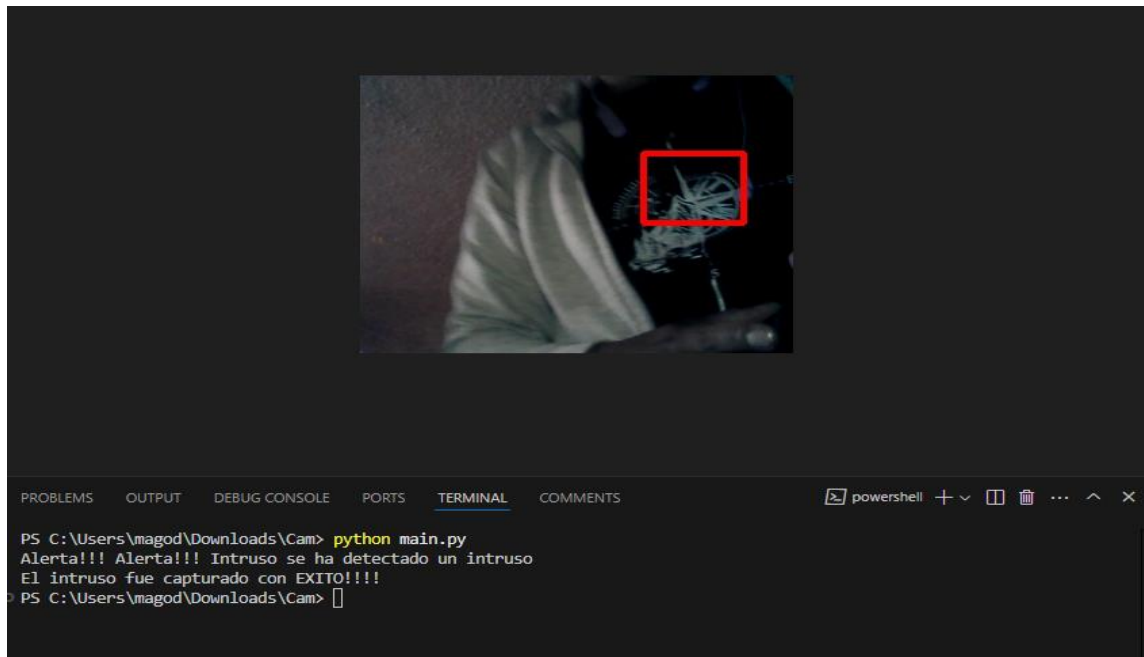
    # Guardar el fotograma como imagen con nombre basado en la hora actual
    timestamp = datetime.now().strftime('%Y-%m-%d_%H-%M-%S')
    filename = os.path.join(output_dir, f'intruso_{timestamp}.jpg')
    cv2.imwrite(filename, frame)
```

El mensaje proporcionado en la consola también para que se baya guardando en un cache y se pueda recuperar los datos si llegaran a perderse o dañarse.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  PORTS  TERMINAL  COMMENTS

PS C:\Users\magod\Downloads\Cam> python main.py
Alerta!!! Alerta!!! Intruso se ha detectado un intruso
█
```

Adicional el mensaje que se capturo al sospechoso con su respectiva captura.



4.11. Costo

En el contexto de los componentes en el mercado tienen un costo considerable en siendo que son un top de casi última tecnología, para este proyecto se usaron todos esos componentes siendo así un proyecto aplicable y renderizado, con costos que serán de ayuda para un futuro ya que sus componentes son buenos, estables y duraderos.

Tabla 4. Costos

Componente	Precio Unitario	Cantidad	Costo Total
Microcontrolador Arduino	\$25	1	\$25
Sensores de Movimiento PIR	\$7	4	\$28
Cámaras de Seguridad	\$45	2	\$90
Sensores de Puertas/Ventanas	\$2.50	4	\$10
Módulo de Comunicación Wi-Fi	\$8	1	\$8
Pantalla LCD o LED	\$6	1	\$6
Controlador de Acceso	\$15	1	\$15
Módulo RTC (DS3231)	\$10	1	\$10
Fuente de Alimentación	\$10	1	\$10
Total, Costos Iniciales			\$192

Consumo de Energía:

El Arduino y los sensores consumen muy poca energía, pero si estás utilizando cámaras de seguridad o un sistema de monitoreo en la nube, esto puede generar un costo de operación adicional. Aproximadamente \$2 - \$3 al mes si es solo Arduino y sensores (sin contar cámaras).

Mantenimiento:

Si el sistema no está conectado a la nube, el mantenimiento será mínimo.

Si se usa una cámara IP o se monitorea en la nube, habrá costos mensuales asociados a servicios de almacenamiento en la nube o mantenimiento de hardware.

Beneficios Directos

Seguridad Mejorada:

La implementación del sistema de detección de intrusos previene robos, daños y otros incidentes de seguridad, lo que podría evitar pérdidas económicas significativas a largo plazo.

En áreas con alta incidencia de robos o intrusiones, el sistema puede reducir sustancialmente los costos asociados a pérdidas materiales, reparación de daños o reemplazo de bienes robados.

Monitoreo Remoto:

La capacidad de monitorear remotamente mediante el uso de Wi-Fi y cámaras IP mejora la seguridad personal, ya que el propietario puede estar al tanto de cualquier incidente sin necesidad de estar presente físicamente.

Esto también puede reducir los costos asociados a la contratación de personal de seguridad físico.

Prevención de Daños Mayores:

Un sistema de detección de intrusos no solo ayuda a prevenir el robo de bienes, sino que también puede alertar sobre situaciones peligrosas, como incendios o intrusos potencialmente violentos, lo que puede prevenir daños mayores o lesiones.

Beneficios Indirectos

Valor de Propiedad:

Un sistema de seguridad incrementa el valor percibido de la propiedad. Las viviendas con sistemas de seguridad tienden a tener una mayor valoración en el mercado en caso de venta.

Descuentos en Seguros:

Algunas compañías de seguros ofrecen descuentos en primas a quienes tienen un sistema de seguridad instalado, lo que puede representar ahorros a largo plazo.

Mejor Tranquilidad y Bienestar:

La sensación de seguridad y el bienestar asociado a saber que tu hogar o empresa está protegido genera un beneficio intangible, pero importante, para la calidad de vida de las personas.

4.12. Análisis de Rentabilidad

Costo y Beneficio del sistema de detección de intrusos con Arduino muestra que, aunque el costo inicial y operativo puede ser moderado, los beneficios en términos de seguridad, ahorro en seguros, y el incremento en el valor de la propiedad hacen que este tipo de proyectos sea altamente rentable, especialmente cuando se considera la tranquilidad que aporta tener un sistema de seguridad fiable.

El sistema de seguridad domiciliario implementado utilizando Arduino ha demostrado ser una solución eficiente y económica para la protección del hogar. La integración de sensores de movimiento PIR, sensores de puertas/ventanas, cámaras de seguridad y alarmas proporciona una seguridad integral. Además, la capacidad de la comunicación Wi-Fi y la integración con Python permite un monitoreo remoto eficiente y el procesamiento de datos en tiempo real.

La implementación con Arduino ha permitido crear un sistema de seguridad fácil de personalizar y expandir. Al estar basado en un microcontrolador de bajo costo, el sistema puede adaptarse a las necesidades específicas de cada usuario. Además, la capacidad de incorporar más sensores y componentes según se requiera otorga flexibilidad y escalabilidad al proyecto.

La programación del Arduino con Python ha facilitado la creación de un sistema de control eficiente. Python permite procesar los datos de los sensores PIR y sensores de puertas/ventanas, de modo que, cuando se detecta una intrusión, el Arduino puede activar la cámara de seguridad para capturar fotogramas y activar la sirena de alarma. Esto asegura una respuesta rápida ante cualquier evento de seguridad.

Aunque el sistema cubre las necesidades básicas de seguridad, se recomienda expandir la cobertura del sistema agregando más cámaras y sensores en diferentes áreas de la propiedad. Esto permitirá una mayor seguridad integral en toda la casa.

Para mejorar la conectividad remota, se recomienda evaluar la posibilidad de mejorar la conexión Wi-Fi o integrar un sistema de notificaciones push a dispositivos móviles. Esto permitirá una alerta instantánea cuando se detecte cualquier intrusión o movimiento, mejorando la respuesta ante incidentes.

Para un funcionamiento autónomo y más sostenible, se podría considerar la incorporación de paneles solares para alimentar el sistema. Esto haría que el sistema de seguridad sea independiente de la red eléctrica y pueda funcionar de manera más eficiente a largo plazo.

Se recomienda implementar protocolos de encriptación más seguros para la comunicación remota, especialmente si el sistema está conectado a internet. Esto garantizaría que los datos del sistema estén protegidos contra posibles vulnerabilidades y ataques de hackers.

Conclusiones

En primer lugar, los resultados obtenidos de la implementación del sistema domótico basado en Arduino demuestran una notable eficacia en la automatización y el control de entornos residenciales. La integración de sensores de movimiento, módulos de comunicación y controladores ha permitido una gestión precisa de eventos, logrando una respuesta rápida y efectiva. Los datos recopilados reflejan que el sistema no solo cumple con los objetivos de automatización propuestos, sino que también mejora significativamente la seguridad y el confort del entorno, corroborando que la automatización puede optimizar la gestión del hogar de manera efectiva.

En segundo lugar, el análisis comparativo con investigaciones previas revela que el sistema domótico presenta avances importantes respecto a la fiabilidad y precisión en la detección y respuesta a eventos. A diferencia de sistemas anteriores que enfrentaban problemas de precisión y lentitud, el sistema propuesto ofrece una solución más robusta y eficiente. Este progreso resalta el valor de incorporar tecnologías modernas como el Arduino y el módulo Bluetooth HC-05 para superar las limitaciones de sistemas anteriores y lograr un rendimiento superior en aplicaciones residenciales.

No obstante, se han identificado algunas limitaciones que afectan la estabilidad del sistema, como la necesidad de recalibración frecuente de los sensores y la susceptibilidad del módulo Bluetooth a interferencias. Estos problemas subrayan la importancia de continuar con el desarrollo y la optimización del sistema. La búsqueda de tecnologías alternativas, como la comunicación Wi-Fi y métodos de calibración automática, podría ofrecer soluciones efectivas a estas limitaciones y mejorar aún más el desempeño del sistema en futuros desarrollos.

Por último, las implicaciones de estos resultados sugieren que el sistema domótico propuesto no solo ofrece una solución innovadora para la automatización del hogar, sino que también abre nuevas vías para investigaciones y desarrollos futuros. La integración de nuevas tecnologías y la mejora de los métodos actuales podrían superar las limitaciones identificadas, proporcionando un sistema aún más eficiente y fiable. Este enfoque no solo beneficiará la automatización del hogar, sino que también sentará las bases para futuras aplicaciones en el campo de la tecnología doméstica y la seguridad residencial.

Recomendaciones

Es recomendable continuar con la optimización del sistema domótico basado en Arduino, centrando los esfuerzos en la mejora de la calibración de los sensores y la estabilidad de la comunicación Bluetooth. La implementación de tecnologías alternativas, como la comunicación Wi-Fi o métodos de calibración automática, podría ayudar a resolver problemas de interferencia y mantenimiento, mejorando la fiabilidad y el rendimiento del sistema en aplicaciones residenciales.

También es aconsejable explorar la integración de nuevas tecnologías que puedan complementar y potenciar las capacidades del sistema. La incorporación de inteligencia artificial para la toma de decisiones automatizada y el aprendizaje de patrones de comportamiento puede aumentar la eficiencia y personalización del sistema, proporcionando un nivel superior de automatización y seguridad en el hogar.

Otra recomendación clave es realizar pruebas exhaustivas en diferentes entornos residenciales para evaluar la adaptabilidad del sistema a diversas configuraciones y condiciones. Esto permitirá identificar posibles mejoras y ajustar el sistema para satisfacer las necesidades específicas de cada usuario, garantizando que la solución sea efectiva y aplicable en una variedad de contextos.

Finalmente, es esencial seguir investigando y desarrollando nuevas aplicaciones para la tecnología domótica, considerando su potencial no solo en entornos residenciales, sino también en contextos comerciales e industriales. Ampliar el alcance de estas tecnologías puede generar avances significativos en la automatización y seguridad, ofreciendo soluciones innovadoras para diversos sectores y mejorando la calidad de vida en general.

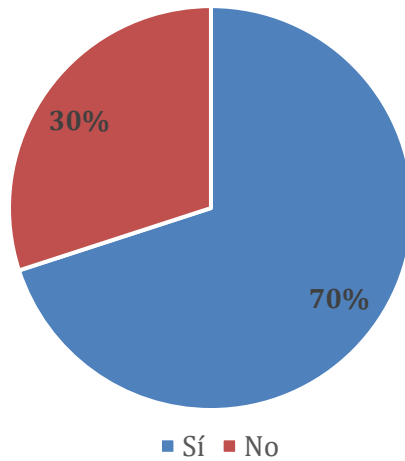
BIBLIOGRAFÍA

- Arce, V. C. (2023). *Diseño de un sistema domótica para una vivienda aplicando arduino*. Obtenido de https://laccei.org/LACCEI2023-BuenosAires/all-papers/Contribution_889_a.pdf
- Chico, A. J. (2022). *Sistema integrado modular para seguridad y confort de Viviendas Inteligentes basado en Internet*. Obtenido de <https://doi.org/10.33262/rmc.v7i4.2681>
- Gómez, S. G. (2022). *Sistema de seguridad con tecnología arduino para la automatización del edificio upocam*. Obtenido de <https://doi.org/10.47230/Journal.TechInnovation.v1.n2.2022.10-17>
- Gutiérrez, O. E. (2021). *“SISTEMA DE SEGURIDAD DOMICILIARIA BASADA EN TECNOLOGÍA ARDUINO Y APLICACIÓN MÓVIL”*. Obtenido de <https://repositorio.umsa.bo/bitstream/handle/123456789/7661/T.3103.pdf?sequence=1&isAllowed=y>
- Portilla, M. S. (2021). *Diseño de un sistema de seguridad inalámbrico para protección de viviendas y locales*. Obtenido de <https://repositorio.espe.edu.ec/bitstream/21000/25984/2/M-ESPEL-ENT-0242.pdf>
- Rodríguez, Y. (2018). *DESARROLLO DE UN SISTEMA DOMÓTICO BASADO EN TECNOLOGÍA ARDUINO*. Obtenido de https://www.researchgate.net/publication/325877967_DESARROLLO_DE_UN_SISTEMA_DOMOTICO_BASADO_EN_TECNOLOGIA_ARDUINO
- Salvador, R. A. (2022). *Sistema de seguridad con reconocimiento facial en módulo ESP32*. Obtenido de <https://orcid.org/0000-0002-2810-9599>
- Vega, A. M. (2021). *Internet de los objetos empleando arduino para la gestión eléctrica domiciliaria*. Obtenido de <https://www.redalyc.org/pdf/206/20633274004.pdf>
- Villegas, J. N. (2021). *Utilización de la placa de Arduino y SIM900 para controlar un prototipo de casa inteligente, utilizando un smartphone a través de mensajes de texto*.

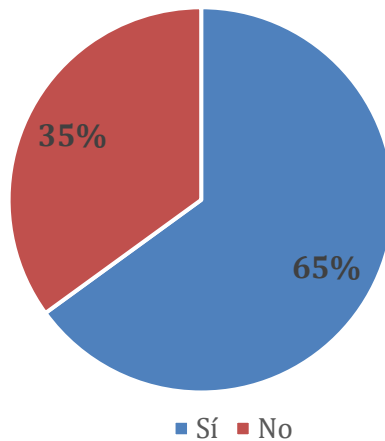
ANEXOS

Anexo 1. Encuesta

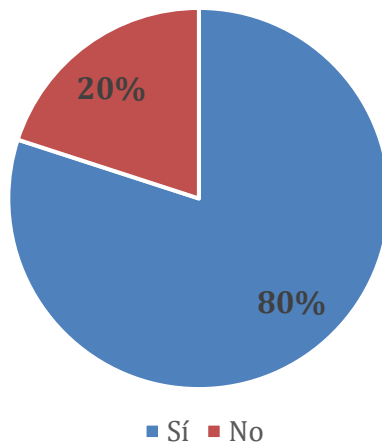
El sistema de seguridad basado en Arduino presenta mayor adaptación a las necesidades del hogar.



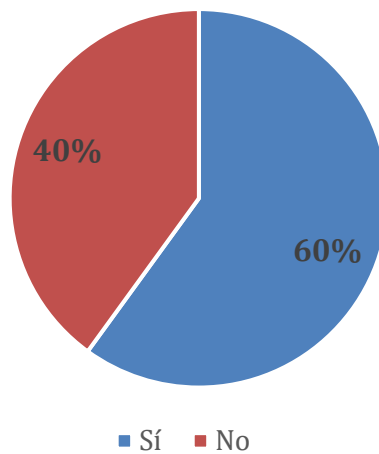
El sistema de seguridad basado en Arduino ofrece mayores utilidades que sistemas convencionales.



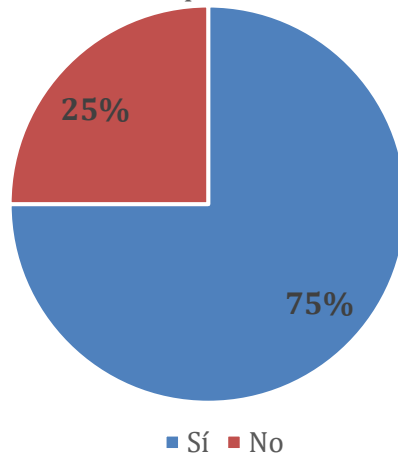
El sistema de seguridad basado en Arduino implementa opciones de protección más confiables.



El sistema de seguridad basado en Arduino usa eficientemente recursos técnicos y tecnológicos.



El sistema de seguridad basado en Arduino se ajusta a las necesidades de protección actualmente.



MANUAL TECNICO

CONFIGURACION E INSTALACION DE Implementación de un Sistema de Seguridad Domiciliario con Arduino

MANUAL TECNICO

1 INTRODUCCION

En la actualidad, la seguridad domiciliaria es una prioridad para muchas personas, ya que la protección de la familia, el hogar y las pertenencias personales es fundamental para mantener la tranquilidad y bienestar. El avance tecnológico ha permitido desarrollar soluciones más accesibles y personalizables para garantizar una protección efectiva, y entre estas soluciones, los sistemas de seguridad basados en tecnologías abiertas como Arduino han ganado popularidad debido a su bajo costo, flexibilidad y facilidad de personalización.

El Sistema de Seguridad Domiciliario con Arduino es una solución innovadora y económica para la protección de los hogares. Utilizando componentes básicos y herramientas disponibles en el mercado, como sensores de movimiento PIR, cámaras de seguridad, sensores de puertas y ventanas, pantallas LCD, y el microcontrolador Arduino, este sistema puede ser configurado para detectar intrusiones, activar alarmas y realizar un monitoreo remoto. Además, la comunicación vía Wi-Fi permite que el usuario controle y reciba notificaciones en tiempo real sobre el estado del sistema de seguridad desde cualquier lugar, brindando mayor seguridad y comodidad.

El presente manual técnico tiene como objetivo proporcionar una guía detallada de instalación y configuración para la implementación del Sistema de Seguridad Domiciliario con Arduino. Esta guía está diseñada para usuarios que deseen instalar, configurar y personalizar el sistema de acuerdo a sus necesidades, garantizando que todos los componentes funcionen correctamente de manera integrada.

1.1 Objetivo del Sistema de control

Objetivo principal de este proyecto es crear un sistema de seguridad autónomo, eficaz y de bajo costo que pueda ser utilizado en el hogar o pequeñas propiedades. Este sistema permite

detectar movimientos mediante sensores PIR, monitorear accesos a través de sensores de puertas y ventanas, visualizar imágenes mediante cámaras de seguridad, y activar alarmas sonoras para alertar de intrusiones. Todo esto es posible mediante un microcontrolador Arduino, que es el cerebro del sistema, y un software en Python para gestionar las comunicaciones y las respuestas automáticas.

Además, el sistema está diseñado para ser escalable y personalizable, permitiendo agregar más componentes en el futuro, como sensores adicionales, módulos de comunicación o incluso paneles solares para hacer que el sistema sea aún más sostenible.

1.2 Componentes del Sistema

El sistema está compuesto por diversos componentes electrónicos que trabajan de manera sincronizada para garantizar la seguridad total del hogar. Los principales componentes son los siguientes:

Microcontrolador Arduino: El corazón del sistema, encargado de controlar los componentes conectados y ejecutar el código de programación.

Sensores de Movimiento PIR: Detectan el movimiento en el área asignada y activan los dispositivos conectados.

Cámaras de Seguridad: Capturan fotogramas y videos cuando se detecta movimiento, proporcionándole al usuario información visual en tiempo real.

Sensores de Puertas/Ventanas: Se instalan en puertas y ventanas para detectar aperturas no autorizadas.

Módulo Wi-Fi ESP8266/ESP32: Permite la comunicación remota con el sistema, enviando notificaciones o alertas al usuario.

Pantalla LCD o LED: Muestra información sobre el estado del sistema, incluyendo si está activado o desactivado.

Controladores de Acceso (Lectores RFID): Permiten que solo personas autorizadas ingresen al área protegida.

Módulo RTC (DS3231): Controla el tiempo y fecha, lo que permite establecer horarios para activar o desactivar el sistema.

Fuente de Alimentación: Proporciona energía al sistema de manera estable.

1.3 Importancia del proyecto

El Sistema de Seguridad Domiciliario con Arduino no solo mejora la seguridad en el hogar, sino que también promueve la autonomía tecnológica. Al ser un sistema basado en código abierto, los usuarios tienen la oportunidad de aprender sobre electrónica, programación y automatización. Además, este sistema representa un modelo accesible para quienes buscan una solución económica para la seguridad de sus hogares, sin recurrir a costosos sistemas comerciales.

1.4 Alcance del Manual

Este manual proporcionará una descripción detallada de cada componente del sistema y sus funciones. Se incluirán instrucciones paso a paso para la instalación física de los dispositivos, así como la programación del microcontrolador Arduino y la configuración del sistema utilizando Python. El manual también abarcará la conexión de los sensores y actuadores a las placas de Arduino, las consideraciones sobre la alimentación eléctrica, y las posibles expansiones del sistema.

A lo largo de este manual, se brindarán detalles técnicos sobre los aspectos de cableado, instalación de librerías, código de programación, y configuración remota del sistema. Además, se incluirán recomendaciones para solucionar problemas comunes y mantener el sistema funcionando de manera óptima a largo plazo.

Este manual es una herramienta esencial tanto para principiantes como para expertos en el ámbito de la automatización del hogar y la seguridad electrónica, permitiendo a los usuarios comprender cada uno de los componentes involucrados y cómo interactúan entre sí para ofrecer una solución eficiente y efectiva.

2 CONFIGURACION DE HARDWARE

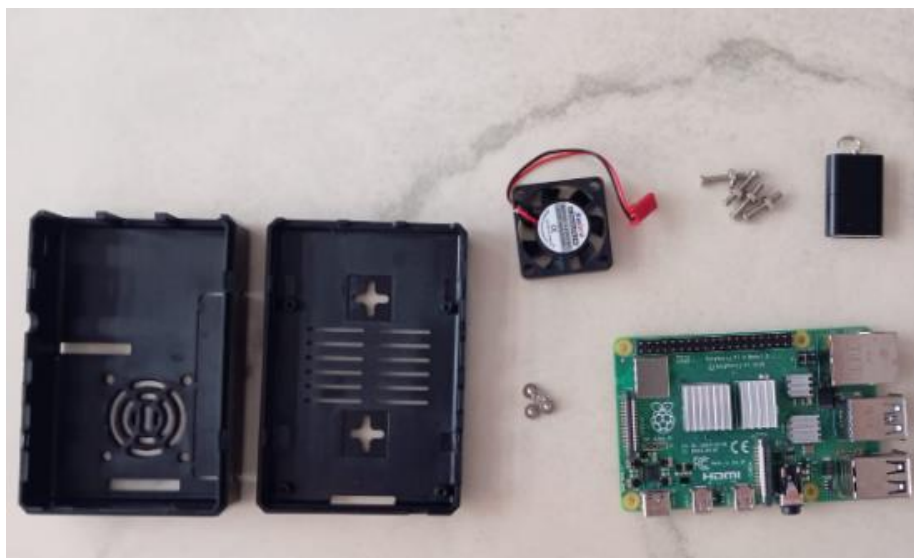
2.1 Configuración del Microcontrolador Arduino

Antes de conectar todos los componentes, es importante tener un **diagrama de conexión** para visualizar cómo se integran los componentes entre sí y con el **microcontrolador Arduino**. A continuación, se describen las conexiones de cada componente.

1.- **Microcontrolador Arduino (Arduino UNO/Nano):** El **microcontrolador Arduino** será el **corazón del sistema**, responsable de recibir las señales de los sensores y activar los dispositivos correspondientes. Conecta los pines de **alimentación** y **tierra** de todos los componentes a los pines **5V** y **GND** de Arduino.



4. Se instalan los componentes una vez que se tenga listo cada sensor PIR se conecta al **pin 5V** de Arduino.



3.

Para

cualquier duda o consulta relacionada con la implementación y el funcionamiento del sistema de seguridad domiciliario utilizando Arduino, se recomienda consultar la documentación oficial de Arduino. Esta documentación ofrece una amplia variedad de recursos, guías y ejemplos que pueden ser de gran ayuda para resolver problemas, entender mejor los componentes utilizados y explorar nuevas funcionalidades. Acceder a la documentación de Arduino es una excelente manera de profundizar en el conocimiento y optimizar el rendimiento de su proyecto.

DOCS

ARDUINO.CC

← All Docs

Hardware

MKR Family

Classic

Nano Family

Portenta Family

Pro Solutions and Kits

Nicla Family

Opta

Hardware

Browse through hundreds of tutorials, datasheets and guides for all Arduino products.

FOR MAKERS

MKR Family

Classic

Nano Family

Boards	Shields	Carriers
MKR 1000 WiFi	MKR 485 Shield	MKR Connector
MKR FOX 1200	MKR CAN Shield	Carrier MKR IoT
MKR GSM 1400	MKR ENV Shield	Carrier MKR IoT
MKR NB 1500	MKR ETH Shield	Carrier Rev2
MKR Vidor	MKR GPS	MKR

2.2 Cámaras de Seguridad

Las **cámaras de seguridad** deben estar conectadas al sistema de monitoreo a través de **Wi-Fi**. Si se usan cámaras IP, conecta la cámara a la red local Wi-Fi utilizando el **módulo ESP8266** o **ESP32**, que facilitará la transmisión de los fotogramas a través de una aplicación o servidor web.



3 INSTALACION A LA RED IP

3.1 Conexión a la red IP

Las cámaras IP se conectan a una red mediante protocolos de comunicación que permiten la transmisión de video digital a través de Internet. Estas cámaras funcionan como dispositivos autónomos con su propia dirección IP, facilitando su acceso y gestión desde cualquier lugar de la red. Además, las cámaras IP ofrecen ventajas significativas sobre las cámaras analógicas, como la capacidad de escalar el sistema sin límites en el número de cámaras, la calidad de Figura superior y la posibilidad de utilizar Power over Ethernet (PoE) para simplificar la instalación. La conexión a la red permite la integración con sistemas de grabación en red (NVR) y software de gestión de video (VMS), lo que facilita la monitorización y el almacenamiento de las grabaciones. Esto convierte a las cámaras IP en una opción ideal para sistemas de videovigilancia modernos y eficientes.



Cámaras de Seguridad: Las cámaras IP (como las de la marca **Nexxt Solutions AHIMPFI4U2 V3**) deben configurarse para que se conecten a la red **Wi-Fi** a través del **ESP8266** o **ESP32**. El microcontrolador envía las señales necesarias para activar la cámara y capturar imágenes o videos cuando se detecta movimiento o apertura de puertas/ventanas.

3.2 Configuración de los Sensores y Componentes

Sensores PIR: El sensor PIR es un dispositivo pasivo de infrarrojos que detecta el movimiento al medir las variaciones de calor en su entorno. Cuando detecta movimiento, su salida cambia a **ALTA (HIGH)**, lo que activa el sistema para hacer alguna acción, como encender una cámara o disparar una alarma. Puedes conectar varios sensores PIR a diferentes pines de entrada digital de Arduino, cada uno configurado para monitorear una sección diferente de la casa.

Sensores de Puertas/Ventanas: Estos sensores son más sencillos y generalmente consisten en dos partes: una parte magnética que se coloca en la puerta/ventana, y un interruptor de contacto que se conecta al microcontrolador. Cuando la puerta o ventana se abre y el contacto se rompe, el Arduino recibe una señal de ALTA y se activa un mecanismo como una alarma o el encendido de las cámaras de seguridad.

```
7
8 // Pines para el sensor PIR
9 const int pirPin = 8;
10
11 // Pines para el lector RFID
12 #define SS_PIN 10
13 #define RST_PIN 9
14 MFRC522 mfrc522(SS_PIN, RST_PIN); // Crear objeto RFID
15
16 // Pines para el teclado matricial (si se usa)
17 const byte ROW_NUM = 4; // Número de filas del teclado
18 const byte COLUMN_NUM = 4; // Número de columnas del teclado
19 char keys[ROW_NUM][COLUMN_NUM] = {
20     {'1','2','3','A'},
21     {'4','5','6','B'},
22     {'7','8','9','C'},
23     {'*','0','#','D'}
24 };
25 byte pin_rows[ROW_NUM] = {7, 6, 5, 4};
26 byte pin_column[COLUMN_NUM] = {3, 2, 1, 0};
27 Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_column, ROW_NUM, COLUMN_NUM);
28
29 // Pines para la sirena y los LEDs
30 const int speakerPin = 9;
31 const int ledRedPin = 10;
32 const int ledBluePin = 11;
33
34 // Pines para el RTC
35 RTC_DS3231 rtc;
36 LiquidCrystal_I2C lcd(0x27, 16, 2); // Configurar LCD con I2C
37 |
```

Anexo 3 Manual de usuario

MANUAL DE USUARIO

CONFIGURACION E INSTALACION DE Implementación de un Sistema de Seguridad Domiciliario con Arduino

MANUAL DE USUARIO

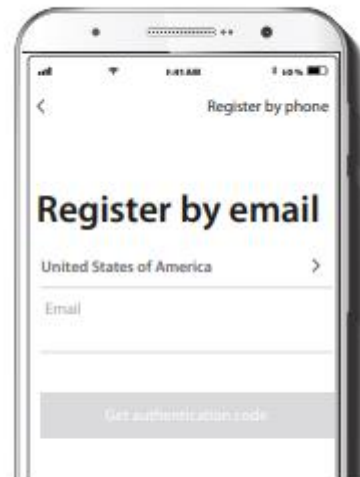
1. Como primer paso es necesario descargar el software ya sea por play store o app store y registrarse

Instalación de la aplicación Nexxt Solutions Home App



Descargue directamente la aplicación
Nexxt Home de Google Play o de Apple App Store.

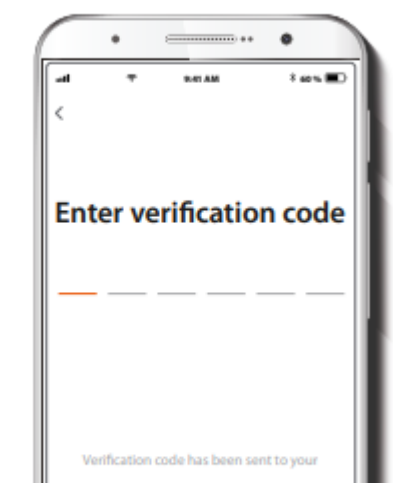
Paso 1



Registro por email. Si desea registrarse por teléfono (EE. UU. solamente), seleccione **Registro por teléfono** en el borde superior derecho de la pantalla.

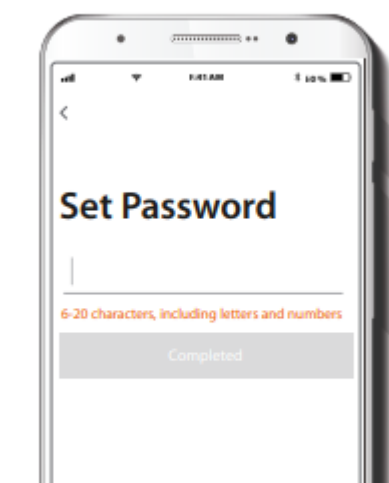
2. Al registrar el correo electrónico llega un numero con el cual se puede ingresar en este paso y se añade una nueva contraseña.

Paso 2



Ingrese el código de verificación que recibió.

Paso 3



Cree una contraseña nueva y seleccione **Completo** a continuación.

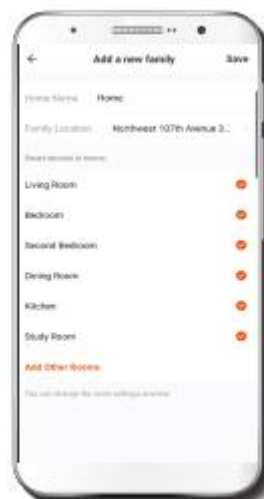
3. Aquí ya con la aplicación activa se crea la cuenta y se escanea a buscar la cámara de seguridad mediante la IP o WI-FI.

Paso 4



La aplicación Nexxt Solutions: Home ha sido instalada satisfactoriamente.

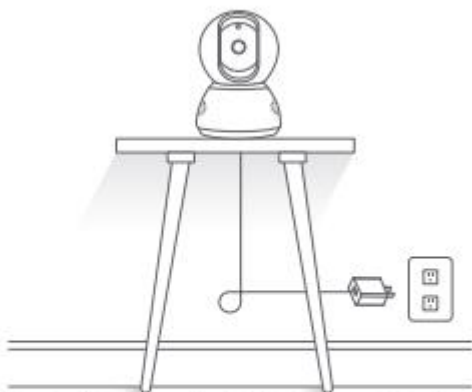
Paso 5



Asigne un **Nombre** a la casa y la **Ubicación** del dispositivo en su interior. Seleccione **Agregar** otra habitación e identifique los puntos de instalación respectivos. Seleccione **Guardar** para almacenar sus parámetros.

4.- Se enchufa en este caso se conecta donde se busque tener puntos ciegos y tener acceso en tiempo real se conecta y después se agrega mediante un código QR.

Conexión



Conecte la cámara con el adaptador de CA suministrado. Después de 30 segundos, la cámara emite el primer tono para indicar que se ha iniciado correctamente. 10 segundos después, se genera un segundo tono para indicar que la cámara está lista para emparejarse. De lo contrario, mantenga presionado el botón de reposición por 8 segundos para reiniciar el dispositivo.

Agregar dispositivo

Paso 1



Seleccione **Agregar dispositivo** y a continuación, la cámara que desea instalar.

5.- Se escanea el QR desde la misma cámara se espera un rato y llega la notificación que fue agregada con éxito, luego se ingresa un nombre a la cámara.

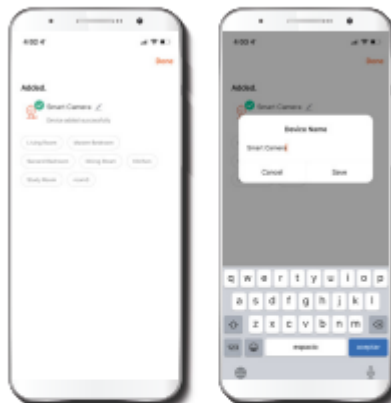
Paso 4



Seleccione **Continuar** para obtener el código QR. Escanee con la cámara inteligente el código QR desplegado en su dispositivo móvil. Una vez que la cámara emita el tono, haga clic en el botón **Escuché la señal**.

De experimentar algún problema, vuelva a iniciar la cámara. Si una **Cuenta registrada** agrega la cámara a través de la aplicación, ningún otro usuario podrá incluir esa cámara en una cuenta, a menos que se le otorgue expresa autorización a través de la opción **Compartir acceso**.

Paso 5



Cuando concluye el proceso de emparejamiento, aparece desplegada la notificación **Dispositivo incorporado exitosamente** en la pantalla del dispositivo.

Pulse el ícono del lápiz con el fin de modificar el nombre de la cámara. Seleccione **Guardar** a continuación para almacenar su configuración.

6.- Se configura la cámara ya sea en calidad en fotos en modo nocturno en almacenamiento en sonido y sobre todo en modo nocturno en calidad de Figura.

Configuraciones de la cámara



- Pantalla completa:** Cambia al modo de visualización a pantalla completa.
- Captura de pantalla:** Toma una foto de la imagen que está siendo captada por la cámara.
- Hablar:** Permite la comunicación por voz a través del micrófono de la cámara.
- Grabar:** Inicia la grabación de un video.
- Reproducir:** Si instala una tarjeta microSD™, puede utilizar la función de reproducción para ver videos.
- Galería:** Despliega una lista de todos los videos y fotos almacenados localmente o en la nube.
- Esquema de colores:** Permite a los usuarios cambiar la apariencia de la interfaz entre el modo claro y el modo oscuro según sus preferencias; esta última regulación es la que se recomienda para conservar energía de la batería de su dispositivo móvil.

Y una vez seguidos todos los pasos y cumplido con lo que se debe hacer tenemos el resultado final el cual es navegar en tiempo real y tener fotogramas exactos, tanto con el tiempo el día y la hora y un almacenamiento mediante micro sd o a su vez desde la ip a la nube y así poder tener nuestra seguridad siempre a la mano.

