



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

**CARRERA DE INGENIERIA ELECTRÓNICA CON MENCIÓN EN
TELECOMUNICACIONES.**

**“DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DIDÁCTICO ORIENTADO A
SEGURIDAD ELECTRÓNICA EN DATA CENTER BASADO EN RASPBERRY PI Y
ARDUINO PARA PRÁCTICAS EN EL LABORATORIO DE
TELECOMUNICACIONES”**

Trabajo de titulación Previa la obtención del título de:

Ingeniero Electrónico con mención en telecomunicaciones.

AUTORES:

Henry Yordano Zamora Mendoza

Andrés Stalin Vera Cando

TUTOR: PhD. Lenin Estuardo Cevallos Robalino

Guayaquil – Ecuador

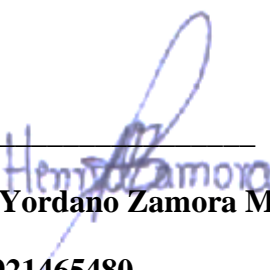
2021

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Henry Yordano Zamora Mendoza con documento de identificación N° 0921465480 y Andrés Stalin Vera Cando con documento de identificación N° 0930767546; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 07 de septiembre del año 2021.


Henry Yordano Zamora Mendoza.
C.I.: 0921465480


Andrés Stalin Vera Cando.
C.I.: 0930767546

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL
TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA
SALESIANA.**

Nosotros, Henry Yordano Zamora Mendoza con documento de identificación No. 0921465480 y Andrés Stalin Vera Cando con documento de identificación No. 0930767546, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación: **“Diseño e implementación de un módulo didáctico orientado a seguridad electrónica en data center basado en Raspberry Pi y Arduino para prácticas en laboratorio de telecomunicaciones”**, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 07 de septiembre del año 2021.



Henry Yordano Zamora Mendoza.

C.I.: 0921465480



Andrés Stalin Vera Cando.

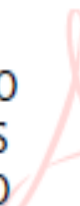
C.I.: 0930767546

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Lenin Estuardo Cevallos Robalino con documento de identificación N° 0603274226, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **“DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DIDÁCTICO ORIENTADO A SEGURIDAD ELECTRÓNICA EN DATA CENTER BASADO EN RASPBERRY PI Y ARDUINO PARA PRÁCTICAS EN LABORATORIO DE TELECOMUNICACIONES”**, realizado por Henry Yordano Zamora Mendoza con documento de identificación N° 0921465480 y Andrés Stalin Vera Cando con documento de identificación N° 0930767546, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 09 de septiembre del año 2021.

LENIN
ESTUARDO
CEVALLOS
ROBALINO



Firmado digitalmente
por LENIN ESTUARDO
CEVALLOS ROBALINO
Fecha: 2021.09.08
14:45:23 +02'00'

PhD. Lenin Estuardo Cevallos Robalino.

C.I.: 0603274226

Dedicatoria

Dedico este proyecto de titulación a mi familia, por ser el pilar fundamental en mi formación personal y profesional, brindándome todo su amor y apoyo incondicional a lo largo de mis estudios.

Se lo dedico a mi madre Msc. Sandra Soraya Mendoza Mendoza por enseñarme los valores y la importancia del estudio, del trabajo duro y del esfuerzo, a nunca rendirme frente a los momentos difíciles y confiar plenamente en mis habilidades.

A mi padre, Enrique Ramón Zamora Decimavilla, por enseñarme con su ejemplo lo que es el esfuerzo y el trabajo; a mi hermana, Odalis Soraya Zamora Mendoza, por siempre apoyarme, por su paciencia y amor.

Henry Zamora Mendoza.

Dedicatoria

Dedico este proyecto de titulacion a Dios y a mi familia, ya que sin el apoyo de ellos no hubiera logrado seguir avante en lo largo de mis estudios, han estado en momentos buenos y malos, siempre aconsejandome en luchar por esta meta, que se esta logrando, gracias a ellos.

En especial esta dedicatoria es para mi madre Lida Cando Saigua, que siempre me guio, me dio fuerzas, a enseñarme lo valioso y primordial en la vida que son los estudios.

Andrés Vera Cando.

Agradecimiento

Agradezco a Dios por guiarme en mi camino, darme salud y por permitirme cumplir mis objetivos universitarios y profesionales.

Agradezco a mis padres por siempre creer en mí, por impulsarme a cumplir mis metas y brindarme su apoyo incondicionalmente.

Agradezco a mis tías por brindarme hospedaje y todo su amor mientras cursaba mis estudios de bachillerato y universitarios.

¡Gracias por creer y confiar en mí!

Henry Zamora Mendoza.

Agradecimiento

Le agradezco en primer lugar a Dios por darme la oportunidad de cumplir una de mis metas, a mi familia por siempre apoyarme en cada momento que los necesite, en especial a mi madre que siempre me aconsejaba en seguir adelante y no desmayar, en seguir luchando todos estos años hasta poder lograr el tan anhelado título profesional, a los profesores que supieron guiarnos con sus enseñanzas en cada año.

Andrés Vera Cando.

Tabla de contenido

Dedicatoria	V
Dedicatoria	VI
Agradecimiento	VII
Agradecimiento	VIII
Resumen	XX
Abstract	XXI
Introducción	1
1. Capítulo I	2
1.1. Planteamiento del problema	2
1.2. Antecedentes	2
1.3. Importancia y Alcance	2
1.4. Delimitación del problema	3
1.4.1. Delimitación temporal	3
1.4.2. Delimitación espacial	3
1.4.3. Delimitación académica	4
1.5. Objetivos	4
1.5.1. Objetivo general	4
1.5.2. Objetivos específicos	4
Capítulo II	5
2. Marco Teórico	5
2.1. Data Center	5
2.2. Seguridad electrónica	5
2.2.1. Elementos que constituyen un sistema de seguridad(2018)	6

2.3. Estándar de seguridad para un Data Center.	7
2.3.1. Estándar TIA 942 (EIA, 2005).....	7
2.4. Norma ISO / IEC 27001 (NQA, 2017).	9
2.5. Estudio del Hardware.....	10
2.5.1. SingleBoardComputers.	10
2.5.2. Arduino.	10
2.5.2.1. Arduino Uno.	12
2.5.2.1.1. Componentes de Arduino Uno (Artero, 2013). Entre los componentes encontrados en esta PCB se detallan a continuación:	13
2.5.3. Raspberry Pi.....	14
2.5.3.1. Diseño de la Raspberry Pi (Patil, 2017):.....	15
2.5.3.2. Raspberry Pi 4(2019).	15
2.6. Circuito Cerrado de Televisión	17
2.7. Cámaras IP	17
2.8. Sensor de temperatura y humedad.	17
2.9. Sensor de movimiento PIR HC-SR501.....	17
2.10. Sistema RFID (Radio FrequencyIdentification).	18
2.11. Biometría.....	18
2.11.1. Reconocimiento de huella dactilar.	19
2.11.2. Acceso con tarjeta Magnéticas(VARGAS VERGARA, 2013).	19
2.12. Estudio del Software.	19
2.12.1. Lenguaje de programación.....	19
2.12.1.1. Lenguaje de programación Python.....	20
2.12.1.1.1. Sintaxis de Python.....	22
2.13. Escritorio remoto.....	22

2.13.1. Protocolo VNC (Virtual Network Computing).....	22
2.13.2. Protocolo SSH (Secure Shell) (Gardoki, n.d.-b).....	23
2.14. Protocolo SMTP (Simple Mail Transfer Protocol).....	24
2.14.1. Estructura de SMTP (Riabov, 2011).....	25
2.15. Protocolo FTP.....	26
2.15.1. Configuración del servicio de transferencia de archivos, permisos y cuotas, conexiones en modo activo(Sánchez et al., 2015).	27
2.16. Tecnología GSM.....	27
2.17. Tecnología GPRS.....	29
Capitulo III.....	30
3. Marco metodológico.....	30
3.1. Método de Investigación.....	30
3.1.1. Método experimental(Rodríguez, 2011).	31
3.2. Descripción de sensores en el estudio.....	32
3.2.1. Cámaras de seguridad.....	32
3.2.2. Sensor de Temperatura y Humedad DTH 11(ELECTRONICS, 1995).	33
3.2.3. Sensor de Movimiento PIR HC -SR501(ELECTRONILAB, 2011).	35
3.2.3.1. Radio de inducción.....	36
3.2.3.2. Aplicaciones.....	36
3.2.4. Control de acceso con RFID.	36
3.2.1. Control de acceso con Lector de huella digitalRC305.....	38
3.3. Descripción de sistemas Actuadores.....	40
3.3.1. Sistemas GSM/GPRS(Federico Nicola, 2015).	40
3.3.2. Módulo SIM800L GSM/GPRS(López Eusebio, Victor Ernesto Gutiérrez Torres, 2017).....	40

3.3.2.1. Características del módulo SIM 800L	41
3.3.2.2. Componentes. El módulo contiene un total de 12 pines que lo conectan al mundo exterior:	41
3.3.3. BUZZER(Alexander et al., 2012).	42
3.3.4. Módulo WIFI ESP8266(Ochoa, 2017).	42
3.3.5. Módulo RELE(Bolaños, 2016).	43
3.3.6. Cerradura Eléctrica(Coba Gibbs & Samaniego González, 2017).	44
3.3.7. Pantalla LCD 16X2 1602A.	44
3.4. Descripción de sistema de procesamiento.....	45
3.4.1. Microcontrolador Arduino.	45
3.4.1.1. Funcionamiento.....	45
3.4.1.2. Conexión de Arduino uno con Módulo RFID RFC522.	47
3.4.1.3. Conexión de Arduino uno con Módulo Sim800L.....	48
3.4.2. Sistema de Seguridad electrónica aplicando IOT con una placa de Arduino un módulo RFID y un módulo Sim800L.	49
3.4.2.1. Aplicación de Arduino en control de acceso.....	50
3.4.3. Microcontrolador Raspberry Pi modelo 4B.	51
3.4.3.1. Funcionamiento.....	53
3.4.3.2. Conexión de Raspberry Pi 4B con una microcámara Full HD.	54
3.4.3.3. Conexión de Raspberry Pi 4B con dispositivo sensor de movimiento.	59
3.4.4. Conexión de Raspberry Pi 4B con dispositivo sensor de temperatura y humedad.....	62
3.4.5. <i>Raspberry Pi 4B como servidor de correos electrónicos.</i>	65
3.4.6. Aplicación de sistema de seguridad electrónica con microcontrolador Raspberry Pi en implementación IOT.	67
3.5. Implementación.....	69

Capítulo IV	71
4. Análisis de resultados.	71
4.1. Módulo de prácticas de seguridad electrónica en Data Center	71
4.2. Listado de prácticas.....	72
4.3. Resultados de las prácticas.....	73
1. Conclusiones.....	76
2. Recomendaciones	77
3. Referencias bibliográficas.....	78
4. Anexos	83

Índice de Tabla

Tabla 1.....	5
Tabla 2.....	7
Tabla 3.....	11
Tabla 4.....	14
Tabla 5.....	28
Tabla 6.....	37

Índice de Figura

Figura 1. Ubicación Universidad Salesiana, tomada de Google Maps.....	3
Figura 2. Bloques funcionales que componen un sistema de seguridad electrónica, tomado del libro “Circuito cerrado de televisión y seguridad electrónica”.	6
Figura 3. Modelo PHVA para ISO / IEC 27001, tomada de NQA- Guía de implantación para la seguridad de la Información.	9
Figura 4. Placa de Circuito Impreso Arduino Uno, tomado de www.arduino.cc	13
Figura 5. Placa de Modelo Impreso Raspberry Pi, tomado de www.raspberrypi.org	14
Figura 6. Raspberry Pi 4 modelo B, tomado de https://datasheets.raspberrypi.org/	16
Figura 7. Historia de los primeros lenguajes de programación, adaptado de “Historia y evolución de los lenguajes de programación”, elaborado por autores.	20
Figura 8. Funcionamiento del programa intérprete, tomado de “Aprende a pensar como un programador”(2000).	21
Figura 9. Funcionamiento del programa compilador, tomado de “Aprende a pensar como un programador”(2000).	22
Figura 10. Gráfico explicativo sobre el funcionamiento del protocolo VNC, tomado de sitio web https://www.serdarbayram.net	23
Figura 11. Gráfico explicativo sobre el funcionamiento del protocolo SSH, tomado de sitio web https://pressroom.hostalia.com	24
Figura 12. Proceso de envío de un correo desde el usuario uno al usuario dos a través del servidor SMTP, tomado del sitio web https://ingsistemasunicor.files.wordpress.com/	26
Figura 13. Proceso realizado por protocolo FTP para la transferencia de archivos desde ordenador cliente – servidor, tomado de sitio web https://neo.lcc.uma.es/	27
Figura 14. Arquitectura de la tecnología GPRS, tomada de sitio web https://es.slideshare.net/	30
Figura 15. Diagrama de bloques del sistema de seguridad electrónica del módulo. Elaborado por autores.	32
Figura 16. Módulo de cámara OV5647. Elaborado por autores.	33

Figura 17. Modelos de sensor de temperatura y humedad DTH22 y DTH11. Elaborado por autores.	34
Figura 18. Diagrama de conexión de un sensor de temperatura DHT11 modelo estándar. Tomado de (ELECTRONICS, 1995).....	34
Figura 19. Módulo de sensor PIR modelo HC-SR501. autores	35
Figura 20. Radio de captación de infrarrojo del módulo HC-SR501. Tomado de (ELECTRONILAB, 2011).	36
Figura 21. Tag transportador RFID, llavero de acceso y lector transceptor RFID. Autores	37
Figura 22. Módulo de identificación de huella digital RC305. Elaborado por autores	39
Figura 23. Conexión de Arduino uno y módulo de identificación de huella digital RC305. Elaborado por autores.	39
Figura 24. Módulo GSM GPRS SIM800L. Elaborado por autores.	41
Figura 25. Módulo Buzzer Piezoeléctricos. Elaborado por autores.....	42
Figura 26. Módulo wifi ESP8266. Elaborado por autores.	42
Figura 27. Módulo RELE. Elaborado por autores.	43
Figura 28. Cerradura eléctrica. Elaborado por autores.	44
Figura 29. Pantalla LCD 16x2. Elaborado por autores.	44
Figura 30. Pantalla IDE (Entorno de desarrollo integrado) Arduino. Elaborado por autores.	46
Figura 31. Placa de Arduino Uno. Elaborado por autores.	46
Figura 32. Conexión de Arduino y tarjeta RFID. Elaborado por autores.	47
Figura 33. Conexión de Arduino y módulo Sim800L. Elaborado por autores.	49
Figura 34. Ejecución de sistema de seguridad de control de acceso con alarmas mediante buzzer y módulo Sim800L en una protoboard. Elaborado por autores.....	50
Figura 35. Diagrama de bloque funcionamiento de control de acceso con microcontrolador Arduino. Elaborado por autores.	51
Figura 36. Placa de Raspberry PI Modelo 4B. Elaborado por autores.	52

Figura 37. GPIO Raspberry Pi 4 modelo B, tomado de https://www.raspberrypi.org/documentation/usage/gpio/	52
Figura 38. Descarga de Sistema Operativo de página oficial Raspberry Pi. Elaborado por autores.	53
Figura 39. Habilitación de interfaz SSH y VNC en IDE Raspberry Pi. Elaborado por autores.	54
Figura 40. Conexión física de CámaraOV5647 y Raspberry Pi. Elaborado por autores.	54
Figura 41. Habilitación de la cámara en IDE Raspberry Pi. Elaborado por autores..	55
Figura 42. Código Fuente para capturar imágenes con Raspberry Pi. Elaborado por autores.	56
Figura 43. Imagen captada con cámara HD y microcontrolador Raspberry Pi. Elaborado por autores.	57
Figura 44. Código Fuente para grabar videos con Raspberry Pi. Elaborado por autores.	57
Figura 45. Video de 30 segundos captada con cámara HD y microcontrolador Raspberry Pi. Elaborado por autores.....	58
Figura 46. Conexión de aplicación RaspiCamRemote con cámara OV5647. Elaborado por autores.	59
Figura 47. Sensor PIR HC-SR501, pines GND – SEÑAL- VCC. Elaborado por autores.	59
Figura 48. Conexión física de sensor PIR HC – SR501 y Raspberry Pi. Elaborado por autores.	60
Figura 49. Sensor PIR HC-SR501 código de programación. Elaborado por autores.	61
Figura 50. Resultado de detección de movimiento en Raspberry Pi. Elaborado por autores.	62
Figura 51. Sensor de temperatura y humedad DHT22(blanco) – DHT11(azul). Elaborado por autores.	63
Figura 52. Conexión física del sensor de temperatura y humedad DHT11 y Raspberry Pi. Elaborado por autores.	64
Figura 53. Sensor de temperatura y humedad DHT11 código de programación. Elaborado por autores.	65

Figura 54. Resultado de temperatura y humedad en Raspberry Pi. Elaborado por autores.	65
Figura 55. Raspberry Pi como servidor de correo. Elaborado por autores.	67
Figura 56. Ejecución de sistema de seguridad electrónica mediante microcontrolador Raspberry Pi con alarmas mediante buzzer, correo electrónico y módulo Sim800L en una protoboard. Elaborado por autores.	68
Figura 57. Dibujo AutoCAD módulo de seguridad electrónica. Elaborado por autores.	69
Figura 58. Renderizado de módulo de prácticas de seguridad electrónica. Elaborado por autores.	70
Figura 59. Módulo didáctico de prácticas de seguridad electrónica. Elaborado por autores.	71
Figura 60. Dispositivos electrónicos para utilizar. Elaborado por autores.	72

Índice de Anexos

Anexo 1. Práctica #1 realizada para programación de Raspberry Pi mediante el sistema operativo Raspbian.....	83
Anexo 2. Práctica desarrollada para la programación de Raspberry Pi como servidor FTP y SMTP.	98
Anexo 3. Práctica realizada para la configuración y conexión de un sistema de circuito cerrado de televisión mediante Raspberry Pi.	120
Anexo 4. Práctica realizada para el control de acceso mediante uso de microcontrolador Arduino.....	139
Anexo 5. Práctica desarrollada para la configuración y conexión de un sensor de humedad y temperatura.	167
Anexo 6. Práctica realizada para la configuración y conexión de un sensor de movimiento mediante Raspberry Pi.	184
Anexo 7. Práctica realizada para la configuración de detector de movimiento en CCTV mediante Raspberry Pi.	199
Anexo 8. Práctica desarrollada para el control de acceso mediante uso de microcontrolador Arduino con módulo SIM800L GPRS/GSM	217
Anexo 9. Práctica realizada para la configuración de alertas mediante “Guardo de capturas”.....	245
Anexo 10. Práctica realizada para la configuración de alertas mediante “envío de correos”.....	265

Resumen

AÑO	ESTUDIANTES	DIRECTOR DEL PROYECTO	TEMA DEL PROYECTO
2021	<ul style="list-style-type: none"> • HENRY YORDANO ZAMORA MENDOZA. • ANDRÉS STALIN VERA CANDO. 	PhD. LENIN CEVALLOS ROBALINO	DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DIDÁCTICO ORIENTADO A SEGURIDAD ELECTRÓNICA EN DATA CENTER BASADO EN RASPBERRY PI Y ARDUINO PARA PRÁCTICAS EN EL LABORATORIO DE TELECOMUNICACIONES

El presente módulo didáctico acopla varios sistemas de sensores en un tablero que brinda la apertura al conocimiento práctico de los estudiantes de Ingeniería Electrónica en la Universidad Politécnica Salesiana.

Los sensores de monitoreo, vigilancia, temperatura y humedad, control de acceso, son controlados mediante miniordenadores tales como Raspberry Pi modelo 4 y Arduino Uno. El diseño y la implementación del proyecto de titulación se basa en tres niveles de ejecución, en primer lugar, se tomará mediciones con los sensores interconectados al miniordenador correspondiente, como segundo paso, se almacenará, procesará y se modificará según las preferencias del usuario y se obtendrá el resultado predispuesto según el sistema efectuado.

Cada práctica presentará como resultado un estado de alarma al estudiante programador, las cuales se podrán representar como alarma visual mediante el encendido de luces led, alarma auditiva a través del encendido de un zumbador electrónico el cual convierte la señal eléctrica en salida sensorial y la alarma inalámbrica la cual presenta mensajes de alerta utilizando dos sistemas tales como mensajería de texto (tecnología GSM) y correos electrónicos (Tecnología SMTP).

Los sistemas de seguridad electrónica presentados en el módulo didáctico cuentan con diagramas de conexiones para facilitar la ejecución de las prácticas y aumentar la interpretación de los estudiantes, los cuales estimularan su comprensión en la lectura de manuales técnicos electrónicos anexados en el presente trabajo de investigación. Así el presente proyecto de investigación tiene como resultado una simulación sistema de seguridad que genera una solución que permite ejecutar la supervisión del Data Center, incentivando al estudiante a desarrollar conocimientos teóricos y prácticos de este.

Palabras Claves: Data Center, sensores, miniordenadores, ingeniería.

Abstract

YEAR	STUDENTS	PROJECT DIRECTOR	PROJECT THEME
2021	<ul style="list-style-type: none"> • HENRY YORDANO ZAMORA MENDOZA. • ANDRÉS STALIN VERA CANDO. 	PhD. LENIN CEVALLOS ROBALINO	DESIGN AND IMPLEMENTATION OF A DIDACTIC MODULE ORIENTED TO ELECTRONIC SECURITY IN DATA CENTER BASED ON RASPEBRRY PI AND ARDUINO FOR PRACTICES IN LOBORATORY OF TELECOMMUNICATIONS

This didactic module couples several sensor systems on a board that provides the opening to practical knowledge of Electronic Engineering students at the Salesian Polytechnic University.

The monitoring, surveillance, temperature and humidity sensors, access control, are controlled by minicomputers such as Raspberry Pi model 4 and Arduino Uno. The design and implementation of the degree project is based on three levels of execution, firstly, Measurements will be taken with the sensors interconnected to the corresponding minicomputer, as a second step, it will be stored, processed and modified according to the user's preferences and the predisposed result will be obtained according to the system performed.

As a result, each practice will present an alarm state to the student programmer, which can be represented as a visual alarm through the lighting of led lights, an auditory alarm through the lighting of an electronic buzzer which converts the electrical signal into sensory output and the alarm wireless which presents alert messages using two systems such as text messaging (GSM technology) and email (SMTP technology).

The electronic security systems presented in the didactic module have connection diagrams to facilitate the execution of the practices and increase the students' interpretation, which will stimulate their understanding in reading the electronic technical manuals attached to this research work. Thus, the present research project results in a security system simulation that generates a solution that allows the supervision of the Data Center to be carried out, encouraging the student to develop theoretical and practical knowledge of it.

Key Words: *Data Center, sensors, minicomputers, engineering.*

Introducción

Con el paso del tiempo, se vuelve imperativo el manejo rápido de información para la correcta toma de decisiones o pronta respuesta a problemas, el avance de la tecnología ha permitido que la brecha entre el ser humano y la información desde cualquier parte del mundo se reduzca cada vez más, dando paso así a la seguridad electrónica y la tecnificación de procesos para el acceso de esta.

Los centros de datos o conocidos comúnmente en el idioma inglés “Data Center” se han vuelto en el centro de información de las compañías de telecomunicaciones en todo el mundo, la información que puede almacenarse en los servidores puede ser de la más simple a la más cifrada, con los años, la evolución del internet y la necesidad de pasar conectados ha forzado a las mismas a requerir un alto nivel de fiabilidad y seguridad, para salvaguardar los datos y mantener en condiciones óptimas los dispositivos.

Debido a la restricción existente en los Centros de Datos, propia de la seguridad de la información que contiene, surge la necesidad de realizar un módulo didáctico para que estudiantes de carreras afines tengan conocimientos tanto teóricos como prácticos de la forma en que operan los sistemas electrónicos de seguridad tales como: CCTV, Sensores de movimiento, control de acceso, Sensores de temperatura y humedad, elementos de la seguridad encontrada en los centros de datos.

El presente estudio tiene como fin la creación de un módulo didáctico sobre programación de elementos con código abierto, diseño y análisis de la seguridad electrónica para el uso académico en el laboratorio de la carrera de Ingeniería Electrónica; el mismo se desarrollará en cuatro capítulos divididos de la siguiente manera: el capítulo I comprenderá el desarrollo del problema, los objetivos tanto general como específicos del estudio; el capítulo II contiene las bases teóricas en las cuales se fundamenta el módulo desarrollado para uso didáctico, el capítulo III determinará los componentes que conformarán el módulo y el porqué de su elección para el entendimiento y práctica del estudio; el capítulo IV detalla las prácticas de programación de cada uno de sus componentes en conjunto, las conclusiones obtenidas y recomendaciones para futuros estudios.

Capítulo I

1.1. Planteamiento del problema

En la actualidad el Laboratorio de Telecomunicaciones de la carrera Ingeniería Electrónica de la Universidad Politécnica Salesiana sede Guayaquil, no cuenta con módulos didácticos para el aprendizaje y programación de sistemas de seguridad electrónica basadas en miniordenadores tales como Raspberry Pi y Arduino.

Debido a la inexistencia de estos equipos especializados en seguridad electrónica, no es posible realizar una enseñanza práctica a los estudiantes con la finalidad de fortalecer los conocimientos teóricos adquiridos en las aulas de estudio.

1.2. Antecedentes

La innovación y el impacto de la tecnología en la actualidad obliga a simplificar estructuras para mejorar y optimizar los recursos que nos brindan sus aplicaciones en el ámbito profesional y laboral.

El área de la electrónica al ser tan extensa avanza y evoluciona rápidamente de acorde a las necesidades, por lo cual, es necesario conocer cómo funcionan los equipos de trabajo de las áreas más importantes en las que se aplica la carrera como lo es la seguridad electrónica.

1.3. Importancia y Alcance

Con la implementación de este proyecto técnico se pretende ampliar los equipos del laboratorio de Telecomunicaciones, al innovar con un módulo didáctico donde se pueda aprender de forma práctica varios sistemas de seguridad aplicados en el ámbito profesional en un solo tablero, como lo son los sistemas de CCTV, sistemas de sensor de movimiento, sistemas de temperatura y humedad, sistemas de control de acceso; los cuales los podemos encontrar profesionalmente en las áreas de Data Center.

De acuerdo con las normas de seguridad de los Data Center, el acceso es restringido y monitoreado, los dispositivos que se necesitan para su implementación, mantenimiento y control deben tener sistemas de seguridad para cumplir con la privacidad de la información. Hoy en día las cámaras de seguridad, sensores de movimiento, control de acceso, sensores de temperatura y humedad, se pueden observar en diferentes situaciones como: sistemas antirrobo, detectores de movimiento, control de temperatura, vigilancia y monitoreo.

El personal técnico del área debe tener conocimientos de todos los sistemas de seguridad para mantener el correcto funcionamiento de los equipos y poder corregir errores en caso de que ameriten. Por esta razón se crea la necesidad de presentar un módulo didáctico dirigido a los estudiantes de la carrera de Ingeniería Electrónica y carreras afines como solución para el aprendizaje y conocimiento de los dispositivos que se necesitan para formar un Data Center y su seguridad.

La demanda de aprendizaje de la seguridad electrónica y todos los dispositivos que esta engloba, ha crecido en los últimos años, debido a la amplitud de su campo de aplicación, el comprender la funcionabilidad de sus accesorios permitirá una ventaja en el ámbito laboral.

Los principales beneficiarios de este proyecto será los estudiantes que empiecen a cursar materias como: Redes de computadoras –Redes inalámbricas, Electiva I, II y II, logrando adquirir conocimientos prácticos de programación en diversos lenguajes como: Java y Python en sistemas operativos LINUX; de la misma manera, los docentes que imparten estas materias en la carrera de Ingeniería Electrónica e Ingeniería de Telecomunicaciones de la Universidad Politécnica Salesiana Sede Guayaquil, tendrán una herramienta para la ejecución de sus clases.

1.4. Delimitación del problema

1.4.1. Delimitación temporal.

El proyecto de titulación se realizó en el período académico 2020 – 2021.

1.4.2. Delimitación espacial.

Debido a la emergencia sanitaria que vive el país y por los permisos correspondientes definidos por las autoridades del COE Cantonal de la ciudad de Guayaquil, el proyecto se desarrollará en un área de trabajo residencial de uno de los coautores y estará disponible en el laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil, tal como se puede apreciar en la Figura 1. Ubicación Universidad Salesiana, tomada de Google Maps. posterior del periodo de sustentación del proyecto técnico.



Figura 1. Ubicación Universidad Salesiana, tomada de Google Maps.

1.4.3. Delimitación académica.

El proyecto estará a disposición de la carrera de Ingeniería Electrónica en el Laboratorio de Telecomunicaciones el cual está situado en las instalaciones del Bloque E de la Universidad Politécnica Salesiana sede Guayaquil en Gral. Francisco Robles 107.

La implementación del proyecto técnico ayudará a los estudiantes que cursarán las materias de Electiva I, II y III, Redes de computadoras y Redes inalámbricas a obtener conocimientos prácticos mediante un módulo didáctico y así mejorar su perfil profesional.

1.5. Objetivos

1.5.1. Objetivo general.

Diseñar e implementar un módulo didáctico orientado a seguridad electrónica en Data Center basado en Raspberry Pi y Arduino para prácticas en el Laboratorio de Telecomunicaciones.

1.5.2. Objetivos específicos.

Entre los objetivos específicos planteados para la presente investigación están:

- Diseñar un sistema de seguridad electrónica basado en un circuito de cámaras de seguridad en Raspberry Pi.
- Implementar y programar el servidor FTP y SMTP en un Raspberry Pi.
- Diseñar un sistema de seguridad electrónica basado detector de movimiento en Raspberry Pi.
- Diseñar un sistema de control de acceso mediante el uso de tarjetas RFID R522.
- Diseñar un sistema de seguridad de medición de temperatura y humedad en Raspberry Pi.
- Diseñar y realizar el montaje mecánico de un tablero de seguridad electrónica con acoples entre las cámaras, sensores y alarmas hacia microcontroladores tales como Raspberry Pi y Arduino.
- Diseñar un conjunto de 10 prácticas con un grado de dificultad ascendente, mostrando las distintas aplicaciones con los equipos de seguridad electrónica que se le puede dar al Raspberry Pi y Arduino.

Capítulo II

Marco Teórico

2.1. Data Center.

Para (PACIO, 2014), el Centro de Procesamiento de Datos, como también se lo conoce, es el lugar donde las compañías alojan todo el equipamiento tecnológico brindando seguridad y confiabilidad; este espacio deberá cumplir con ciertas características físicas de refrigeración, protección y redundancia, asegurando así la disponibilidad de los servicios de red.

Un centro de datos como lo indican Briones, Landires y Andrade (2020) es esencial en una compañía, ya que en él se alojan los activos más importantes de la empresa, los servidores, que en esencia son los encargados de la información enviada a través de los correos electrónicos y las transacciones del negocio en general.

2.2. Seguridad electrónica.

Como lo determina Rodríguez (2018), la seguridad electrónica se fundamenta en todos los sistemas que soportados en componentes tecnológicos interconectados entre sí y gestionados de una o varias centrales, generen la seguridad/protección de los bienes o personas encontradas en el interior del área de la que registran información.

Todos los equipos que conforman el sistema se encuentran conectados con un centro de alarmas que, dependiendo de la situación, podrán accionar sirenas, activar equipos de extinción de fuego, grabar lo que ocurre en el área, detectar problemas de temperaturas, con el fin de tener una respuesta rápida; los sistemas de seguridad serán adaptados según las necesidades y presupuesto que tenga el interesado (FERNÁNDEZ, 2018).

Como se especifica en la Tabla 1., existe una diferencia en los sistemas de seguridad electrónica, definiéndolos en su enfoque de seguridad, sea este desde mediciones de temperaturas o riesgos físicos hasta grabaciones del área a resguardar.

Tabla 1.

Clasificación de los sistemas de seguridad electrónica.

SEGURIDAD DE INSTALACIONES		SEGURIDAD PRIVADA	
Sistemas de seguridad contra incendio.	Sistema de detección de temperatura y humedad.	Sistemas de seguridad de control de acceso.	Sistemas de movimiento antirrobo e intrusión.
			Circuito cerrado de televisión e (CCTV).

Información tomada del libro “Circuito cerrado de televisión y seguridad electrónica”, elaborado por Julián Rodríguez Fernández.

2.2.1. Elementos que constituyen un sistema de seguridad(2018).

Los componentes tecnológicos que pueden formar las diferentes instalaciones de seguridad electrónica se encuentran determinados en la Figura 2, teniendo como punto central de cada uno de los artefactos tecnológicos la central de alarmas, a través de la cual se recibe la información en forma de señales originarias de los sensores, interpretadas en función a la programación establecida para el sistema.

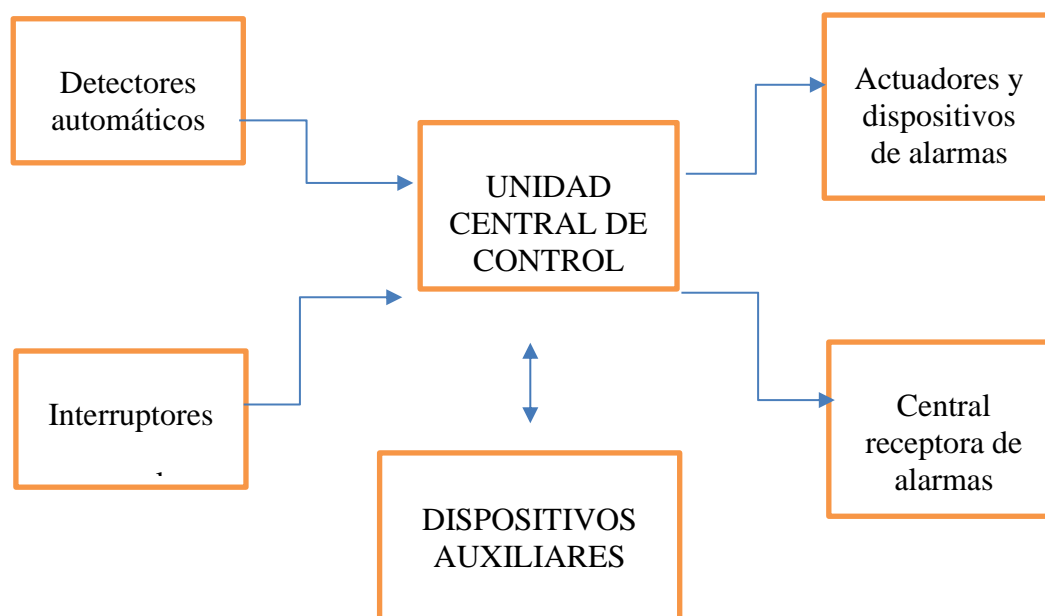


Figura 2. Bloques funcionales que componen un sistema de seguridad electrónica, tomado del libro “Circuito cerrado de televisión y seguridad electrónica”

2.3. Estándar de seguridad para un Data Center.

Para el diseño de una data center, se consideran algunas normas y estándares para su correcto funcionamiento y protección, para el desarrollo del presente estudio que se basa en un módulo didáctico, se procede a seguir los requisitos expuestos en el Estándar TIA-942 y la norma ISO IEC 27001, las cuales son descritas a continuación.

2.3.1. Estándar TIA 942 (EIA, 2005).

Telecommunication Industry Association en el año 2005 publica el Estandar TIA-942, teniendo como finalidad la unificación de los criterios a considerarse en el diseño de áreas de tecnologías y comunicaciones como lo es el Data Center (Enrich, 2007). Se origina en un conjunto de especificaciones para comunicación y cableado de estructuras, divide la infraestructura y soporte en cuatro subsistemas:

- Telecomunicaciones.
- Arquitectura.
- Sistema eléctrico.
- Sistema Mecánico.

Estos subsistemas serán aplicados para los cuatros tipo de Data Center pero serán según la necesidad que se presenta para cada nivel, los cuatros niveles de tiers que plantea el estándar se corresponde con cuatros niveles de disponibilidad, teniendo que a mayor número de tier, mayor número de disponibilidad, generando como consecuencia mayores costos de construcción (Enrich, 2007).

Este estándar o práctica recomendada sugiere al poseedor o diseñador del Data Center un sin número de recomendaciones para el diseño y estructura con el fin de avalar la seguridad tanto interna (redes) como externa del Data Center, según el tipo de Data Center, en este caso de estudio la simulación de sistema de seguridad estará apto para un Data Center TIER II, III y IV. A continuación, en la

Tabla 2 se procede a establecer los puntos para cada subsistema

Tabla 2.

Items clasificados por Subsistemas.

Telecomunicaciones	Arquitectura	Eléctrica	Mecánica
Cableado de racks, gabinetes, Pathways.	Selección del lugar.	Cantidad de accesos.	Sistemas de climatización.
Acceso redundante.	Tipo de construcción.	Puntos únicos de falla,	Presión positiva.
Sala de Entrada secundaria.	Protección ignífuga.	Cargas críticas.	Cañerías y drenajes.
Área de distribución secundaria.	Requerimientos NFPA 75.	Redundancia de UPS.	Chillers.
Rutas de red troncal.	Barrera de vapor.	Topología de UPS	CRAC s y condensadores.
Cableado horizontal redundante.	Techos y pisos.	PDU s	Control de HVAC.
Routers y Switches	Áreas de oficinas.	Puesta a tierra.	Detección de incendio.
Patch panels, tomas de corrientes y cableado.	NOC	EPO (Emergency Power Off).	Sprinklers.
Cables de conexión y puentes.	Sala de UPS y Baterías.	Baterías	Extinción por agente limpio (NFPA 2001).
Patch Panel Patch Cable documentación.	Sala de generador.	Monitoreo.	Detección por aspiración (ASD).
	Control de acceso	Generadores.	Detección de líquidos.
	CCTV	Transfer Switch	

Tomado del Artículo “Estándar TIA 942”, elaborado por Gustavo García (Enrich, 2007).

2.4. Norma ISO / IEC 27001 (NQA, 2017).

Las normas de la serie 27000 tienen su origen en el año 1995, su correcta denominación es “ISO / IEC” debido a que son desarrolladas y mantenidas en conjunto

por dos organismos internacionales de normas: ISO (Organización Internacional de Normalización) e IEC (Comisión Electrónica Internacional).

Esta norma es utilizada para los sistemas de gestión de la seguridad de la información, genera un conjunto de procedimientos con la finalidad de poder planificar en función a la solicitud del cliente, estableciendo previamente una serie de objetivos, recursos, políticas organizativa e identificación de riesgos; luego de esta planificación previa, se procede a implementarla según lo planificado; se verifica los procesos establecidos a través de controles y análisis de rendimientos tanto de políticas y cumplimientos de objetivos; a través de esta verificación le permite a la empresa tomar acciones basándose en los resultados obtenidos, como podemos observar en la Figura 3.

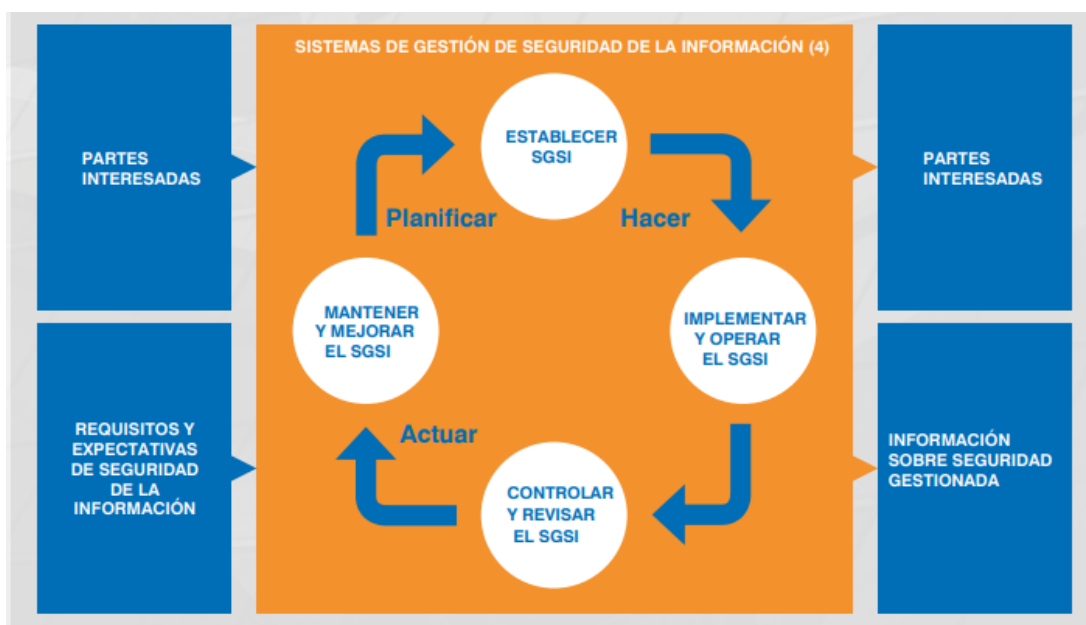


Figura 3. Modelo PHVA para ISO / IEC 27001, tomada de NQA- Guía de implantación para la seguridad de la Información.

Como se menciona en la “Guía de Implantación para norma ISO / IEC 27001” (2017) el objetivo principal de un Sistema de Gestión de Seguridad de la Información es generar protección a la información de valor de la empresa, siendo esta considerada dentro de la norma como un Activo, desde esta perspectiva se clasifican los riesgos existentes en 3 categorías: Confidencialidad, Integridad, Disponibilidad.

Un Sistema de Gestión de Seguridad de la Información que cumple con la ISO 27001, tiene un conjunto interrelacionado de procesos de mejores prácticas que facilitan y respaldan el diseño, implementación y mantenimiento de los controles

2.5. Estudio del Hardware

2.5.1. SingleBoardComputers.

Para(Sandoval-gutiérrez & García-sánchez, 2021), la primera aparición de este mini ordenador data en la década de los 70s, conocida como Dyna-Micro, basada en el procesador C8080A de Intel, más adelante, en el año 2008 con el nacimiento de la fundación BeagleBoard.org¹, se desarrollan tarjetas educativas que puedan ser usadas en la enseñanza de las capacidades de Software y hardware libre.

Las placas SBC, como lo indica (Garcia Cobo, 2017), son placas que contienen la mayor parte de los componentes que tiene un ordenador común; su principal característica es su reducido tamaño, aproximadamente los mini ordenadores están montados sobre placas con medidas inferiores a 17x17 cm., como ejemplo, tenemos la Raspberry Pi que mide 8.5x 5.3 cm.

De igual manera, (Artero, 2013) define al microcontrolador como un circuito integrado que tiene la característica de ser programable, siendo capaz de ejecutar una acción de manera autónoma, siguiendo las instrucciones pregrabadas inicialmente por el usuario. Se considera el principal componente de cualquier circuitería de procesamiento y control, que permite a un sistema electrónico obtener y procesar información a través de sensores (medios de entrada), adicional, permiten grabar y generar salidas hacia los actuadores que admitirán interactuar con el medio físico.

Entre las placas más conocidas en el mercado se encuentran: Raspberry Pi, BeagleBone Black, PcDuino, Pandaboard y Arduino; para desarrollo del presente estudio se conceptualizarán a continuación las SBC a utilizar.

2.5.2. Arduino.

Oscar Artero en su curso práctico de formación (2013) lo define, como una placa desnuda conocida también como placa Hardware libre, que tiene adecuado en su centro un microcontrolador con funciones pregrabadas y una serie de pines de conectores de entrada y salida, los cuales permiten establecer una comunicación física de manera simple y eficaz a los distintos sensores y actuadores.

En su artículo “Una mirada al mundo Arduino (2015)”, Herrero y Sánchez, señalan que el principal objetivo de la creación de Arduino es aportar una plataforma de costo económico, manejable y de fácil acceso para la enseñanza electrónica, inicialmente en

¹ Fundación BeagleBoard.org(BeagleBoard, n.d.): Fue creada con la finalidad de brindar educación y colaboración en relación al diseño y uso de software y hardware de código abierto en la computación integrada.

Italia (país de origen de los creadores), patentando la marca pero permitiendo que el hardware y el software sean de acceso libre.

El hardware de Arduino, como lo señalan sus creadores (2019), está constituido por una baqueta electrónica desarrollada en torno a un microcontrolador de Atmel, tomando información del entorno a través de sus pines de entrada de toda una gama de sensores.

En su artículo, (Herrero Herranz & Sánchez Allende, 2015), señalan que hasta el momento existen más de 20 modelos de Arduino, con diferentes características relacionadas a su uso y tamaño, las cuales se detallan en la Tabla 3.

Tabla 3.

Modelos básicos de la placa Arduino

Modelo	Características
Arduino UNO R3	Microcontrolador ATmega328, a 16MHz. Tiene 14 pines de entrada y salida digitales, 6 de ellas también tiene función como salidas PWM, 6 entradas analógicas de 10 bits de resolución, 1 UART Memoria Flash 32KB, SRAM 2KB, EEPROM 1KB
Arduino DUE	Microcontrolador Atmel AT91SAM3X8E ARM Cortex-M3 de 32 bits, a 84 MHz. Tiene 54 entradas y salidas digitales, de las cuales 12 se pueden utilizar como salidas PWM, 12 entradas analógicas con resolución de 12 bits, 4 UARTs, dos salidas analógicas con DAC, memoria flash 512KB, SRAM 96KB.
Arduino Leonardo	Microcontrolador ATmega32u4, a 16 MHz. Tiene 20 pines de entradas y salidas digitales, 7 de ellas también tienen función como salidas PWM, 12 entradas analógicas de 10 bits de resolución, 1 UART. Memoria flash 32KB, SRAM 2.5KB, EEPROM 1KB.
Arduino Mega 2560 Rev 3	Microcontrolador ATmega2560, a 16 MHz. Tiene 54 pines de entradas y salidas digitales, 15 de ellas también tienen función como salidas PWM, 16 entradas analógicas de 10 bits de resolución 4 UARTs. Memoria flash 256KB, SRAM 8KB, EEPROM 4KB.
Arduino ADK Rev3	Microcontrolador ATmega2560, a 16 MHz. Tiene 54 pines de entradas y salidas digitales, 15 de ellas también tienen función como salidas PWM, 16 entradas analógicas de 10 bits de resolución 4 UARTs. Memoria flash 256KB, SRAM 8KB, EEPROM 4KB.

Adaptado del libro “Arduino. Edición (2018). Curso Práctico, elaborado por autores.

2.5.2.1. Arduino Uno.

Como lo denomina (Artero, 2013), en esencia es un PCB² que permite conectar los diversos dispositivos electrónicos mediante soldadura a los caminos eléctricos. Su software es gratuito, de acceso libre y multiplataforma, es decir, nos permite el uso de

² Printed Circuit Board – Placa de circuito impreso.

diversos sistemas operativos en nuestro ordenador, debido a que funciona con Windows, MacOS y Linux.



Figura 4. Placa de Circuito Impreso Arduino Uno, tomado de www.arduino.cc

Herrero y Sánchez (2015) señalan que Arduino Uno R3 es una de las plataformas más representativas entre los usuarios, siendo esta, la versión actual de las primeras versiones de la plataforma.

2.5.2.1.1. Componentes de Arduino Uno (Artero, 2013). Entre los componentes encontrados en esta PCB se detallan a continuación:

- Conector USB: En la versión de Arduino Uno puede ser tipo B o mini, por medio de él se ejerce la comunicación para la programación y también para la alimentación de fuente 5 voltios.
- Regulador de voltaje de 5V: Está encargado de convertir el voltaje de ingreso por el plug a un voltaje regulado.
- Plug de alimentación externa, puede ser entre 6V – 18V.
- Puerto de conexiones, 6 pines (Reset Pin de 3.3v, Pin de 5v., dos pines de GND, pin Vin).
- Puerto de entrada analógica.
- Microcontrolador Atmega 328.
- Botón Reset.
- Pines de programación ICSP (Usados para programar microcontroladores).

- Chip de comunicación.

2.5.3. Raspberry Pi.

En su artículo, (Salcedo & Cendrós, 2016), señala que la creación de la Raspberry Pi encuentra su base de arquitectura en un microprocesador ARM, el cual suele ser utilizado en teléfonos inteligentes; originado por la fundación educacional que posee el mismo nombre, la cual busca colaborar en la enseñanza de la programación en lenguajes como Scratch y Python, así como el manejo de sistemas operativos de distribución de Linux(2017).

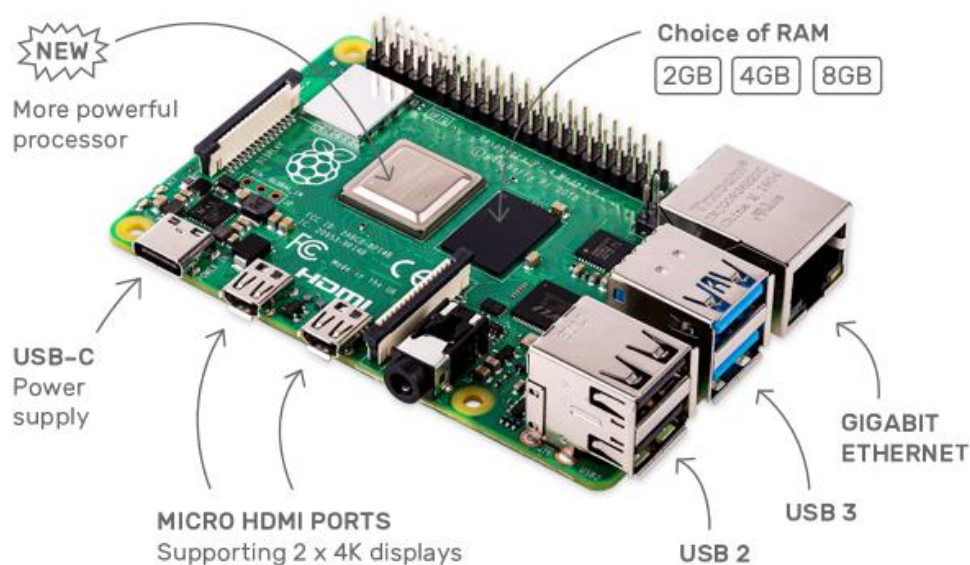


Figura 5. Placa de Modelo Impreso Raspberry Pi, tomado de www.raspberrypi.org.

Varias versiones de Linux han sido portadas al chip de la placa en mención (Patil, 2017) incluyendo entre ellos Debian, Fedora Remix y Arch Linux, cada uno atendiendo a las necesidades presentadas, teniendo en común una sola cosa: son de código abierto y, por lo general, todas compatibles entre sí, a excepción de Windows y Os que no son compatibles con Linux.

Raspberry Pi es considerado un ordenador de placa reducida, llegando a indicarse que su tamaño es semejante al de una tarjeta de crédito (Eugenio, López, 2017), el cual puede conectarse a un televisor y teclado. A continuación, en la Tabla 4 se detallan los modelos básicos existentes en el medio.

Tabla 4.

Modelos básicos de la SBC Raspberry Pi.

Modelo	Características
Raspberry Pi modelo A	Posee un solo puerto USB, carece de controlador Ethernet, se puede conectar a una red usando un adaptador USB-Ethernet. El sistema cuenta con 256MB de memoria RAM, se puede utilizar teclados y mouse con conexión USB.
Raspberry Pi modelo B+	Versión mejorada, el sistema cuenta con 512MB de memoria RAM, 3 puertos USB, conexión de Red Ethernet 10/100 Mbit/seg., lo que le permite conectarse a una red.
Raspberry Pi Modelo B 2	Versión mejorada, el sistema cuenta con 512MB de memoria RAM, 3 puertos USB, conexión de Red Ethernet 10/100 Mbit/seg., lo que le permite conectarse a una red, 40 pines de GPIO, salida de video 1080píxeles, lector de tarjeta Micro SD, puerto de red, interfaz de cámara CSI.

Adaptado de sitio web www.raspberrypi.org. elaborado por: autores.

2.5.3.1. Diseño de la Raspberry Pi (Patil, 2017):

- Un Chipset Broadcom BCM2835, lo cual contiene un procesador central ARM1176JZF-S a 700MHz.
- Un procesador gráfico (GPU) VideoCore IV.
- Un módulo de 512MB de memoria RAM.
- Un conector de RJ45 conectado a un integrado lan9512-jzx de SMSC.
- 2 buses USB 2.0.
- Una salida analógica de audio estéreo por Jack de 3.5 mm.
- Salida digital de video –audio HDMI.
- Salida analógica de video RCA.
- Pines de entrada y salida de propósito general.
- Conector de alimentación micro USB.
- Lector de tarjeta SD.

2.5.3.2. Raspberry Pi 4(2019).

Es el último producto de la gama Raspberry Pi de ordenadores, teniendo como principal ventaja la velocidad mejorada del procesador, rendimiento multimedia, su memoria y conectividad, conservando su compatibilidad con versiones anteriores.



Figura 6. Raspberry Pi 4 modelo B, tomado de <https://datasheets.raspberrypi.org/>

Entre las especificaciones específicas se detallan:

- Procesador: Broadcom BCM2711, Cortex-A72 de cuatro núcleos (ARM v8) SoC de 64 bits a 1,5 GHz.
- Memoria: LPDDR4 de 1 GB, 2 GB, 4 GB u 8 GB con ECC en matriz.
- Conectividad: 2.4 GHz y 5.0 GHz IEEE 802.11b/ g / n / ac inalámbrico LAN, Bluetooth 5.0, BLE.
- Gigabit Ethernet,
- 2 puertos USB 3.0
- 2 puertos USB 2.0.
- GPIO: Cabecera GPIO estándar de 40 pines.
- Sonido del video: 2 x puertos micro HDMI (hasta 4kp60 compatible).
- Puerto de pantalla MIPI DSI de 2 carriles.
- Puerto de cámara MIPI CSI de 2 carriles.
- Puerto de video compuesto y audio estéreo de 4 polos.
- Multimedia: H.265 (Decodificación 4Kp60);
- H.264 (Decodificación 1080p60, codificación 1080p30);
- OpenGL ES, gráficos 3.0
- Soporte de tarjeta SD: Ranura para tarjeta microSD para cargar el sistema operativo y almacenamiento de datos.

- Potencia de entrada: 5 V CC a través del conector USB-C (mínimo 3 A) 5 V CC a través del encabezado GPIO (mínimo 3 A). Alimentación a través de Ethernet (PoE): habilitado.
- Ambiente: Temperatura de funcionamiento 0 – 50°C.

2.6. Circuito Cerrado de Televisión

En su artículo “Fundamentos de diseño para un Circuito Cerrado de televisión (2009)”, los autores señalan como su objetivo principal la supervisión, control y registro de actividad física dentro del Data Center, a diferencia de la televisión tradicional, este circuito sólo permite un acceso contenido de las imágenes a algunos usuarios.

Por su parte, (FERNÁNDEZ, 2018), en su libro “Circuito Cerrado de televisión y seguridad electrónica” señala que su funcionamiento se encuentra basado en un conjunto de cámaras, monitores entre otros dispositivos de tratamiento de señal de video y audio, logrando enviar imágenes de manera remota conectados a una red.

2.7. Cámaras IP

También conocida como cámara web, (Tavil, 2012), en su informe la define como una videocámara diseñada con el fin de enviar señales de video y dependiendo el modelo de la misma, hasta audio a través de internet o de un switch en una red local; en estas pueden integrarse aplicaciones como detección de individuos a través del detector de movimiento, envío de correos en caso de detección e incluso grabar imágenes de lo sucedido.

2.8. Sensor de temperatura y humedad.

La humedad se considera como un fenómeno natural, como indica (Bergkvist et al., 2012), se presenta a nivel molecular y se encuentra relacionada con la cantidad de moléculas de agua que están presentes sea está en estado líquido o gaseoso. A pesar de no tener un gran impacto en la vida cotidiana del ser, se vuelve relevante en procesos industrializados.

Como señala (Tuero Suárez et al., 2016), la medición de la humedad se ha vuelto de gran relevancia en distintos sectores económicos, existiendo así en el mercado más de un instrumento que cumpla esa función, sin embargo, las propiedades de los sensores que conforman los distintos dispositivos suelen ser el problema raíz de la incorrecta medición de los mismos.

En la actualidad (2021), se han creado muchos instrumentos para la medición de temperatura con la misma diversidad que se encuentra en los sensores de temperatura.

2.9. Sensor de movimiento PIR HC-SR501

El sensor de movimiento PIR es un dispositivo que mide los cambios en los niveles de radiación infrarroja, a una distancia de máximo 6 metros. Si recibir energía el Sensor

PIR de un tiempo de encierro completo, el cual requiere un tiempo mayor para iniciar su funcionamiento.

Como señala (Cajo & Galarza, 2010), es un dispositivo piro eléctrico que mide cambios en los niveles de radiación infrarroja emitida, contando con tres terminales, entre sus características:

- Voltaje de alimentación: 5VDC.
- Rango de medición: hasta 6 metros.
- Salida: Estado de un pin TTN.
- Polaridad de activación de salida seleccionable.
- Mínimo tiempo de calibración.

2.10. Sistema RFID (Radio Frequency Identification).

Como señala (Weinstein, 2005), el año 1940 muestra el inicio y desarrollo de la tecnología de radiofrecuencia, teniendo como principal motor la Segunda Guerra Mundial, siendo usada para identificación y rastreo de aviones aliados y enemigos. Con el paso del tiempo, esta fue implementada en la industria ferroviaria y en la década de los 60 y 70 fue orientada en la seguridad de materiales nucleares.

Actualmente la utilización del sistema RFID se enfoca en el rubro de la seguridad, ya sea en el control de cruces fronterizos, identificaciones de credenciales, vigilancia y control de vehículos, identificación de animales de ganado, rastreo de envoltorios, seguridad de paquetes de avión, y artículos de compra y venta, y en seguridad electrónica como dispositivos de antirrobo.

La tecnología de RFID es un sistema de control que consiste en asignar un código de información de un producto determinado.

Los sistemas de RFID de componen de:

- El transponder, o también llamado tag, una pequeña etiqueta electrónica, que tiene grabado una identificación de lector único.
- El lector, encargado de la identificación única del transponder.

2.11. Biometría.

Es considerada como bien lo define (Michilena & Torres, 2018), una tecnología de seguridad fundamentada en el reconocimiento de una característica física e intransferible, constando de tres partes principales: Mecanismo automático que lee y captura una imagen digital, entidad para mejorar aspectos de comprensión, almacenamiento de datos capturados con los guardados en su base de datos.

Entre los métodos principales que señala (Michilena & Torres, 2018) en su estudio encontramos los siguientes:

- Identificación de huella dactilar.
- Reconocimiento visual a través de la Iris.
- Reconocimiento visual a través de la retina.
- Reconocimiento facial.
- Análisis de ADN.

2.11.1. Reconocimiento de huella dactilar.

Como indican (Gualberto Aguilar, Gabriel Sánchez, Karina Toscano, Mariko Nakano, 2008), el reconocimiento de huella dactilar es uno de los métodos más conocidos para la identificación de usuarios, debido a las características únicas (minucias) encontradas en ellas.

Dentro de su proceso, es considerado por (Vargas, 2013) como uno de los más fáciles de entender y revisar para el registro eficaz de usuarios aptos para el ingreso del lugar que lo resguarda, dentro de su funcionamiento, se procede inicialmente con la captura registrada en su base de datos, la cual reúne las características únicas de la huella del usuario, una vez el mismo, registre su huella, al ingresar, el sistema procede a procesarlo, realizando la comparación de características con la información registrada en relación a la extraída y si éstas presentan similitud la decisión de autorización es procesada.

2.11.2. Acceso con tarjeta Magnéticas(VARGAS VERGARA, 2013).

Los sistemas de control de acceso que utilizan la lectura de tarjetas magnéticas se basan en la lectura de una banda magnética adherida al objetivo pasivo del lector. Estas tarjetas utilizan señales electromagnéticas para registrar y codificar la información impresa en una banda que se puede leer desde una máquina de identificación instantánea. En la vida cotidiana lo encontramos representado en todas las tarjetas de crédito.

2.12. Estudio del Software.

2.12.1. Lenguaje de programación.

Como lo define (Trigo, 2004), se denomina como programa al grupo de órdenes e instrucciones que se dan al ordenador para que resuelva un problema o realice cierta acción. Si bien es cierto, en el inicio, la programación se realizaba en su propio código binario, conocido también como código máquina, sin embargo, no es considerada la más rápida, debido a que la única manera de introducir la información es en sistema binario.

Llorens (2015), acerca de la programación señala “Las competencias que se desarrollan al escribir un código en un lenguaje informático son la parte más visible de esta forma de pensar, de organizar ideas y relacionarlas, de representar conocimiento de una manera lógica”.

Como lo describe en su libro (Joyanes Aguilar, 2003), los lenguajes utilizados en la programación se pueden clasificar en: Lenguajes máquina, lenguajes de bajo nivel y de alto nivel. Los lenguajes de máquina proveen instrucciones puntuales para un determinado tipo de Hardware; el lenguaje ensamblador se caracteriza por la sencillez de sus instrucciones, el lenguaje de alto nivel aporta sentencias fáciles de recordar (programas fuentes).

En su artículo, (Trigo, 2004), detalla la historia de los lenguajes de programación creados, los cuales se describen en la Figura 7 detallando el uso de los mismos y origen. Esto nos permite ver cómo el avance tenido en hardware y arquitectura de los ordenadores influye en la construcción de lenguajes con el fin de complementar lo antes creado.

Lenguaje de programación	Creador	Año de creación	Características
FORTAN	John Backus	1956	Se crea con el fin de agregar prestaciones al modelo IBM 704, con la finalidad de resolver problemas científicos - técnicos.
COBOL	Grace Hopper	1960	Nace en la conferencia CODASYL, Conference on Data Systems Languages, orientado a las funciones administrativas, de gran portabilidad y legibilidad.
BASIC	John G. Kemeny y Thomas E. Kurtz	1964	Lenguaje que permite introducirse a estudiantes en los sistemas de tiempo compartido, debido a su sencillez de manejo y su poco peso en la memoria del ordenador.
LOGO	Jean Pager	1967	Se crea con el fin de introducir en el mundo de la programación a estudiantes de menor edad, estudiantes de primaria.
C	Kenneth Thompson y Dennis Ritchie	1970	No depende de la arquitectura de los hardware para su funcionamiento, siendo así uno de los más portables del mercado.
PASCAL	Niklaus Wirth	1974	Es el lenguaje más sencillo que posibilita el acceso a la informática teórica.
JAVA	James Gosling	1990	Basado en C y C++, cuyo objetivo inicial era crear un interfaz atractivo e intuitivo para electrónica de consumo, pero luego de años pasó a ser utilizado para internet.

Figura 7. Historia de los primeros lenguajes de programación, adaptado de “Historia y evolución de los lenguajes de programación”, elaborado por autores.

Para el presente estudio, se profundizará en el lenguaje de programación Python, considerado en programación como uno de alto nivel.

2.12.1.1. Lenguaje de programación Python.

Categorizado en los lenguajes de programación como uno de alto nivel, el cual presenta ventajas de gran nivel para el desarrollo educativo de estudiantes en Ingeniería electrónica.

Python, como lo indica en la traducción del manual de uso de Python 3 (Rossum, 2015), es un lenguaje de programación fuerte y de fácil aprendizaje, contando con una arquitectura de datos eficientes y de gran nivel, a su vez, un enfoque efectivo a la programación dirigida a objetos. Siendo diseñado a finales de la década de los 80 por Guido Van Rossum, el cual permite programación imperativa y funcional (Silva, 2009).

Entre las ventajas destacadas de Python, (Marzal Varó et al., 2014), determina como interesantes desde el punto de vista didáctico a las siguientes:

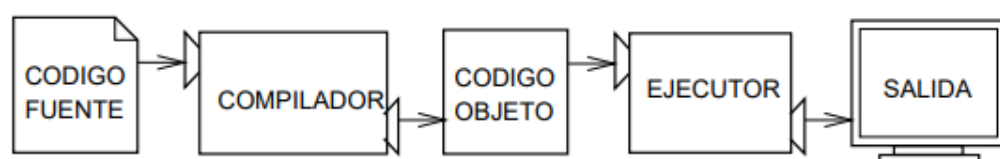
- Los programas de Python se consideran muy compactos, incluso más cortos que los de lenguajes como C.
- La sintaxis de Python es legible, permitiendo la escritura de programas cuya lectura sea de fácil entendimiento.
- Python brinda un entorno interactivo que ayuda en la elaboración de pruebas y permite el despeje de dudas sobre características del lenguaje, lo que permite un mayor entendimiento en un ambiente didáctico.
- Al ser de fácil escritura de programas, permiten detectar errores de programación que suelen pasar desapercibido al control de los compiladores y brinda las herramientas para corregirlos.
- Python presenta de forma gratuita su intérprete, siendo de fácil acceso en su página web <http://www.python.org>. Siendo de fácil acceso a cualquier sistema operativo.

Como lo determina (Bahit & ALCAZAR LUQUE, 2000) en su libro “Aprende a pensar como un programador”, existen dos tipos de programas que interpretan lenguajes de alto nivel a lenguaje de bajo nivel, tenemos el intérprete que lee un programa de alto nivel y lo ejecuta, es decir, realiza lo que el programa indica, como se puede observar en la Figura 8.



Figura 8. Funcionamiento del programa intérprete, tomado de “Aprende a pensar como un programador”(2000).

Así mismo, (Bahit & ALCAZAR LUQUE, 2000), señala que un compilador lee el programa y lo traduce en el mismo momento en que ingresa la información, antes de realizar cualquier instrucción dada por el programa, determinando como código fuente



al programa de alto nivel y al programa traducido como código de objeto, una vez agrupado el programa, puede ser ejecutado más de una vez sin tener que volver a realizar todo el procedimiento, su proceso se grafica en la Figura 9.

Figura 9. Funcionamiento del programa compilador, tomado de “Aprende a pensar como un programador”(2000).

2.12.1.1.1. *Sintaxis de Python.*

Al igual que otros lenguajes de programación, Python, está compuesto por algunos elementos los cuales son descritos a continuación basados en el artículo “Introducción al lenguaje Python (Valls, 2003)”:

- Variables: Espacio para almacenar datos modificables, en la memoria del ordenador, en Python esta se definirá con la sintaxis. Las variables pueden ser cadenas de texto, números enteros o reales o booleano (verdadero o falso).
- Sentencias: Se determina así a la ejecución de una instrucción realizada por el intérprete de Python, sea por print o por asignación.
- Comentarios: Texto explicativo que no es procesado por el intérprete y sólo tiene una utilidad humana.
- Codificaciones de caracteres.
- Tuplas: Variable que permite el almacenamiento de varios datos que no podrán ser modificados una vez se creen en el programa.
- Listas: Se diferencia de las tuplas, dado de que sus datos almacenados si pueden sufrir de modificación.
- Diccionarios: Al igual que las tuplas y las listas, son colecciones, la diferencia radica en la asociación de sus datos a través de un nombre clave.

2.13. Escritorio remoto.

Debido al continuo avance de la tecnología, la posibilidad de acceder a datos de un lugar determinado de almacenamiento del ordenador de tu oficina desde tu casa, ha incrementado, como lo señala (Edwin, Boza, 2012), actualmente, de la mano del desarrollo de las comunicaciones y dispositivos de procesamiento, permite el acceso a entornos gráficos comandando desde el computador remoto, teniendo prácticamente acceso a todo el escritorio. Esto se da a través de la creación de algunos protocolos, que para estudio y desarrollo del presente trabajo se procede a definirlos y profundizar en ellos.

2.13.1. Protocolo VNC (Virtual Network Computing).

Se considera como una aplicación que permite de manera remota la conexión entre equipos conectados en red, como lo explica (Ruben, 2006), para poder acceder al otro servidor, es necesario instalar dos programas, en el ordenador que deseamos acceder se necesita instalar un servidor VNC, mientras que en la máquina del cliente o de quién

solicita el acceso es necesario un visualizador VNC, este proceso puede visualizarse en la Figura 10.

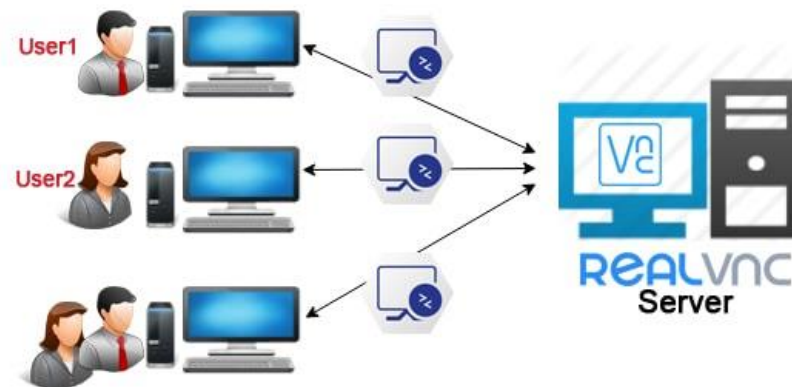


Figura 10. Gráfico explicativo sobre el funcionamiento del protocolo VNC, tomado de sitio web <https://www.serdarbayram.net>

Como lo detalla en su página web (RealVNC, 2021) las ventajas que presenta esta aplicación es amplia, entre las principales se detallan a continuación:

- Sencillo de instalación: Los usuarios no necesitan una formación previa para su manejo, debido a que su implementación y uso es fácil.
- Flexible: El acceso puede darse estando presente o no el propietario de los equipos remotos.
- Seguro: Las sesiones están encriptadas de extremo a extremo con el protocolo AES de 256bits.

2.13.2. Protocolo SSH (Secure Shell) (Gardoki, n.d.-b).

Señala que fue desarrollado en el año 1995 por el finlandés TatuYlonen, con una licencia de libre uso, sin embargo, por el impacto del programa y acogida, patentó la marca y la creó como empresa. Más adelante, los desarrolladores del sistema operativo OpenBSD empezaron a desarrollar una versión libre de este protocolo con el nombre de OpenSSH.

(Smaldone, 2004) en su “Introducción a Secure Shell”, indica que SSH es una aplicación cliente servidor que utiliza el puerto 22 de TCP, al conectarse el cliente con

el servidor, deberá seguir los pasos a continuación detallados (2004) y graficados en la Figura 11.:

- El cliente procede a abrir una conexión TCP al puerto 22 del host servidor.
- Se procede a acordar la versión del protocolo a utilizar entre cliente y servidor.
- El servidor posee dos claves públicas/privada (claves de host), de las cuales envía al cliente la clave pública, la cual se sujeta a verificación con la almacenada.
- El cliente genera una clave de sesión aleatoria y selecciona un algoritmo de cifrado simétrico.
- El cliente envía al servidor un mensaje con la clave y el algoritmo cifrado con la clave pública de host del servidor.
- De ahí en adelante se procede con el cifrado simétrico para el resto de comunicaciones.



Figura 11. Gráfico explicativo sobre el funcionamiento del protocolo SSH, tomado de sitio web <https://pressroom.hostalia.com>

2.14. Protocolo SMTP (Simple Mail Transfer Protocol)

(Young & Aitel, 2003) define al Simple Mail Transfer Protocolo como un protocolo de red utilizado para el intercambio de mensajes de correo electrónico entre ordenadores, su funcionamiento se basa en línea de manera que opera en los servicios de correo electrónico, sin embargo, posee algunas limitantes.

De igual manera lo define en su estudio (2003), como un protocolo que funciona en línea, encapsulado en una trama TCP/IP, es decir, el correo se envía directamente al servidor del correo del destinatario.

2.14.1. Estructura de SMTP (Riabov, 2011).

Cuando un servidor desea enviar un correo debe seguir una serie de pasos para poder transmitir el mensaje deseado al otro servidor, como se puede observar en la Figura 12.

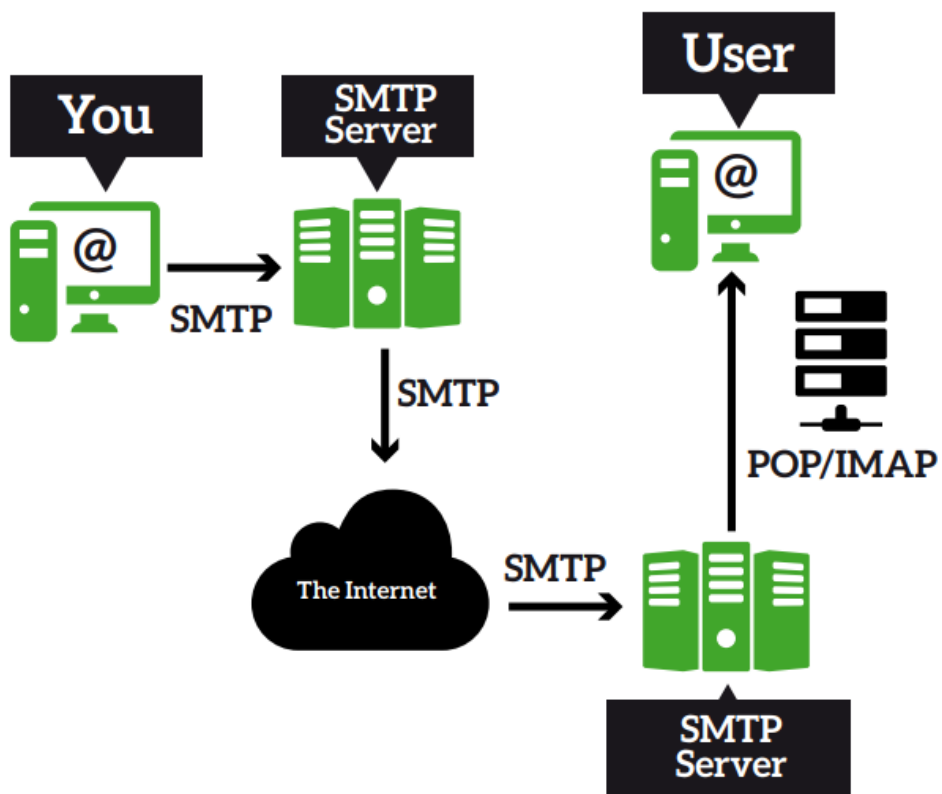


Figura 12. Proceso de envío de un correo desde el usuario uno al usuario dos a través del servidor SMTP, tomado del sitio web <https://ingsistemasunicor.files.wordpress.com/>

Es decir, que el esquema de este protocolo, se encuentra basado en el modelo de comunicación a continuación descrito (Vinícius V. da Conceicao, Leandro de B. Francisco, 2003):

- Debido a una solicitud realizada del usuario de enviar un correo electrónico, se genera una conexión entre el emisor y el receptor.
- El emisor genera los comandos SMTP en formato ASCII, los cuales los envía al receptor y este genera las respuestas a los comandos enviados por el emisor.
- Cuando se encuentra ya generado el canal de transmisión, el emisor envía el comando MAIL indicando que él es el emisor del correo, si el canal se había generado efectivamente el comando era respondido con un OK.
- El emisor envía el comando RCPTO identificando el destinatario del correo, si el receptor puede aceptar correo para ese destino con una respuesta ok, sino automáticamente se enviará una respuesta de rechazo al emisor.

2.15. Protocolo FTP

File Transfer Protocol o conocido por sus siglas FTP, es definido por (Gardoki, n.d.-a) como un protocolo para la transferencia de archivos entre sistemas que se encuentran conectados a una red, es decir, que desde un equipo base/cliente se puede conectar a otro/servidor para realizar la descarga de archivos o en su defecto realizar el envío de archivos, siendo indistinto el sistema operativo utilizado por cada equipo.

En su artículo de estudio sobre FTP, (Cobo, 2014), señala que este protocolo usa dos modos de transferencia de archivos: Modo texto y binario; en el modo texto aplica caracteres ASCII y utiliza retorno de carro y carácter de nueva línea, mientras que en el modo binario no realiza ninguna conversión de formato.

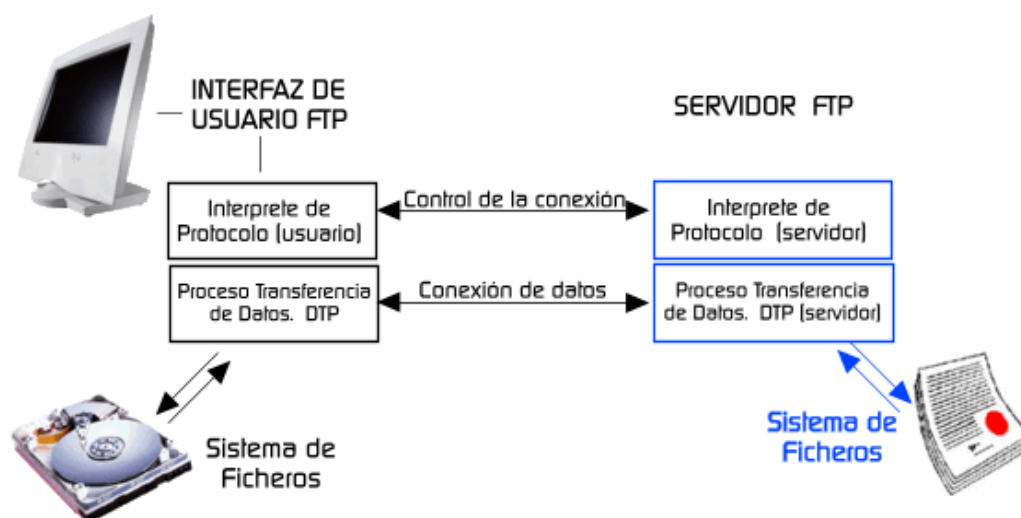


Figura 13. Proceso realizado por protocolo FTP para la transferencia de archivos desde ordenador cliente – servidor, tomado de sitio web <https://neo.lcc.uma.es/>

2.15.1. Configuración del servicio de transferencia de archivos, permisos y cuotas, conexiones en modo activo(Sánchez et al., 2015).

En la Figura 13, se puede observar el procedimiento realizado gráficamente del protocolo FTP desde el ordenador del cliente al ordenador del servidor, este procedimiento se detalla a continuación:

- El protocolo FTP al desarrollarse en entornos tipo UNIX les da acceso a los permisos de ejecución, lectura y escritura, estableciendo tres tipos de usuarios: Propietario (creador de archivo subido al FTP), Grupo (Usuarios que pueden pertenecer al propietario) y otros (usuarios anónimos o sin grupo).
- Para establecer los permisos de escritura existe un algoritmo, el cual asigna valores al tipo de acceso que se quiere otorgar a cada usuario, los cuales se asigna de acuerdo a la suma de los tipos ya descritos,
- En modo activo, el servidor creará el canal de datos en su puerto 20, mientras que el canal de datos del cliente se asocia a un puerto aleatorio mayor que el 1024; el cliente envía un comando PORT al servidor para poder abrir una conexión de datos donde se transferirán los archivos enviados.

2.16. Tecnología GSM

La tecnología GSM para (Espín Estévez & Maldonado Tinizaray, 2008), es calificado como un sistema de segunda generación, basado en el uso de celdas digitales, desarrollado para crear un sistemas para móviles que fuese compatible con los servicios existentes y futuros; la tecnología GSM ofrece movilidad internacional y una gran variedad de servicios (telefonía, mensaje de texto corto, fax).

Como lo señala (Velasco, 2005), GSM o Global Systemsfor Mobile communications tiene cuatro versiones principales fundamentadas en las bandas: GSM-850, GSM-900, GSM-1800 y 1900, siendo estas utilizadas en casi todo el mundo; fue creada por la CEPT³ y desarrollado por el ETSI⁴ más adelante, como un estándar para los teléfonos móviles europeos, con el objetivo de que esta se volviese una normativa adoptada en todo el mundo.

GSM (Espín Estévez & Maldonado Tinizaray, 2008), digitaliza y comprime los datos con información del emisor enviándolo a través de un canal, señala Las especificaciones GSM consisten en doce series redactadas en diferentes grupos de trabajo (GSM, ETSI), las cuales son detalladas en la

Tabla 5.

³ Conferencia Europea de Administraciones de Correos y Telecomunicaciones.

⁴ Instituto Europeo de Normas de Telecomunicaciones

Tabla 5.

Serie GSM

Serie	Contenido
01	Especificaciones generales
02	Características del servicio.
03	Características de la red
04	Protocolo e interfaz entre el terminal móvil y la estación base.
05	Capa física del canal radio.
06	Codificación de la información
07	Especificaciones del terminal móvil.
08	Interfaz entre la estación base y el terminal móvil.
09	Modo de funcionamiento de la red.
10	Modo de funcionamiento del servicio.
11	Aprobación de las especificaciones.
12	Operación y mantenimiento.

Tomado de artículo “El Sistema GSM”

2.17. Tecnología GPRS

En su artículo sobre GPRS (General Packet Radio Service), (Prieto, 2016), lo define como una tecnología que mejora las carencias presentadas por la tecnología GSM en un punto específico: la transmisión de datos, ingresando una red de conmutación de paquetes que funciona de manera paralela a la conmutación de circuitos de GSM, a su vez, no conlleva grandes inversiones y reutiliza parte de la arquitectura actual de GSM.

Como lo indica (Paz, 2009) una ventaja aportada por esta tecnología es que su aparición no es con el fin de eliminar o sustituir la tecnología GSM, sino

complementarla con los servicios que facilita la comunicación, permitiendo así, la personalización de las funciones, el desarrollo de juegos interactivos y la incorporación de aplicaciones para el intercambio de mensajes y correos electrónicos.

La arquitectura de la red GPRS, como lo indica (Wevar, 2005), es una red complementaria a GSM, lográndose a través de la introducción de dos nodos: GGSN y SGSN. GGSN actúa como un interfaz lógico hacia las redes de paquetes de datos externas como se observa en la Figura 14, conectándose a redes externas (internet o X.25); por su parte, SGSN⁵, es un nodo de conmutación de paquetes que se sitúa jerárquicamente al mismo nivel que las MSC, encargándose de transportar los paquetes de datos que se encuentran en su servidor, adicional a esto, detecta nuevos móviles GPRS en dicha área respaldando un registro de su ubicación.

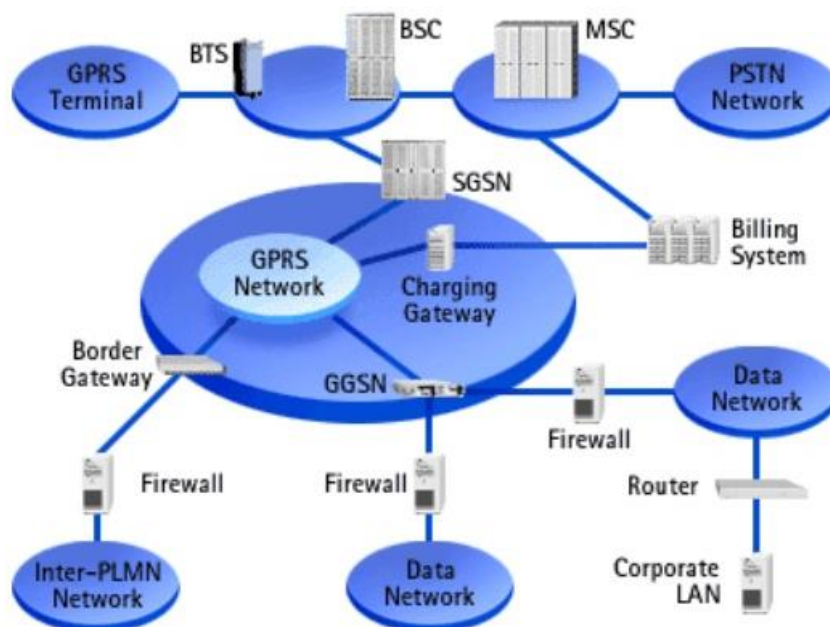


Figura 14. Arquitectura de la tecnología GPRS, tomada de sitio web Slideshare <https://es.slideshare.net/>.

⁵ Serving GPRS Support Node.

Capítulo III

Marco metodológico

3.1. Método de Investigación

En el presente capítulo se explicará el método que se tomó para la ejecución del trabajo de titulación. Se describirá y mostrará el desarrollo del sistema de control de cada una de las prácticas para ayudar a la comprensión del estudiante y aprendizaje de la electrónica y Telecomunicaciones.

3.1.1. Método experimental (Rodríguez, 2011).

Tiene como objetivo descubrir y exponer el funcionamiento de los objetos de experimentación con un nivel mayor de confianza, relaciones de tipo causal entre hechos y fenómenos de la realidad, siendo conceptualizado de la siguiente manera:

- Actuar reiteradamente, de forma planificada, mediante la manipulación de la variable independiente o experimental para determinar si hubo influencia o no sobre la variable dependiente.
- Aislar este proceso de la acción de las variables ajenas, ejerciendo un control que permita el análisis del fenómeno de forma pura.

El método experimental ejerce una influencia directa sobre el objeto de estudio bajo condiciones de control para determinar con exactitud dependencias y relaciones causales para obtener los datos que necesita el investigador para determinar los cambios y cumplir los objetivos.

Es así como el método experimental será el medio con el cual se demostrará la funcionalidad de un sistema de seguridad electrónica basado en la programación de microordenadores, sensores y actuadores en condiciones de controladas manipulando las variables independientes de las variables dependientes.

El presente proyecto de titulación tiene por finalidad realizar un sistema de sensores capaz de captar datos físicos del exterior, procesarlo mediante el uso práctico de microordenadores y mostrarlos en un sistema actuadores representadas en alarmas.

El conjunto de estos sistemas forma parte de la seguridad electrónica que se desea plasmar para el uso didáctico de los estudiantes de la carrera de Ingeniería electrónica y Telecomunicaciones, mediante la elaboración de un tablero comprensible para el usuario final.

Este conjunto de sistemas se lo puede explicar mediante el siguiente diagrama de bloques que está representado en la Figura 15 **Figura 15**, el cual identifica el sistema de sensores (cámaras, sensor de movimiento, sensor de temperatura y humedad, control de acceso), un sistema de almacenamiento y procesamiento

(microordenadores), sistema de actuadores (Módulo Gprs, Correo electrónico, mensaje de texto), y un sistema de ingreso de datos (Programación por teclado).

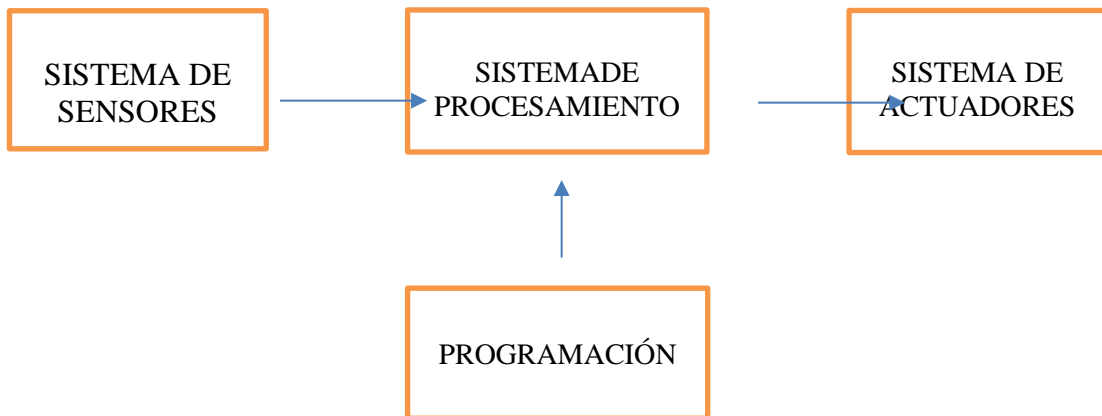


Figura 15. Diagrama de bloques del sistema de seguridad electrónica del módulo. Elaborado por autores.

3.2. Descripción de sensores en el estudio.

Se presentan el respaldo de los elementos seleccionados para el proyecto implementado en el área sensores de toma de muestras físicas.

3.2.1. Cámaras de seguridad

El principal objetivo de un sistema de seguridad es poder visualizar lo que esté ocurriendo en determinado ambiente debido a alguna inquietud con respecto a un movimiento o una alarma. Para cumplir con lo propuesto se plantea realizar un sistema de seguridad con cámaras que permitan visualizar mediante una dirección web la información que está siendo almacenada y procesada en el microcontrolador raspberry PI.

Por ello, se está seleccionando cámaras compatibles con el microcontrolador Raspberry Pi como lo son las cámaras 5MPX OV5647, Figura 16, la cual cuenta con un buz de dato de 15 cm para la conexión con la placa Raspberry mediante el puerto de cámara.

La cámara OV5647 funciona con un voltaje bajo, su sensor de imagen CMOS de 5 megapíxeles nos brinda un rendimiento alto en resolución y nos proporciona una salida de video de 2592x1944 mediante la tecnología OmniBSITM.

El núcleo del sensor OV5647 transmite datos de píxeles generales a una velocidad constante de fotogramas, indicadas por VSYNC⁶.

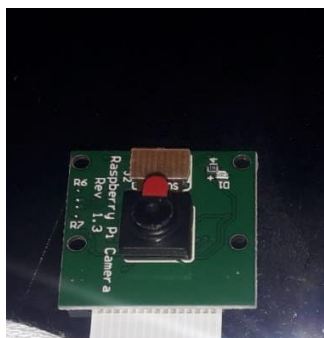


Figura 16. Módulo de cámara OV5647. Elaborado por autores.

3.2.2. Sensor de Temperatura y Humedad DTH 11(ELECTRONICS, 1995).

El sensor de temperatura y humedad DHT11 es un sensor digital dedicado a la tecnología de recolección y de detección de muestras físicas de temperatura y humedad, contiene una salida de señal digital, está conectado con un microcontrolador de 8 bits de alto rendimiento que da una alta confiabilidad y excelente estabilidad a largo plazo, respuesta rápida y anti-interferencias del medio o también llamado ruido.

Cada elemento DHT11 es estrictamente calibrado en los laboratorios para garantizar la precisión en las mediciones de Humedad y temperatura, su interfaz serial de comunicación por medio de un solo cable bidireccional genera una integración del sistema de forma rápida y sencilla.

Su bajo costo y excelente desempeño en la transmisión de la señal del medidor lo convierte en la mejor opción para diversas aplicaciones. Incluidas las más exigentes en el ambiente laboral.

Existen dos modelos de sensores DHT11, por un lado, el modelo del sensor integrado en una PCB que tiene 3 pines de conexión y al interno una resistencia pull-up de 5k Ohmios y el modelo normal con 4 pines de conexión de los cuales un pin no se utiliza.

Así es como el DHT11 y el DHT22 son dispositivos son herramientas de medición de microordenadores tales como Arduino y Raspberry, su comunicación se basa en la conexión mediante GPIO de los microordenadores y los pines de los sensores.

⁶ VSYNC: Sincronización vertical.

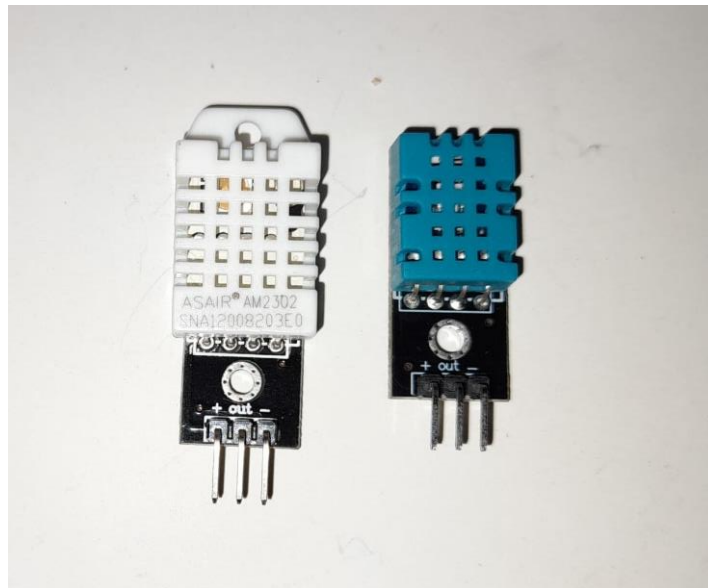


Figura 17. Modelos de sensor de temperatura y humedad DTH22 y DTH11. Elaborado por autores.

Los pines están identificados según sus fabricantes, pero el principio de funcionamiento es el mismo, un pin es VCC, un segundo pin es tierra GND, y el tercer pin es de salida digital en ambos modelos tanto como en PCB como el modelo original.

Su alimentación es de 3,5 V a 5V. su consumo es de 2,5 mA, su salida de señal es digital.

Su rango de medición de temperatura es de 0°C a 50°C con una precisión de resultado de $\pm 2^\circ\text{C}$ y nos da una resolución de 1°C (8 bit).

Su rango de medición de humedad es de 20% RH al 90% con una precisión de resultado de $\pm 5\%$ RH y nos da una resolución de 1%.

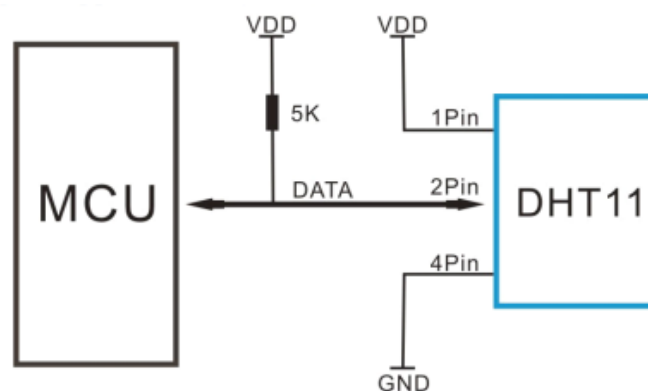


Figura 18. Diagrama de conexión de un sensor de temperatura DHT11 modelo estándar. Tomado de (ELECTRONICS, 1995).

3.2.3. Sensor de Movimiento PIR HC -SR501(ELECTRONILAB, 2011).

Para la implementación de la seguridad electrónica, es necesario la ejecución de sistemas de seguridad que nos ayuden a proteger espacialmente los espacios de monitoreo. Es decir, conocer novedades, alertas sobre anomalías de movimiento, detectar cambios de anormales en el espacio visualizado.

Para cumplir con este objetivo, surge como solución la implementación de un sistema de seguridad basado en la instalación de sensores de detección de infrarrojo pasivo que detentan cambios de temperatura por energía térmica que irradian naturalmente todos los seres vivos.

Su bajo costo, fácil instalación, hábil configuración y su posibilidad de adaptarse a sistema de control de microcontroladores tales como Raspberry Pi y Arduino, lo hacen la mejor opción al escoger un sistema de detección de movimiento.

El sensor de movimiento (PIR) “Pasive Infra Red” modelo HC-SR501, Figura 19, es un sensor piro eléctrico capaz de medir los niveles de radiación infrarroja emitida por cada cuerpo y objeto en un rango determinado de Máximo 7 metros (Figura 20).

- Es un sensor pasivo ideal para la detección de dentro de su franja de barrido.
- Su rango de operacional es de 4,5V – 20V.
- Tiene 3 pines, que se denominan VCC, OUTPUT y GND.
- Tiene 2 potenciómetro para ajustar los parámetros de sensibilidad y retardo. El cual puede configurarse entre 5 y 300 segundos (retardo), y 3 a 7 metros (sensibilidad).



Figura 19.Módulo de sensor PIR modelo HC-SR501. autores

3.2.3.1. Radio de inducción

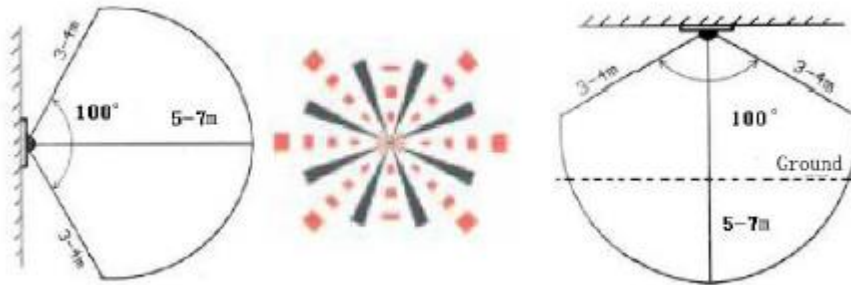


Figura 20. Radio de captación de infrarrojo del módulo HC-SR501. Tomado de (ELECTRONILAB, 2011).

3.2.3.2. Aplicaciones.

Sus aplicaciones pueden clasificarse en 4 grandes sistemas:

- Productos de seguridad
- En juguetes con sensores en el cuerpo
- En la detección de infrarrojos del cuerpo humano.
- Automatización y control industrial, etc.

3.2.4. Control de acceso con RFID.

En la actualidad entendemos como control de acceso al mecanismo de seguridad utilizado para restringir o permitir la entrada y salida de una persona o vehículo a determinada área, con el objetivo de la supervisión de la seguridad, registro de información y organización colectiva.

Si bien, lo más común para la aplicación de estos sistemas de control de acceso es el uso de la biometría, para el presente proyecto presentamos una solución técnica y confiable para la seguridad. Al utilizar sistema de Control de Acceso mediante identificación por Radiofrecuencia (RFID), se aprovechará su funcionamiento inalámbrico de almacenamiento y recuperación de datos que se basa en receptor ondas de radio en la cual se almacena información única de identificación (VARGAS VERGARA, 2013).

Como se observa en la figura 20, existen tres componentes básicos del sistema RFID:

1. El tag o etiqueta o transportador RFID.
2. El lector o transceptor RFID.
3. El controlador.

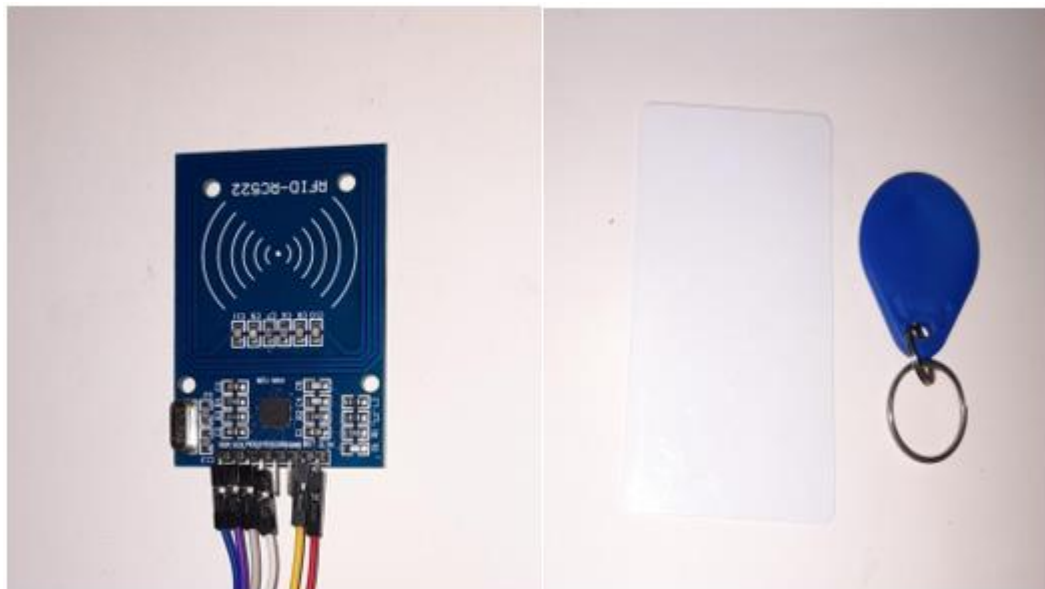


Figura 21. Tag transportador RFID, llavero de acceso y lector transceptor RFID.
Autores

La radiofrecuencia es capaz de atravesar objetos por lo que no necesita una línea de visión directa entre el emisor y el receptor como lo necesita los sistemas infrarrojos.

Para el presente proyecto utilizaremos las herramientas del Módulo RFID RC522 el cual es compatible con microcontroladores tales como Arduino Uno y raspberry Pi mediante un bus de interfaz periférico serial (SPI) que nos permite la lectura y escritura sobre este sistema de seguridad por radiofrecuencia.

La etiqueta o tag no necesita alimentación eléctrica, su funcionamiento se basa en el método de la inducción, se lo conoce como tarjetas pasivas, en cambio, el lector necesita estar conectado eléctricamente para emitir un campo electromagnético con un determinado rango dependiendo de la aplicación que se le dará al sistema de seguridad.

Operan en la banda de radiofrecuencia ELF (Frecuencia extremadamente baja), y se pueden utilizar en varias frecuencias

Tabla 6.

Aplicación de RFID

Tipo de sistema	Rango de operación	Aplicaciones	Objetivo
Baja Frecuencia o LF	125-134KHZ	Control de acceso	Leer objetos con alto contenido de agua.
Alta frecuencia o HF	13,56 MHZ	Marketing	Tarjetas, pagos, aplicaciones de marketing.
Frecuencia Ultra Alta UHF	433, 860 y 960 MHZ	Sistemas de control	Controlar artículos de almacenes, conteo de personas, control en carreteras.

Elaborado por Autores.

3.2.1. Control de acceso con Lector de huella digital RC305

Los sistemas de seguridad por control de acceso con huella digital se utilizan con frecuencia para aumentar la restricción de acceso a personas no identificadas a un área de una demanda de protección alta.

El lector de huellas dactilar es un dispositivo electrónico que identifica huellas digitales con una calidad de precisión muy buena, como se muestra en la Figura 22, tiene en su parte frontal una superficie sólida, que a pesar de su imagen es resistente en general.

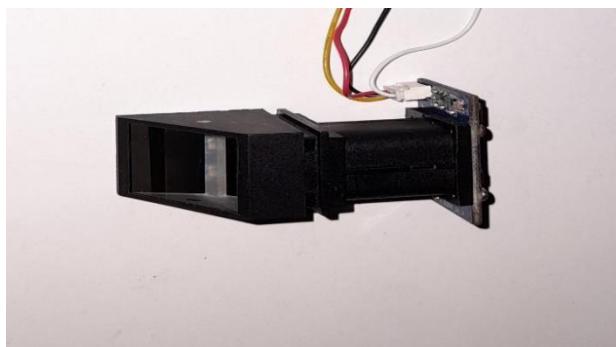


Figura 22.Módulo de identificación de huella digital RC305.Elaborado por autores

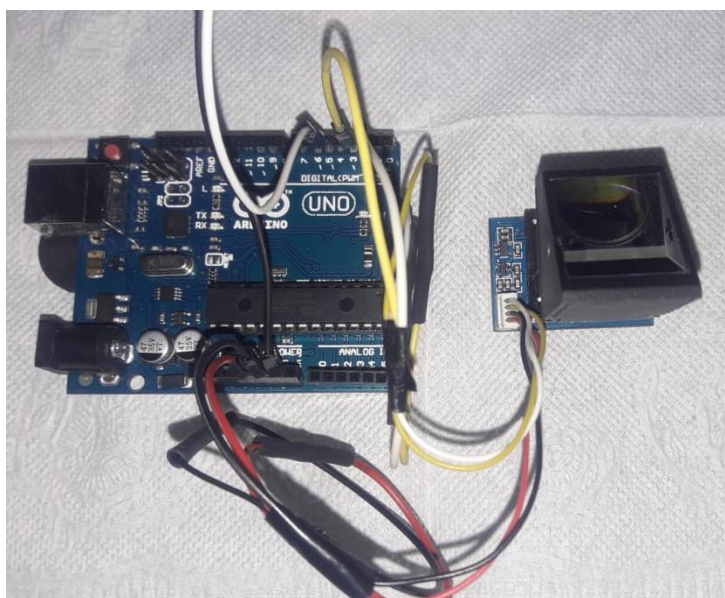


Figura 23.Conexión de Arduino uno y módulo de identificación de huella digital RC305. Elaborado por autores.

Su funcionamiento permite el análisis de las huellas ingresadas mediante el microcontrolador Arduino, cuenta con un procesamiento digital de las imágenes interna o también llamado DSP⁷.

Para el uso de este dispositivo de manera correcta, se debe conectar a los pines del Arduino TX y RX para el manejo de las señales captadas, como se muestra en la Figura

⁷ DSP: Procesador de Señales Digitales.

23, se debe energizar con el voltaje del microcontrolador mediante los pines de salida 5V y GND.

Para el funcionamiento se debe grabar en la base de datos del módulo lector de huellas las credenciales que podrán acceder, se les asigna un ID a cada huella dactilar y se programa para que realice una secuencia de tipo rastreo y comparativo, la cual nos indique si es permitido el acceso o denegado para el usuario que desee ingresar al área restringida por este control de acceso.

3.3. Descripción de sistemas Actuadores

Se presentan el respaldo de los elementos seleccionados para el proyecto implementado en el área alarmas y muestra de resultados.

3.3.1. Sistemas GSM/GPRS(Federico Nicola, 2015).

Los sistemas GSM/GPRS son los responsables de la transmisión de información entre módulos inalámbricos. Gracias a esta tecnología se ha podido potenciar los sistemas de seguridad para que funcionen de forma inalámbrica, segura y rápida, abaratando los costos y permitiendo la flexibilidad para poder ser instalados según la necesidad. Por ejemplo: su aplicación GSM nos ayuda a identificar alteraciones en las alarmas de seguridad en domicilios, principalmente utilizadas en entradas como puertas, ventanas al enviar mensajería cifrada por SMS mediante su conexión GPRS con los dispositivos del usuario final o propietario que los esté controlando ya sean dispositivos móviles, computadoras, mediante llamadas y mensajes.

3.3.2. Módulo SIM800L GSM/GPRS(López Eusebio, Victor Ernesto Gutiérrez Torres, 2017).

La innovación de la tecnología ha obligado a simplificar y unir estas dos tecnologías con la creación de una PCB capaz realizar todas las funciones que normalmente se harían por separado.

Como se muestra en la Figura 24, el módulo SIM800L GSM/GPRS es un modem en miniatura, una PCB que nos ayuda a entender de forma práctica el principio de la domótica y el funcionamiento de IOT (internet de las cosas), gracias a su arquitectura nos permite realizar la interconexión digital de los objetos cotidianos mediante las redes Inalámbricas sin necesidad de la operatividad de un humano.

El módulo SIM800L se asemeja en funcionamiento a los teléfonos móviles ya que posee la capacidad de generación y envío de mensajes de texto SMS, nos permite realizar y recibir llamadas telefónicas, conectarse a internet mediante el uso de protocolo GPRS o TCP/IP.

Su cobertura es a nivel mundial debido a que posee un módulo de red que soporta la banda cuádruple: (GSM850, EGSM900, DCS1800 Y PCS1900).

Nos permite la conexión a internet mediante la red global GSM admitiendo tarjetas Sim y microSim mediante la ranura en su parte posterior, las mismas que abren el camino para la comunicación desde la tecnología 2G hasta la 4G en la actualidad.

Funcionan con una alimentación de voltaje de 3,4V – 4,4V. Utiliza una antena soldable en la PCB modelo Helicoidal para conectarse a la red.

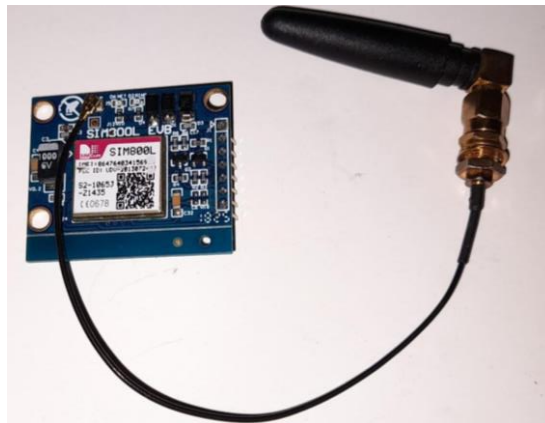


Figura 24. Módulo GSM GPRS SIM800L. Elaborado por autores.

3.3.2.1. Características del módulo SIM 800L

- Soporta las bandas de frecuencia en MHz 850, 900, 1800 Y 1900.
- Conecta a cualquier Red Global GSM con cualquier SIM 2G.
- Capaz de hacer y recibir llamada de voz gracias a un altavoz externo de 8 Ohmios y un micrófono electrolet.
- Capaz de enviar y recibir mensajería GSM.
- Capaz de enviar y recibir datos GPRS mediante protocolos TCP/IP, HTTP,
- Capaz de escanear y recibir emisiones de Radio FM.
- Ranura para leer cualquier tarjeta Sim y microSim.

3.3.2.2. Componentes. El módulo contiene un total de 12 pines que lo conectan al mundo exterior:

1. NET es el pin de la PCB donde se debe soldar la antena helicoidal.
2. VCC es el pin que suministra la energía para el módulo, puede ser de 3,4 a 4,4V.
3. RST (reset) es un pin de reajuste duro.
4. RXT (receptor).
5. TXT (transmisor).
6. GND es el pin de tierra y necesita ser conectada al microcontrolador Arduino.
7. RING Actúa como indicador de anillo.

8. DTR activa o desactiva el modo de espera.
9. MIC es una entrada de micrófono diferencial.
10. SPK es una interfaz de altavoz diferencia.

3.3.3. BUZZER(Alexander et al., 2012).

Es un dispositivo capaz de convertir las señales eléctricas en sonido, como se aprecia en la Figura 25, el Buzzer tiene dos cables de conexión que sobresalen de su gabinete de fabricación, estos cables representan la polaridad del dispositivo los cuales son GND y VCC que se conectan a cualquier microcontrolador ya sea Raspberry o Arduino.

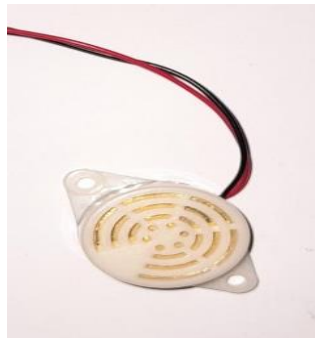


Figura 25. Módulo Buzzer Piezoeléctricos. Elaborado por autores.

3.3.4. Módulo WIFI ESP8266(Ochoa, 2017).

El móduloESP8266 es un dispositivo que contiene un chip integrado con accesibilidad Wifi, su diseño fue pensado para ser compatible con el protocolo TCP/IP, y es capaz de dar acceso a cualquier microcontrolador a una red LAN. Como se muestra en laFigura 26, el módulo ESP8266 contiene 8 puertos de GPIO (entradas y salidas de propósitos general), las cuales se identifican en dos filas de 4 puertos.

Primera fila superior de izquierda a derecha: GND – GPIO2 – GPIO0 – RXD.

Segunda fila inferior de izquierda a derecha: TXD – CH_PD – RESET – VCC.



Figura 26.Módulo wifi ESP8266. Elaborado por autores.

3.3.5. Módulo RELE(Bolaños, 2016).

Un dispositivo electrónico RELE es un switch mecánico que opera recibiendo señales eléctricas que pueden tener dos estados encendido o apagado, dando paso a la corriente eléctrica o no de ser el caso, su diseño fue pensado para poder ser controlado con voltajes bajos para que sea compatible con los microordenadores del mercado, ya sea Arduino o Raspberry, debido a que puede operar con 5V.



Figura 27. Módulo RELE. Elaborado por autores.

En la Figura 27, se puede apreciar que el módulo RELE cuenta con 7 entradas y 6 salidas.

Las cuales se identifican de la siguiente forma:

Entradas	Salidas
+5V lógica	Normalmente Abierto
Entrada 2	Común
Entrada 1	Normalmente Cerrado
GND	Normalmente Abierto
GND	Común
+5V lógica	Normalmente Cerrado
+5V Relés	

3.3.6. Cerradura Eléctrica(Coba Gibbs & Samaniego González, 2017).

Es un dispositivo electromagnético que permite según su aplicación el cierre o apertura de un área determinada mediante el uso de señales eléctricas, como se aprecia en la Figura 28 **Figura 28**, la cerradura eléctrica tiene dos cables de conexión que sobresalen de su gabinete de fabricación, estos cables representan la polaridad del dispositivo los cuales son GND y VCC que se conectan a cualquier microcontrolador ya sea Raspberry o Arduino.



Figura 28.Cerradura eléctrica. Elaborado por autores.

3.3.7. Pantalla LCD 16X2 1602A.

Es un dispositivo electrónico que contiene un display alfanumérico con 2 filas de 16 caracteres cada una, Cuenta con 16 pines que son compatibles con los microcontroladores del mercado, Raspberry Pi y Arduino, su diseño fue pensado para poder ser controlado con voltajes bajos debido a que puede operar con 5VDC, como se observa en la Figura 29 **Figura 29**.



Figura 29.Pantalla LCD 16x2. Elaborado por autores.

3.4. Descripción de sistema de procesamiento

Se presentan el respaldo de los elementos seleccionados para el proyecto implementado los cuales serán los responsables del almacenamiento de la toma de muestras y el procesamiento de las salidas con resultados.

3.4.1. Microcontrolador Arduino.

El microcontrolador Arduino nos permitirá la realización del presente proyecto ejecutando la función de procesador para nuestro sistema de seguridad electrónica en el área de control de acceso. Gracias a su compatibilidad en su arquitectura y conectividad en su tecnología con el módulo RFID permite la lectura de los datos que puede generar el lector RFC522 procesarlos y almacenarlos.

Gracias a su compatibilidad con el Módulo Sim800L podremos ejecutar las salidas digitales mediante su aplicación GPS/GPRS que su función en este proyecto será como actuador o medio de salida para alarmar cuando el sistema de sensores emita una alerta que de intento erróneo de acceso.

3.4.1.1. Funcionamiento.

La placa microcontroladora se inicia mediante la conexión de una fuente de 5V y para poder efectuar trabajos se da por medio de programación mediante la aplicación Arduino, para poder realizar diversos proyectos es necesario realizar las descargas de las diversas librerías que nos da el aplicativo, para de esa manera enlazar todos los dispositivos que se pueden usar en el equipo como se puede apreciar en la Figura 30.

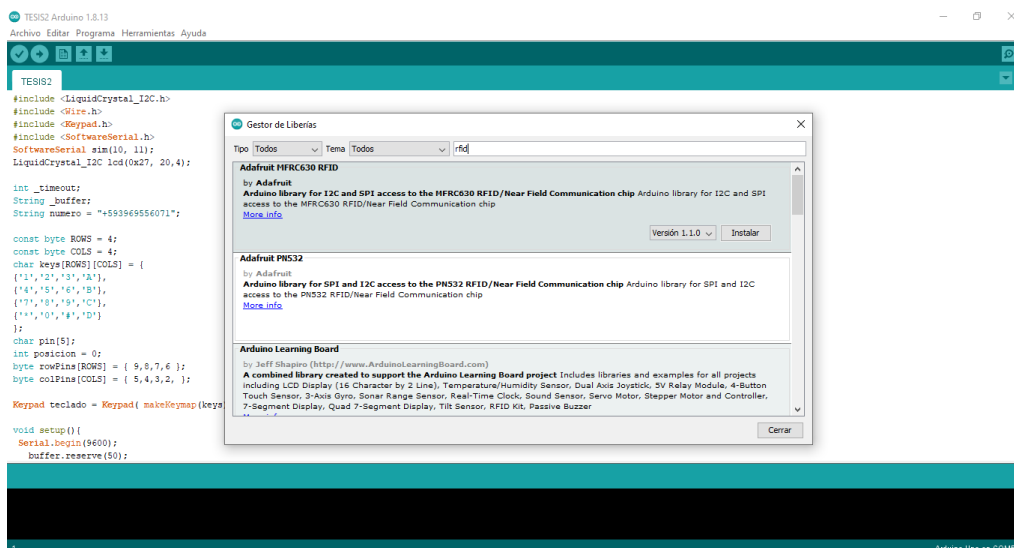


Figura 30. Pantalla IDE (Entorno de desarrollo integrado) Arduino. Elaborado por autores.

El uso de un microcontrolador Arduino beneficia al programador, ya que su arquitectura de placa SBC nos ayuda a poder optimizar componentes, al brindarnos salidas analógicas, digitales y de alimentación podemos vincular otros dispositivos sin necesidad de reguladores de voltaje adicionales, se puede apreciar en la Figura 31. Su gran funcionabilidad se la debe al que posee circuitos integrados que permiten grabar instrucciones, los cuales se escriben en lenguaje de programación que utiliza Arduino IDE.

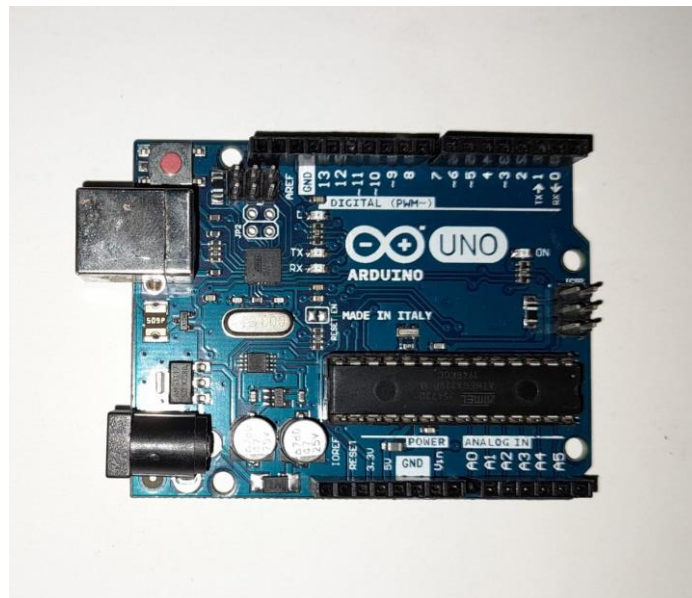


Figura 31. Placa de Arduino Uno. Elaborado por autores.

En seguridad electrónica, es muy útil ya que permite realizar procesos con otros dispositivos que son usados como sistemas de alarmas por ejemplos módulos wifi para conectarse a una red y emitir alertas mediante mensajes de correos electrónicos, esta placa es muy eficaz para trabajar en diferentes ambientes, ya que permite trabajar como

sistemas de alertas, sistemas de automatización, ya que cuenta con comunicación UART, permitiendo enviar comandos AT.

3.4.1.2. Conexión de Arduino uno con Módulo RFID RFC522.

La conexión se da mediante los puertos de entrada digitales y fuente de alimentación, para que el módulo RFID RFC522 comience a operar el cual tiene un alcance de detección de 5 a 7 cm como lo muestra en la Figura 32.

Figura 32.

Figura 32. Conexión de Arduino y tarjeta RFID. Elaborado por autores.

El cable negro que sale de los pines del módulo RFID representa la alimentación 3.3V desde la placa Arduino a la tarjeta RFID, los siguientes cables blanco (Pin 9), cable amarillo (Pin 12), cable naranja (Pin 11), cable rojo (Pin 13), cable café (Pin 10), están conectadas a los pines de propósito general del microcontrolador y a los pines del módulo.

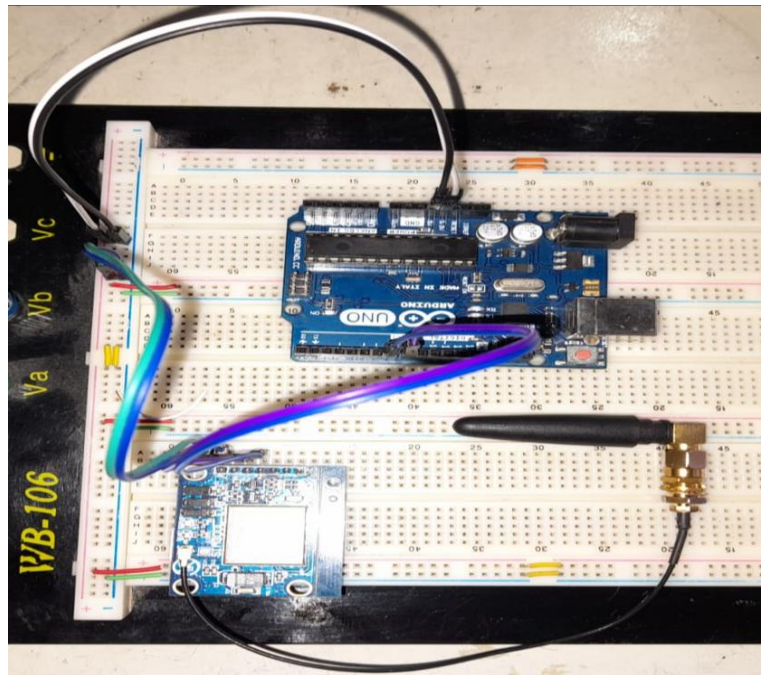
Una vez realizada las conexiones físicas de los pines del lector RFID, se ingresará la programación mediante el IDE del microcontrolador Arduino para las configuraciones de control de acceso sobre las tarjetas, llaveros de acceso y la base de datos del lector RFID.

3.4.1.3. Conexión de Arduino uno con Módulo Sim800L.

El módulo Sim 800L, es módulo de telefonía móvil GSM/GprsQuad-Band, que opera en bandas de 850, 900, 1800, 1900 MHz, gracias a su diseño nos permite añadir voz y datos a la diversidad de proyectos en la que se lo implemente.

Este módulo es utilizado en sistemas de IOT para el envío de alertas de equipos remotos, sistemas de seguridad caseros, enviar datos de telemetría, aplicaciones con tecnología M2M, automatización industrial, domótica.

Para la vinculación entre un microcontrolador Arduino y un Módulo Sim800L se realizan las conexiones físicas del módulo Arduino a un



tomacorriente de 110v con un convertidor a voltaje DC el cual utiliza un voltaje de 3.4 a 4.4 V, se conecta la antena del módulo SIM800L, se inserta la Simcard comenzara a parpadear 2 segundos el led que indica la conexión a la red, una vez el módulo se conecte correctamente parpadeara cada 5 segundos el led, se realiza el cableado con jumper desde los pines de Tx y Rx del módulo Sim800L de comunicación a unos de los pines digitales para tener comunicación AT como se muestra en la Figura 33.

Figura 33. Conexión de Arduino y módulo Sim800L. Elaborado por autores.

3.4.2. Sistema de Seguridad electrónica aplicando IOT con una placa de Arduino un módulo RFID y un módulo Sim800L.

Una vez explicado cómo funciona los dispositivos y sensores se puede realizar un sistema de seguridad que nos permita controlar el acceder a cualquier área mediante el uso de lectores electromagnéticos como las de tarjetas de acceso.

Como ya es de conocimiento general, este primer paso lo realizamos con la placa Arduino y la conexiones con la tarjeta RFID RFC522, programar en el IDE⁸ de Arduino todos los pines a usar y establecer el código fuente a la tarjeta que tendrá permitido el acceso y otras tarjetas no tendrán, para completar este sistema de seguridad debemos añadir la tecnología IOT mediante el envío inalámbrico de alertas mediante la utilización un módulo Sim800L el cual al recibir la señal de un intento de acceso erróneo lo tomara como señal de intruso y notificará al operador encargado por medio de mensaje SMS de que se está vulnerando la seguridad del Data Center y puede ir a esa área y dar seguridad preventiva.

Adicional a este servicio de mensajería se puede implementar el uso de un módulo wifi, para realizar una notificación mediante el uso del servicio de internet, el módulo podrá enviar la alerta mediante correo electrónicos a los encargados de seguridad del área.

En la Figura 34, se observa la estructura del sistema de control en una protoboard de prácticas.

⁸ I.D.E.: Entorno de Desarrollo Integrado.

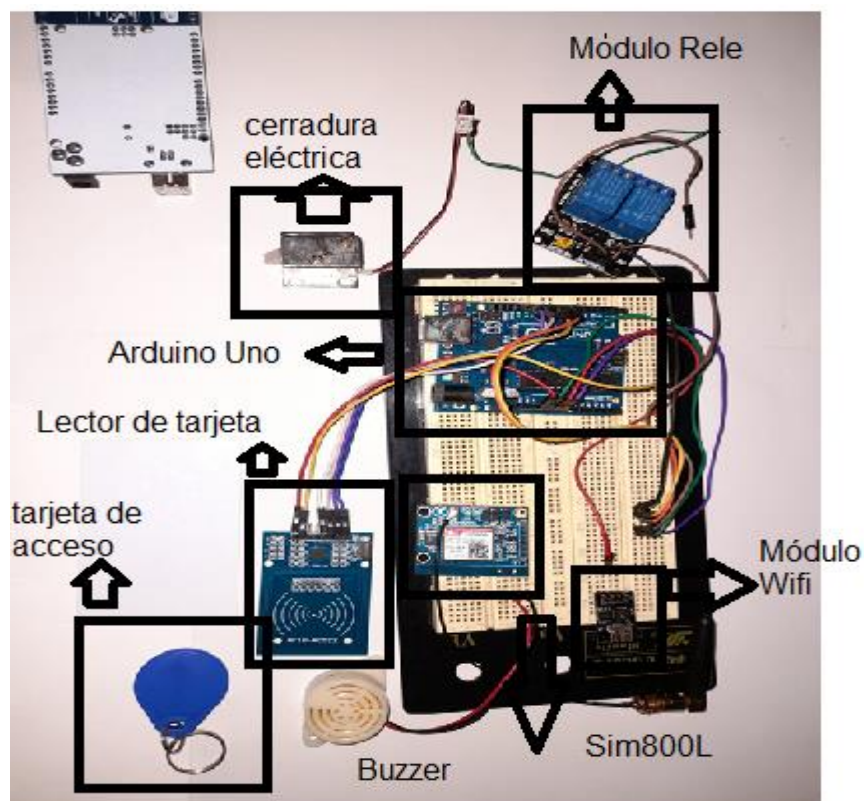


Figura 34. Ejecución de sistema de seguridad de control de acceso con alarmas mediante buzzer y módulo Sim800L en una protoboard. Elaborado por autores

3.4.2.1. Aplicación de Arduino en control de acceso.

A continuación, se muestra en la **Figura 35**, el diagrama de bloques del sistema de control de acceso conformado con un sensor representado por el módulo RFID modelo RFC522, lector de huella e ingreso por teclado, mediante un procesador representado por el microordenador Arduino Uno, y una salida representada por el módulo Sim800L utilizando la red GSM para alarmas mediante SMS, módulo wifi alarmas por correo, buzzer y led alarma visible.

Muestra
mensaje
Bienvenido
Se activa
cerradura



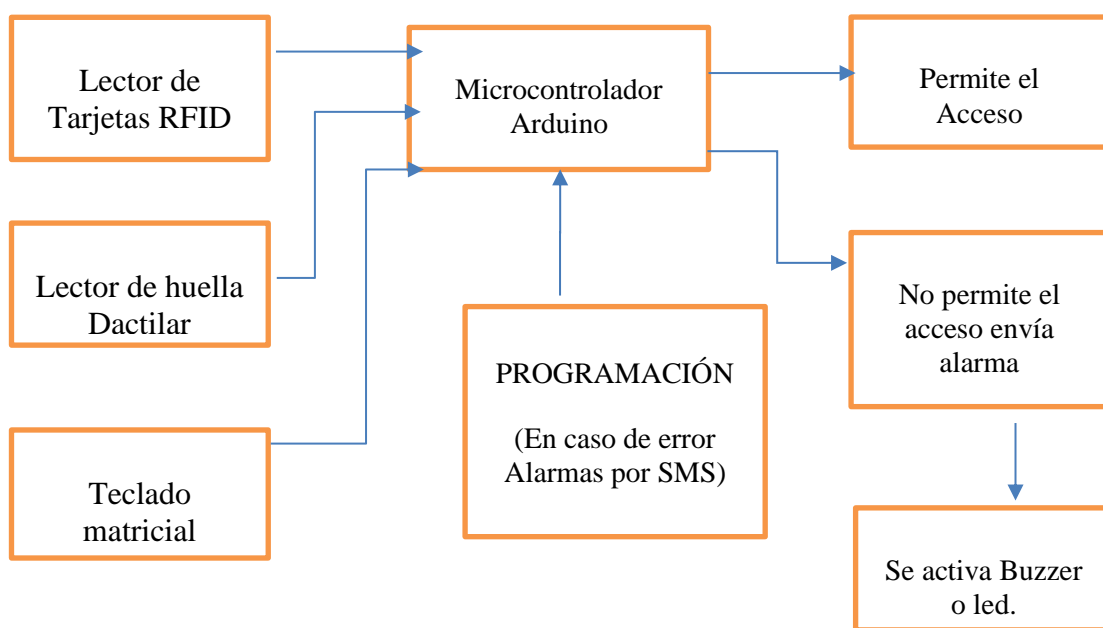


Figura 35. Diagrama de bloque funcionamiento de control de acceso con microcontrolador Arduino. Elaborado por autores.

3.4.3. Microcontrolador Raspberry Pi modelo 4B.

El microcontrolador Raspberry Pi modelo 4B será nuestra llave maestra para la implementación de IOT (internet de las cosas) del presente proyecto ejecutando la función de procesador para nuestro sistema de seguridad electrónica en el área de control de monitoreo, rastreo, habilitación de sensores y actuación de alarmas.

El modelo 4B nos brindará una mayor su funcionabilidad debido a que su potencia que se asemeja a una PC para las funciones de procesamiento de entradas y salidas de periféricos.

En el presente proyecto utilizaremos todas los puertos de ingreso y salida que nos brinda esta placa SBC, desde los puertos USB 3.0 y 2.0 hasta los pines GPIO, como se muestra en la Figura 36.

Su diseño de arquitectura, nos permite utilizarlo en sistemas de seguridad de circuito cerrado de televisión o en aplicaciones de cámaras inteligentes, esto es debido a que su tecnología adapta una entrada para mini-cámaras en full HD que lo podemos programar en base a un lenguaje de código abierto en específico Linux.

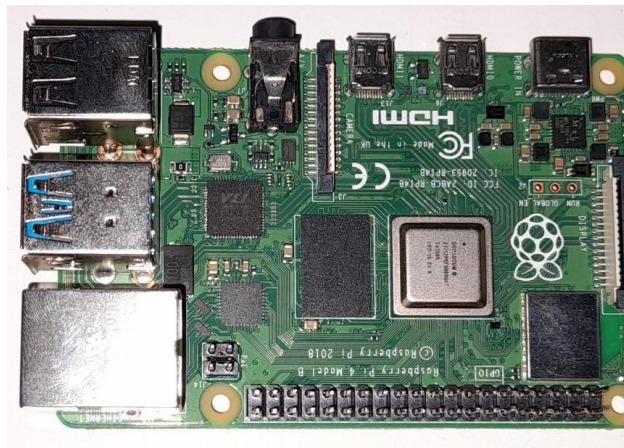


Figura 36.Placa de Raspberry PI Modelo 4B. Elaborado por autores.

Sus filas de pines de GPIO (entradas y salidas de propósitos general), nos permiten vincular todos nuestros sensores a utilizar en el proyecto, ya sea nuestro sensor de movimiento, sensor de temperatura, sensor de humedad, además de permitirnos seleccionar alarmas según sea la demanda, estas pueden ser alarmas físicas (encendido de led, encendido de sonidos), o virtuales (correo electrónico, notificaciones en pantalla). cualquiera de estos 40 pines lo podemos modificar y programar mediante el software de Raspberry pi OS para distintos propósitos que el autor desee. A continuación, detallamos cada uno de los pines en la Figura 37.

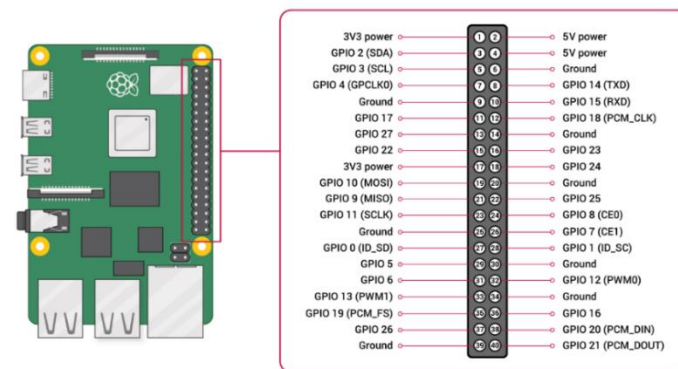


Figura 37.GPIO Raspberry Pi 4 modelo B, tomado de <https://www.raspberrypi.org/documentation/usage/gpio/>

3.4.3.1.Funcionamiento.

La placa SBC de Raspberry Pi 4B se inicia con una fuente de alimentación de 5V y 3A, es importante antes de encenderla haber cargado el sistema operativo con el que se va a configurar el microcontrolador, este sistema operativo debe estar grabado en la tarjeta SD e insertado en la ranura de la Raspberry antes de encenderla.

El sistema operativo para utilizar lo puedes descargar directamente desde la página oficial de Raspberry Pi <https://www.raspberrypi.org/>, este procedimiento se lo puede visualizar en la Figura 38, una vez realizado este paso inicial se debe conectar un teclado, mouse a las ranuras de USB de la placa SBC y una pantalla o monitor a la ranura micro-HDMI por medio de una conexión cableada.

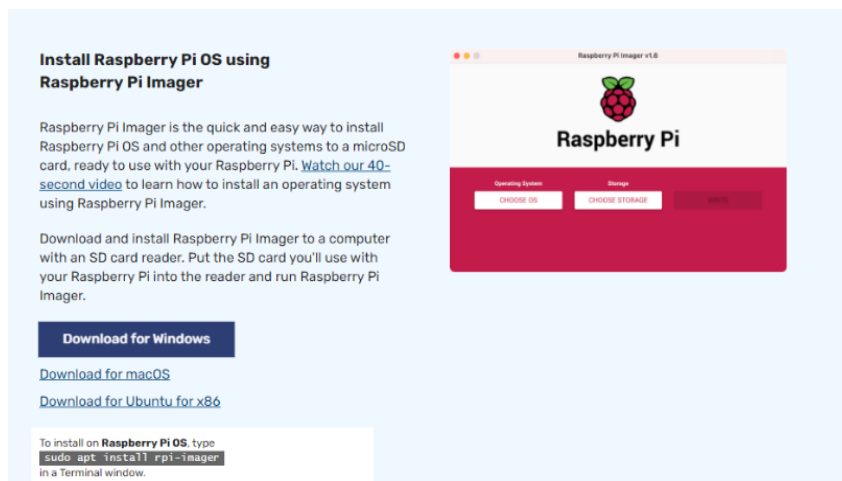
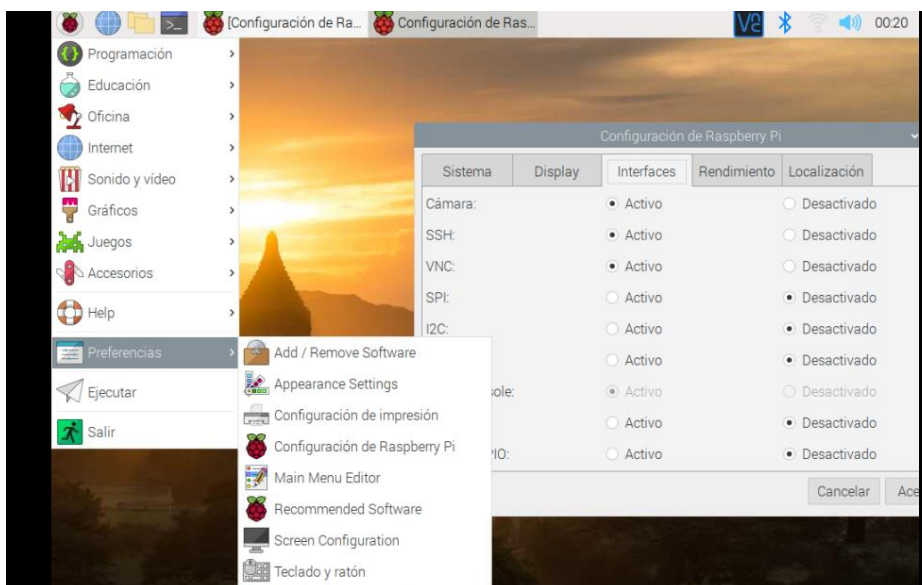


Figura 38. Descarga de Sistema Operativo de página oficial Raspberry Pi. Elaborado por autores.

Una vez listo, todo es conectado a un tomacorriente de 110V a una fuente de alimentación de 5v y 3A, es importante verificar que la fuente otorgue la corriente solicitada debido a que si no la cumple afecta directamente el rendimiento.

Para facilitar el uso de los microcontroladores Raspberry Pi es necesario permitir la comunicación de forma remota, como se muestra en la Figura 39, para ello la placa SBC dentro de su configuración nos ofrece conexión mediante SSH y VNC, los cuales nos permiten trabajar con el equipo con el requisito único de la dirección IP del dispositivo Raspberry Pi.



Figura

39.Habilitación de interfaz SSH y VNC en IDE Raspberry Pi. Elaborado por autores.

3.4.3.2. Conexión de Raspberry Pi 4B con una microcámara Full HD.

Para realizar una conexión entre una cámara Full HD y una placa SBC de Raspberry Pi 4B, se debe conectar la cámara mediante un bus de cable plano de 15 cm fijado a las ranuras del dispositivo hasta el puerto de cámara MIPI CSI de 2 carriles del microcontrolador como se muestra en la **Figura 39**.



Figura 40.Conexión física de CámaraOV5647 y Raspberry Pi. Elaborado por autores.

Bien, esto es en cuanto a conexión física, ya que si se procede a encender la Raspberry Pi y se visualiza mediante la conexión remota VNC no se tendrá visualización de la cámara hasta que se permita el uso de este puerto en la configuración interna del dispositivo.

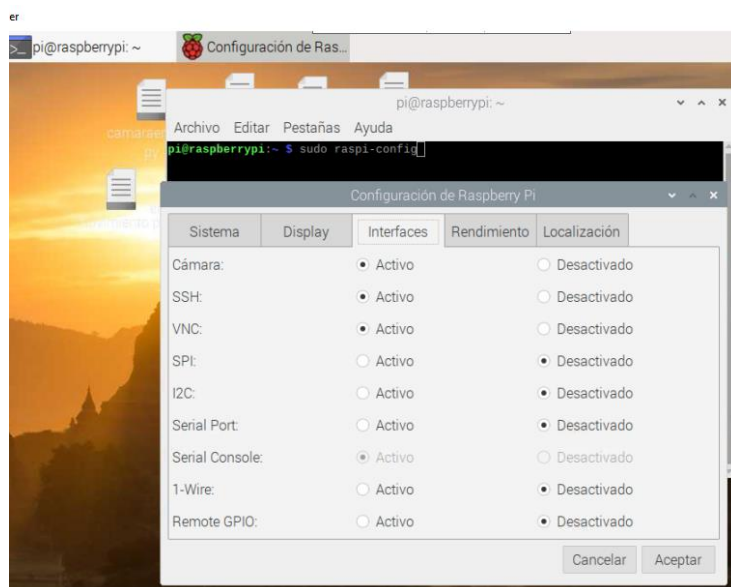
La habilitación de la cámara se realiza en el sistema operativo de la Raspberry Pi 4B, iniciando sesión y dirigiéndose a configuraciones, preferencias, configuraciones de la

Raspberry Pi, pestaña interfaz y se habilita la cámara, para guardar y ejecutar se debe reiniciar la placa de Raspberry Pi.

También se puede proceder escribiendo en la ventana de la terminal el siguiente código:

-Sudo raspi-config

Como se muestra en la Figura 41, se despliegan varias ventanas, se selecciona la opción de interfaz, puerto de cámara y se selecciona habilitar, guarde los cambios, si el programa le pide reiniciar “reboot” ejecute el sí.



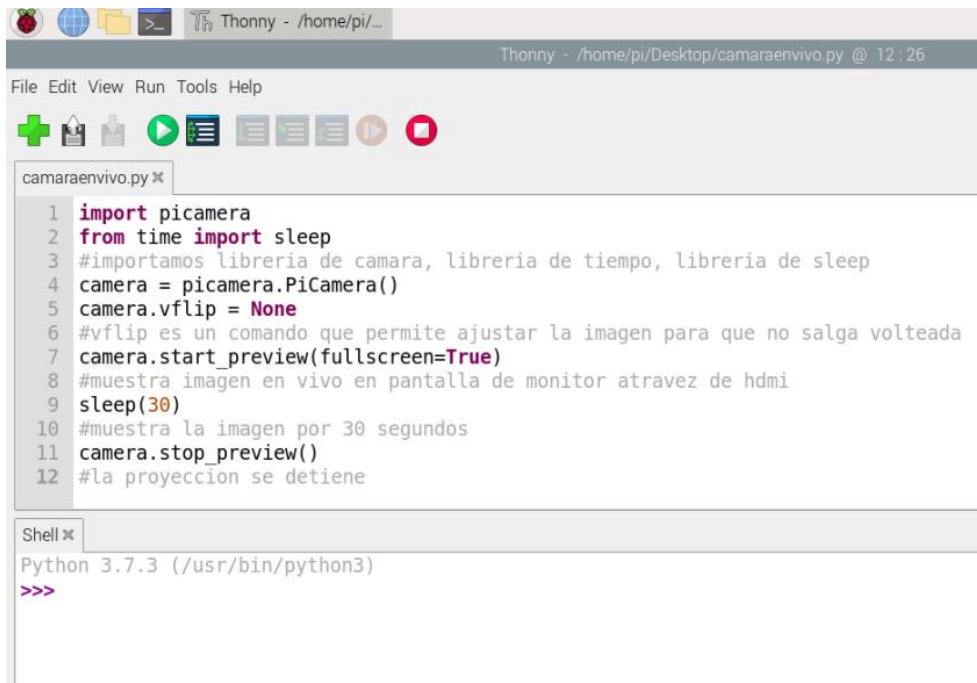
Figura

41.Habilitación de la cámara en IDE Raspberry Pi. Elaborado por autores.

El presente módulo de cámara Full HD es ideal para realizar sistemas de seguridad de video vigilancia debido a su bajo costo y fácil instalación, además, su tecnología cuenta con una excelente calidad de imagen y video y una conexión inalámbrica mediante app móvil propias de la cámara.

Una vez habilitada la conexión a la cámara mediante la configuración en la plataforma, se deberá realizar un código de prueba en el terminal de programación de la raspberry, con el código se podrá visualizar la calidad de la cámara y su funcionalidad para el proyecto, al ser una cámara tan pequeña, pero con una calidad excelente nos permite esconderla y realizar un sistema de seguridad espía.

Para poder iniciar la programación, el usuario debe seleccionar con el puntero del mouse sobre el icono de inicio de raspberry y seleccionando la aplicación Thonny Python que es nuestra plataforma de programación. En la Figura 42 se describe el código de la captura de imagen.



```
Thonny - /home/pi/...
Thonny - /home/pi/Desktop/camaraenvivo.py @ 12:26
File Edit View Run Tools Help
+ [Icons]
camaraenvivo.py ✕
1 import picamera
2 from time import sleep
3 #importamos libreria de camara, libreria de tiempo, libreria de sleep
4 camera = picamera.PiCamera()
5 camera.vflip = None
6 #vflip es un comando que permite ajustar la imagen para que no salga volteada
7 camera.start_preview(fullscreen=True)
8 #muestra imagen en vivo en pantalla de monitor atravez de hdmi
9 sleep(30)
10 #muestra la imagen por 30 segundos
11 camera.stop_preview()
12 #la proyección se detiene

Shell ✕
Python 3.7.3 (/usr/bin/python3)
>>>
```

Figura 42. Código Fuente para capturar imágenes con Raspberry Pi. Elaborado por autores.

Una vez escrito el código en el programa, el usuario debe seleccionar el botón de comando RUN que está identificado con el símbolo de “PLAY” dentro de un círculo color verde, antes de realizar el proceso, “Thonny Python” nos pedirá guardar el código con un nombre y dominio(xxx.py) antes de ejecutarlo.

Como se podrá apreciar en la

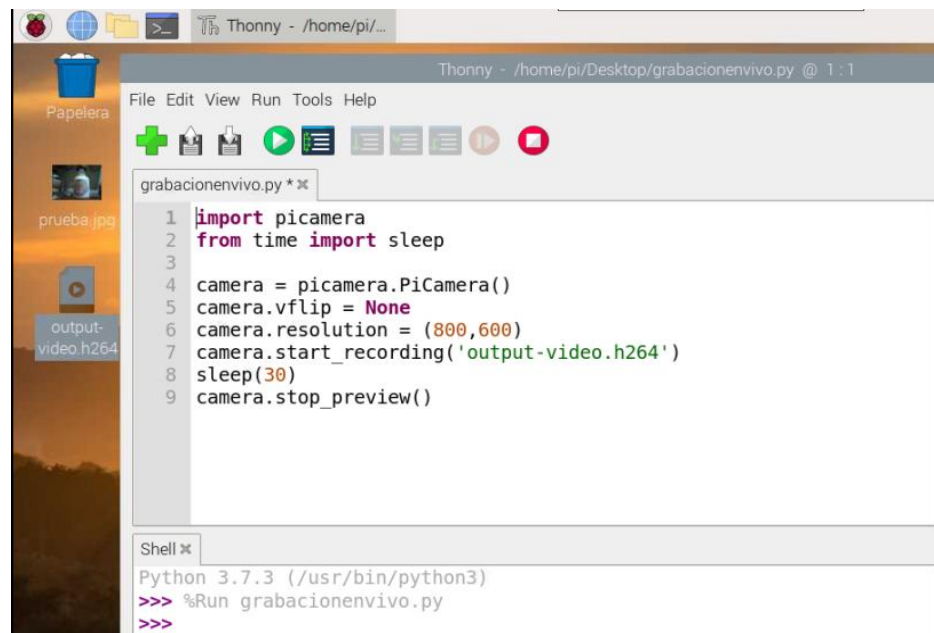
Figura 43,el



microcontrolador Raspberry Pi ha inicializado el código y ha dado la orden a la cámara de capturar la imagen en el momento exacto que se ejecutó. La imagen se encontrará guardada dentro de la plataforma virtual de Raspberry Pi OS o se visualiza en la pantalla como escritorio, dando doble clic sobre la imagen guardada con el nombre establecido en el código el ambiente virtual la proyecta en la pantalla y el usuario podrá comprobar la funcionalidad de este.

Figura 43.Imagen captada con cámara HD y microcontrolador Raspberry Pi. Elaborado por autores.

Ahora bien, esto es para capturar imagen, pero poder realizar el grabado de video se sigue el mismo proceso, pero se realiza una modificación en el código inicial. En la Figura 44, se describe el código de la grabación de video.



```

1 import picamera
2 from time import sleep
3
4 camera = picamera.PiCamera()
5 camera.vflip = None
6 camera.resolution = (800,600)
7 camera.start_recording('output-video.h264')
8 sleep(30)
9 camera.stop_preview()

```

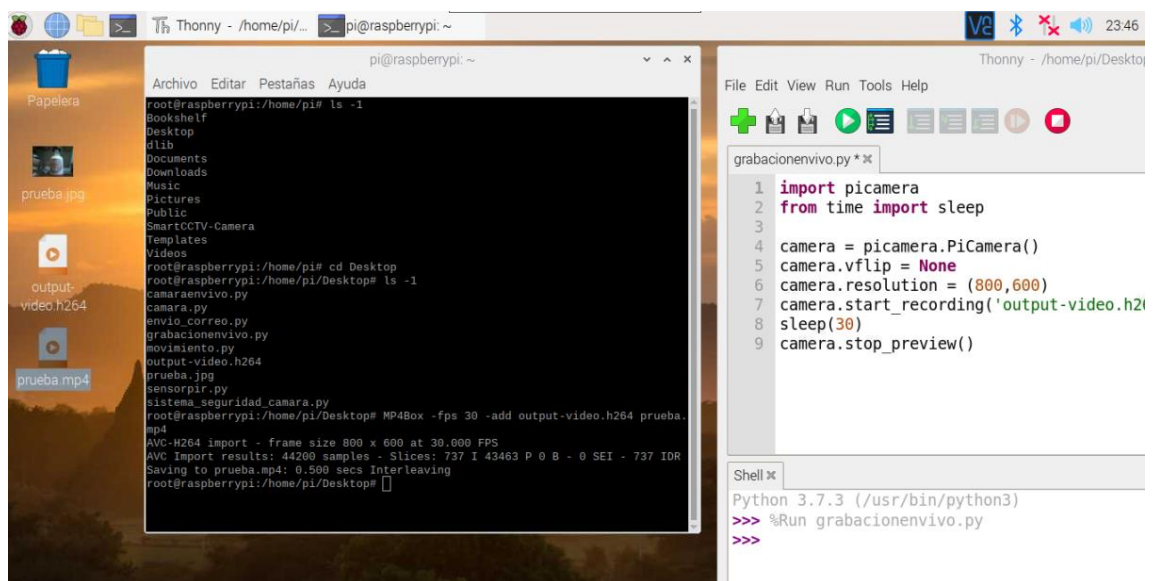
```

Python 3.7.3 (/usr/bin/python3)
>>> %Run grabacionenvivo.py
>>>

```

Figura 44. Código Fuente para grabar videos con Raspberry Pi. Elaborado por autores.

Una vez realizado el código, el usuario debe guardarlo con otro nombre el programa y procede a ejecutarlo mediante Thonny Python, el lenguaje de programación de la Raspberry Pi tiene su configuración para que los videos exportados del código salgan en un formato único llamado dominio(xxx.py) el cual debemos transformarlo para poder visualizarlo mediante un reproductor de media player.



```

root@raspberrypi:~/home/pi# ls -l
total 12
drwxr-xr-x 3 root root 4096 Aug 16 12:24 Desktop
drwxr-xr-x 2 root root 4096 Aug 16 12:24 Documents
drwxr-xr-x 2 root root 4096 Aug 16 12:24 Downloads
drwxr-xr-x 2 root root 4096 Aug 16 12:24 Music
drwxr-xr-x 2 root root 4096 Aug 16 12:24 Pictures
drwxr-xr-x 2 root root 4096 Aug 16 12:24 Public
drwxr-xr-x 2 root root 4096 Aug 16 12:24 SmartCCTV-Camera
drwxr-xr-x 2 root root 4096 Aug 16 12:24 Templates
drwxr-xr-x 2 root root 4096 Aug 16 12:24 Videos
root@raspberrypi:~/home/pi# cd Desktop
root@raspberrypi:~/home/pi/Desktop# ls -l
total 12
-rw-r--r-- 1 root root 1024 Aug 16 12:24 camaraenvivo.py
-rw-r--r-- 1 root root 1024 Aug 16 12:24 camara.py
-rw-r--r-- 1 root root 1024 Aug 16 12:24 envio_correo.py
-rw-r--r-- 1 root root 1024 Aug 16 12:24 grabacionenvivo.py
-rw-r--r-- 1 root root 1024 Aug 16 12:24 movimiento.py
-rw-r--r-- 1 root root 1024 Aug 16 12:24 output-video.h264
-rw-r--r-- 1 root root 1024 Aug 16 12:24 prueba.jpg
-rw-r--r-- 1 root root 1024 Aug 16 12:24 sensorpir.py
-rw-r--r-- 1 root root 1024 Aug 16 12:24 sistema_seguridad_camara.py
root@raspberrypi:~/home/pi/Desktop# MP4Box -fps 30 -add output-video.h264 prueba.mp4
AVC-H264 import - frame size 800 x 600 at 30.000 FPS
AVC Import results: 44260 samples - Slices: 737 I 43463 P 0 B - 0 SEI - 737 IDR
Saving to prueba.mp4: 0.900 secs Interleaving
root@raspberrypi:~/home/pi/Desktop#

```

```

1 import picamera
2 from time import sleep
3
4 camera = picamera.PiCamera()
5 camera.vflip = None
6 camera.resolution = (800,600)
7 camera.start_recording('output-video.h264')
8 sleep(30)
9 camera.stop_preview()

```

```

Python 3.7.3 (/usr/bin/python3)
>>> %Run grabacionenvivo.py
>>>

```


Figura 45. Video de 30 segundos captada con cámara HD y microcontrolador Raspberry Pi. Elaborado por autores.

Una vez terminado el procedimiento básico para la utilización de la cámara, se procederá con el siguiente nivel de este proyecto: Introducción en el formato de IOT de este sistema de seguridad con cámaras de vigilancia a través del control generado a través de una app móvil vinculada al sistema, para lo cual se necesitará la dirección IP de la placa Raspberry y que se encuentra a la misma red que la LAN del usuario, adicional, se necesitará conocer el puerto de conexión (puerto 22 en el sistema del usuario de estudio), el cual puede ser observado en la Figura 46.

En nuestro caso:

Dirección Ip: 192.168.39.167
Nombre del dispositivo: pi (nombre por defecto)
Password: raspberrypi
Puerto de conexión: 22.

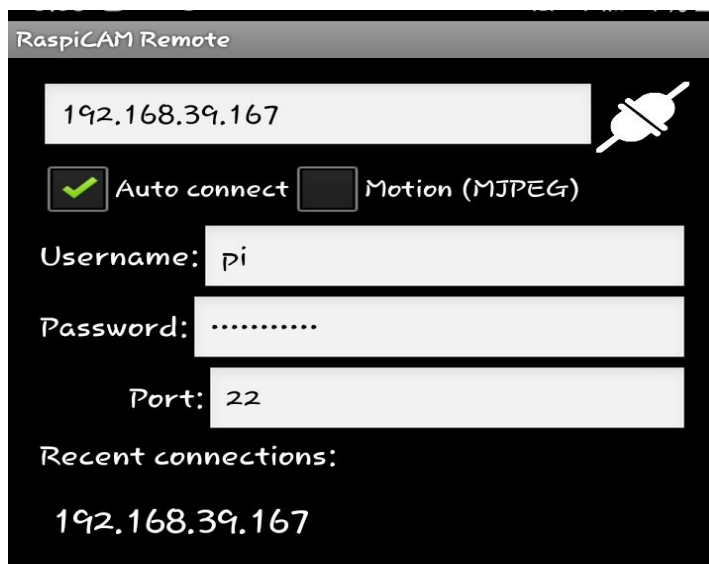


Figura 46. Conexión de aplicación RaspiCamRemote con cámara OV5647. Elaborado por autores.

3.4.3.3. Conexión de Raspberry Pi 4B con dispositivo sensor de movimiento.

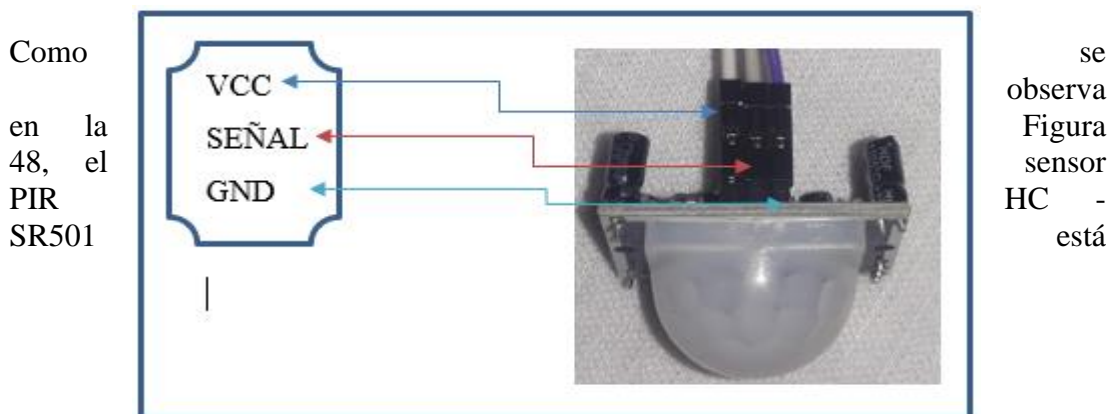
Para realizar una conexión entre un sensor de movimiento y una placa SBC de Raspberry Pi 4B, se debe conectar el sensor mediante sus pines a los pines de GPIO

del microcontrolador, para ello se debe conocer cómo identificar los pines del sensor y a que pines de la placa de Raspberry lo podemos conectar.

En la Figura 47, el usuario identificará que el sensor PIR HC - SR501 cuenta con tres pines que sobresalen de su SBC interna, los cuales son diseñados para su alimentación de voltaje positivo, tierra y salida de señal captada.

Figura 47. Sensor PIR HC-SR501, pines GND – SEÑAL- VCC. Elaborado por autores.

Conociendo lo siguiente se debe definir en cuales de los 40 pines de GPIO del microcontrolador se debe conectar. En la Figura 37, se encontrará la descripción de cada Pin y su funcionalidad.



conectado a los pines GPIO de Raspberry Pi, todo esto sin energizar el módulo aun mientras no se haya configurado los parámetros de medición del sensor, el sensor PIR HC – SR501 tiene dos potenciómetros dentro de su arquitectura de placa SBC, los cuales ajustan los parámetros de sensibilidad y retardo, que se pueden configurar de acuerdo con la necesidad del proyecto, estos tienen un rango entre 5 y 300 segundos (retardo), y 3 a 7 metros (sensibilidad).

A continuación, se detalla el proceso que se debe seguir para la modificación de los parámetros del sensor PIR:

- Para aumentar el tiempo de respuesta debe girar el potenciómetro izquierdo en sentido horario.
- Para aumentar la sensibilidad debe girar el potenciómetro derecho en sentido antihorario.

Si bien este sensor se lo llama comercialmente como sensor de movimiento, técnicamente es un sensor que capta a temperatura ambiente radiación infrarroja, aprovecha de manera eficiente la emisión de radiación de todo cuerpo, mientras más temperatura emane un cuerpo, mayor nivel de radiación presentará, así el sensor puede detectar personas y animales grande.

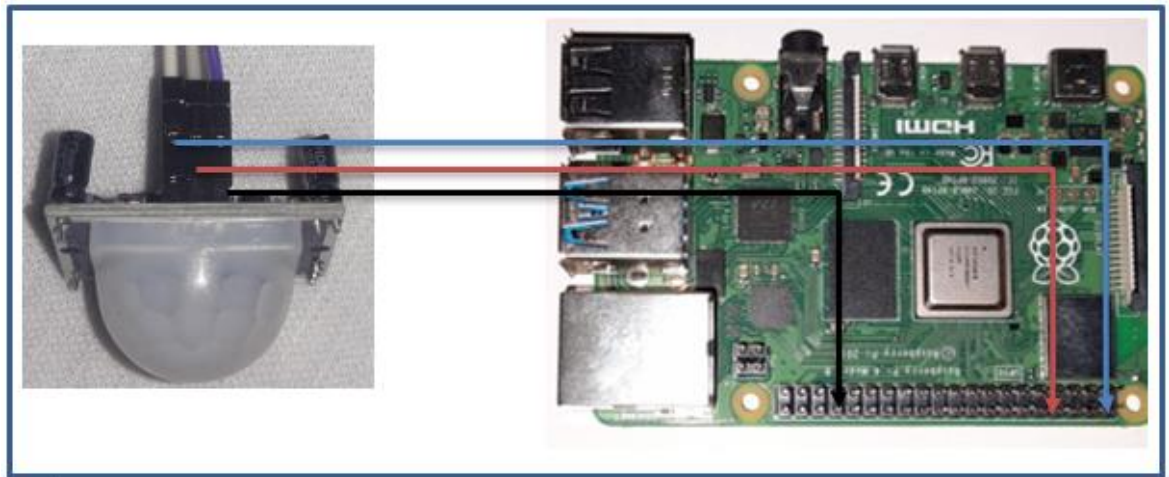


Figura 48. Conexión física de sensor PIR HC – SR501 y Raspberry Pi. Elaborado por autores.

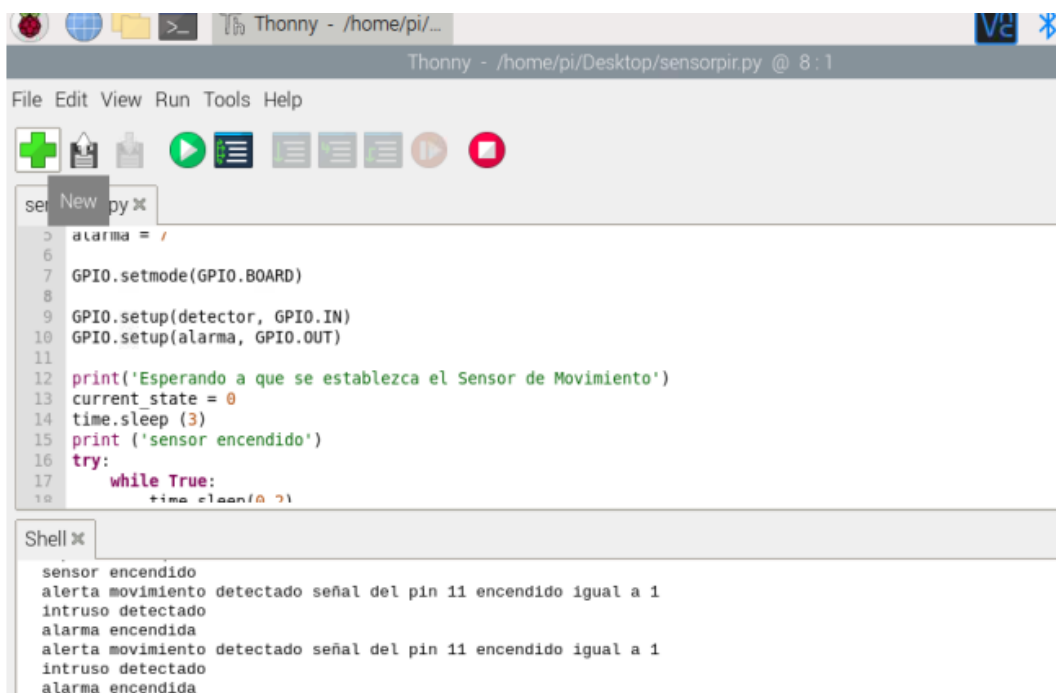
Bien, esto es en cuanto a conexión física, para la ejecución de un sensor de movimiento mediante comandos dentro del ambiente virtual o IDE el usuario debe realizarla creación de un código inicial para activar al sensor de movimiento PIR HC - SR501, el cual se genera a través de la elaboración de un archivo formato (xxx.py) desde el programa Thonny Python. En la Figura 49, se describe el código para el uso del sensor:

 The interface includes a menu bar (File, Edit, View, Run, Tools, Help) and a toolbar with icons for file operations and execution. The system tray at the bottom shows network, Bluetooth, and audio icons.

Figura 49. Sensor PIR HC-SR501 código de programación. Elaborado por autores.

Luego de esto el usuario deberá guardar previamente a iniciar, luego procede a accionar el botón Run.

En la Figura 50, cuando el sensor capta una señal de radiación en su rango hábil, envía un uno lógico al microcontrolador el cual inmediatamente lo interpreta y acciona la alarma por pantalla y activa el buzzer para alarma física – auditiva.



```

Thonny - /home/pi/...
Thonny - /home/pi/Desktop/sensorpir.py @ 8.1
File Edit View Run Tools Help
New py x
5 alarma = /
6
7 GPIO.setmode(GPIO.BOARD)
8
9 GPIO.setup(detector, GPIO.IN)
10 GPIO.setup(alarma, GPIO.OUT)
11
12 print('Esperando a que se establezca el Sensor de Movimiento')
13 current_state = 0
14 time.sleep(3)
15 print('sensor encendido')
16 try:
17     while True:
18         time.sleep(0.1)
Shell x
sensor encendido
alerta movimiento detectado señal del pin 11 encendido igual a 1
intruso detectado
alarma encendida
alerta movimiento detectado señal del pin 11 encendido igual a 1
intruso detectado
alarma encendida

```

Figura 50. Resultado de detección de movimiento en Raspberry Pi. Elaborado por autores.

3.4.4. Conexión de Raspberry Pi 4B con dispositivo sensor de temperatura y humedad.

El dispositivo Sensor de temperatura y humedad es de un tamaño tan pequeño y de una precisión tan confiable que demuestra que es el indicado para un sistema de seguridad electrónica imperceptible para el intruso en un área de acceso restringido. El sensor DHT tiene dos modelos de fabricación DHT11 Y DHT22 se puede apreciar en la Figura 51.

Para realizar una conexión entre un sensor de temperatura y humedad DHT11/DHT22 con un microcontrolador Raspberry Pi 4B, se debe conectar sus tres pines expuestos a los pines de GPIO de la placa de Raspberry Pi, para ello se debe saber cómo identificar los pines del sensor y a que pines de la placa de Raspberry lo podemos conectar.

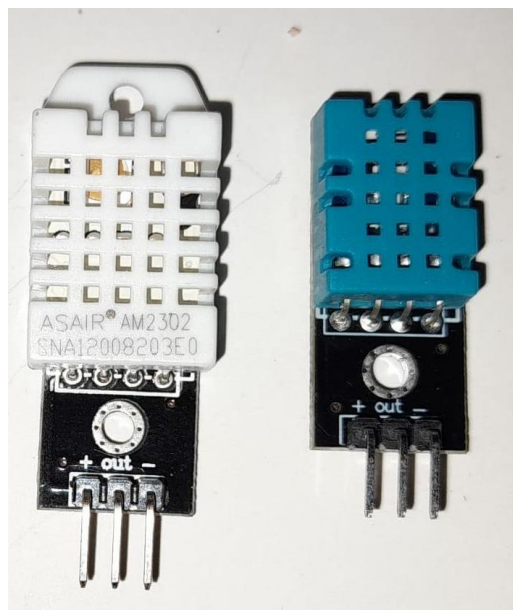


Figura
y humedad
DHT11(azul).

51.Sensor de temperatura
DHT22(blanco) –
Elaborado por autores.

En la Figura 51, se puede observar que el sensor DHT11/DHT22 cuenta con tres pines que sobresalen de su SBC interna, los cuales son diseñados para su alimentación de voltaje +Vcc, tierra y salida de señal captada.

Este sensor digital lleva un microcontrolador para realizar el tratamiento de la señal y este compuesto de un sensor capacitivo para medir la humedad y de un termistor.

Para poder programar este tipo de sensores con el microcontrolador Raspberry Pi existen muchas librerías que se pueden utilizar en el lenguaje de programación de Python, la más utilizada, y en este proyecto caso particular se utilizará la librería Adafruit_DHT.

La presente librería la podemos cargar en el sistema operativo ingresando en el ambiente de escritorio o Raspberry Pi Os en el ícono de terminal programado por teclado la siguiente línea de código:

```
Sudo pip3 install Adafruit_DHT
```

Esta línea de código nos permite utilizar a nuestro gusto todos los sensores DHT11/22 que deseemos programar en nuestro sistema de seguridad.

Al ser un sensor de costo accesible al mercado y por tener una arquitectura de fabricación diminuta permite mantener el sigilo en nuestro sistema de seguridad, ya que al ser instalado no ocupa mucho espacio y su composición le permite mantener una excelente calidad de captación de señal en un ambiente laboral exigente.

En la Figura 52 se observa la conexión entre el sensor DHT11 y el microcontrolador Raspberry Pi, se realiza la conexión de voltaje en el Pin 1 del sensor DHT al pin 1 del

raspberry para la alimentación VCC, luego se conecta pin 2 con el pin GPIO 4 señal, por último, se conecta pin 3 al pin GROUND.

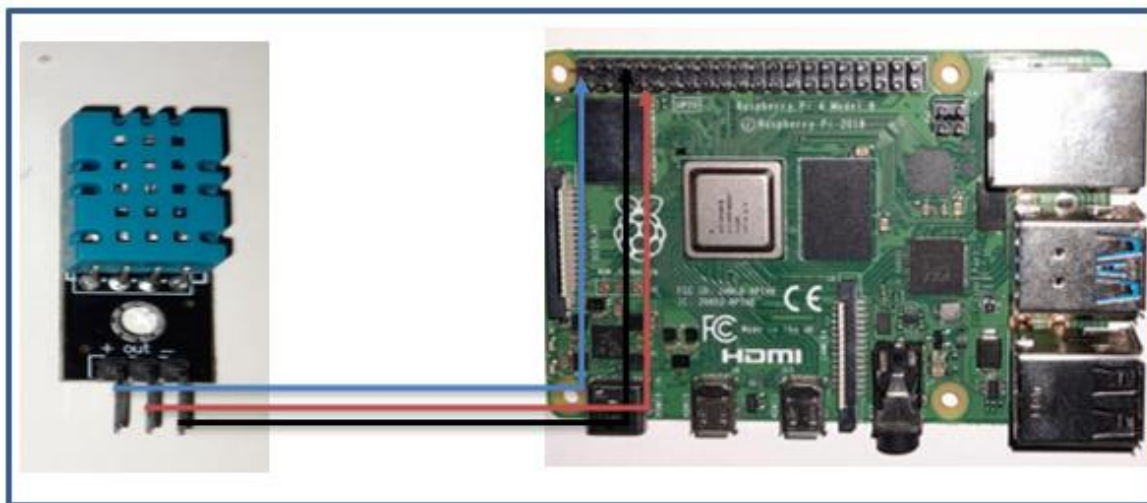


Figura 52. Conexión física del sensor de temperatura y humedad DHT11 y Raspberry Pi. Elaborado por autores.

En la Figura 52, se puede identificar los pines utilizados del sensor de temperatura y humedad DHT11/DHT22 con los pines del microcontrolador Raspberry Pi.

Como se observa, la conexión es mediante dispositivos alámbricos ideales para que el proyecto sea adaptablemente didáctico.

Una vez conectado el sensor de temperatura al microcontrolador el usuario puede crear un código base para la utilización del sistema de seguridad de temperatura y humedad. En la plataforma virtual se debe crear un archivo formato (xxx.py) desde nuestro programa Thonny Python. En la

Figura 53, se describe el código para la utilización del sensor:

*Figura 53. Sensor de temperatura y humedad DHT11 código de programación.
Elaborado por autores.*

Nuevamente el usuario debe guardar previo a iniciar y accionar el botón Run.

Según lo

```

temperatuadht11.py *
1
2 import time
3 import board
4 import adafruit_dht
5
6 dhtDevice = adafruit_dht.DHT11(board.D4)
7
8
9
10 while True:
11     try:
12         temperature_c = dhtDevice.temperature
13         temperature_f = temperature_c * (9 / 5) + 32
14         humidity = dhtDevice.humidity
15         print(
16             "Temp: {:.1f} F / {:.1f} C   Humidity: {}% ".format(
17                 temperature_f, temperature_c, humidity
18             )
19         )
20     )
21
22 except RuntimeError as error:
23     # Errors happen fairly often, DHT's are hard to read, just keep going
24     print(error.args[0])
25     time.sleep(2.0)
26     continue
27 except Exception as error:
28     dhtDevice.exit()
29     raise error
30
31 time.sleep(2.0)
32
33 try:
34     temperature_c = dhtDevice.temperature
35     temperature_f = temperature_c * (9 / 5) + 32
36     humidity = dhtDevice.humidity
37     print(
38         "Temp: {:.1f} F / {:.1f} C   Humidity: {}% ".format(
39             temperature_f, temperature_c, humidity
40         )
41     )
42
43 except RuntimeError as error:
44     # Errors happen fairly often, DHT's are hard to read, just keep going
45     print(error.args[0])
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

Shell
Python 3.7.3 (/usr/bin/python3)
>>> %Run temperatuadht11.py
Temp: 77.0 F / 25.0 C   Humidity: 75%
Temp: 75.2 F / 24.0 C   Humidity: 70%
Checksum did not validate. Try again.
Temp: 75.2 F / 24.0 C   Humidity: 70%
Temp: 75.2 F / 24.0 C   Humidity: 70%

```

programado, el sensor de temperatura y humedad brindará detalles la muestra tomada en el medio ambiente expuesto y nos mostrará por pantalla los datos recolectados como se puede apreciar en la Figura 54, en este caso nos mostrará la humedad y temperatura de Guayaquil mes de Julio 2021.

Figura 54.Resultado de temperatura y humedad en Raspberry Pi. Elaborado por autores.

3.4.5. Raspberry Pi 4B como servidor de correos electrónicos.

Para el presente trabajo de estudio de seguridad electrónica es muy importante tener un medio de cómo comunicar las advertencias cuando los sensores detecten un parámetro fuera de lo normal o para informar al programador/responsable del área alguna novedad posible, en el caso particular de este proyecto, dentro del Data Center.

El presente trabajo de estudio incluye dentro de sus sistemas de alarmas la seguridad de visión, movimiento, temperatura y control de acceso, todas cumplen su función en perfecto control, pero se limita sólo a enviar la señal de alerta al controlador, para dar una solución IOT y poder tener advertencias de estas novedades detectadas, es importante que el microcontrolador notifique al usuario lo sucedido mediante un medio remoto que no necesite tener presencia humana las 24 horas del día.

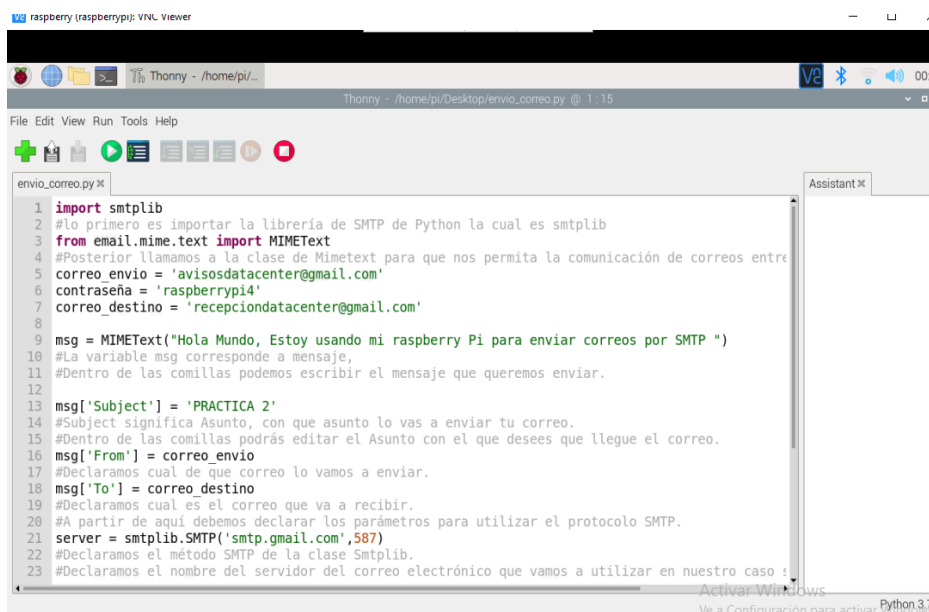
Para el envío de correo electrónico normalmente en las PC, teléfonos celulares y demás dispositivos Smart, lo ejecutan mediante un protocolo llamado SMTP (Protocolo de transferencia simple de correo), sin embargo, en los microcontroladores mediante el lenguaje de programación Python, el usuario puede utilizar una librería desarrollada en código abierto llamado SMTPlib (SMTP protocolclient), el cual define un objeto cliente SMTP para poder enviar correo electrónico a cualquier dispositivo.

A través del protocolo SMTP se envían los correos electrónicos mediante el formato MIME (“extensiones multipropósito de correo de internet”) el cual permite el envío si y solo si cumple con la estructura y formato regido por él.

Para el envío de correos electrónicos mediante SMTPlib es necesario ingresar un correo de origen, su contraseña y uno o varios correos de destino, además de conocer el puerto de comunicación de los servidores de correos electrónicos para poder completar el envío.

Para poder probar el funcionamiento del Raspberry Pi como servidor SMTPlib se debe ingresar al IDE de Raspberry Pi OS y crear un archivo formato (xxx.py) desde nuestro programa Thonny Python.

En la Figura 55 se describe el código para el envío de correo electrónico:



```

1 import smtplib
2 #Lo primero es importar la librería de SMTP de Python la cual es smtplib
3 from email.mime.text import MIMEText
4 #Posterior llamamos a la clase de MimeText para que nos permita la comunicación de correos entre
5 correo_envio = 'avisosdatacenter@gmail.com'
6 contraseña = 'raspberrypi4'
7 correo_destino = 'repciondatacenter@gmail.com'
8
9 msg = MIMEText("Hola Mundo, Estoy usando mi raspberry Pi para enviar correos por SMTP ")
10 #La variable msg corresponde a mensaje,
11 #Dentro de las comillas podremos escribir el mensaje que queremos enviar.
12
13 msg['Subject'] = 'PRACTICA 2'
14 #Subject significa Asunto, con que asunto lo vas a enviar tu correo.
15 #Dentro de las comillas podrás editar el Asunto con el que deseas que llegue el correo.
16 msg['From'] = correo_envio
17 #Declaramos cual de que correo lo vamos a enviar.
18 msg['To'] = correo_destino
19 #Declaramos cual es el correo que va a recibir.
20 #A partir de aquí debemos declarar los parámetros para utilizar el protocolo SMTP.
21 server = smtplib.SMTP('smtp.gmail.com',587)
22 #Declaramos el método SMTP de la clase Smtplib.
23 #Declaramos el nombre del servidor del correo electrónico que vamos a utilizar en nuestro caso :

```

Figura 55.Raspberry Pi como servidor de correo. Elaborado por autores.

3.4.6. Aplicación de sistema de seguridad electrónica con microcontrolador Raspberry Pi en implementación IOT.

Una vez explicado cómo funciona los dispositivos y sensores se demuestra la funcionalidad del dispositivo microcontrolador para realizar sistemas de seguridad que nos permita controlar y supervisar los niveles de temperatura y humedad, niveles de movimiento infrarrojo, y visualización de imagen por video.

Para su configuración cada uno de estos sistemas de alarma genera un código fuente independiente y ejecutable según las conexiones físicas que existan en la comunicación entre el microcontrolador y el sensor, se programa mediante el IDE (Entorno de desarrollo integrado) de Raspberry Pi, donde tiene que existir relación en la programación y en el físico para su correcto funcionamiento.

Para complementar este sistema de seguridad el usuario debe añadir la tecnología IOT mediante el envío inalámbrico de alertas por correo electrónico utilizando la librería de Python SMTPlib que tomará las señales que emitan los sensores de alarmas y notificará por correo al cliente, gerente, supervisor de redes, que se está vulnerando la seguridad del Data Center y poder tomar las respectivas acciones.

En la Figura 56, se puede apreciar como es el sistema de seguridad de cada área específica en una protoboard de prácticas.

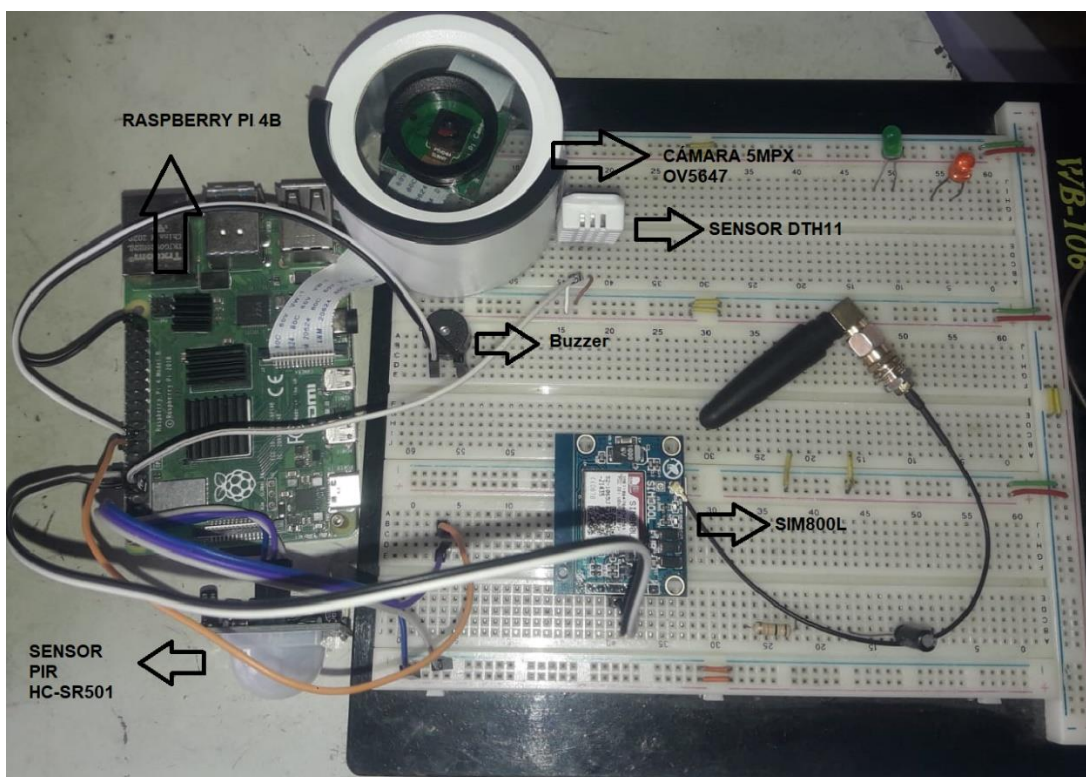


Figura 56. Ejecución de sistema de seguridad electrónica mediante microcontrolador Raspberry Pi con alarmas mediante buzzer, correo electrónico y módulo Sim800L en una protoboard. Elaborado por autores.

3.5. Implementación

La implementación de un módulo didáctico orientado en seguridad electrónica en data center basado en raspberry Pi y Arduino para prácticas en el laboratorio de telecomunicaciones, se llevará a cabo de forma física mediante la fabricación de un módulo que simula una pizarra con facilidad de movilización debido a su estructura con ruedas desplazables. Este modelo es una de las formas ideales que tendrá el estudiante podrá utilizar todos los dispositivos electrónicos del proyecto. A continuación, en la Figura 57, se muestra el plano en 2D realizado en software AutoCAD del módulo de prácticas de seguridad electrónica en Data Center.

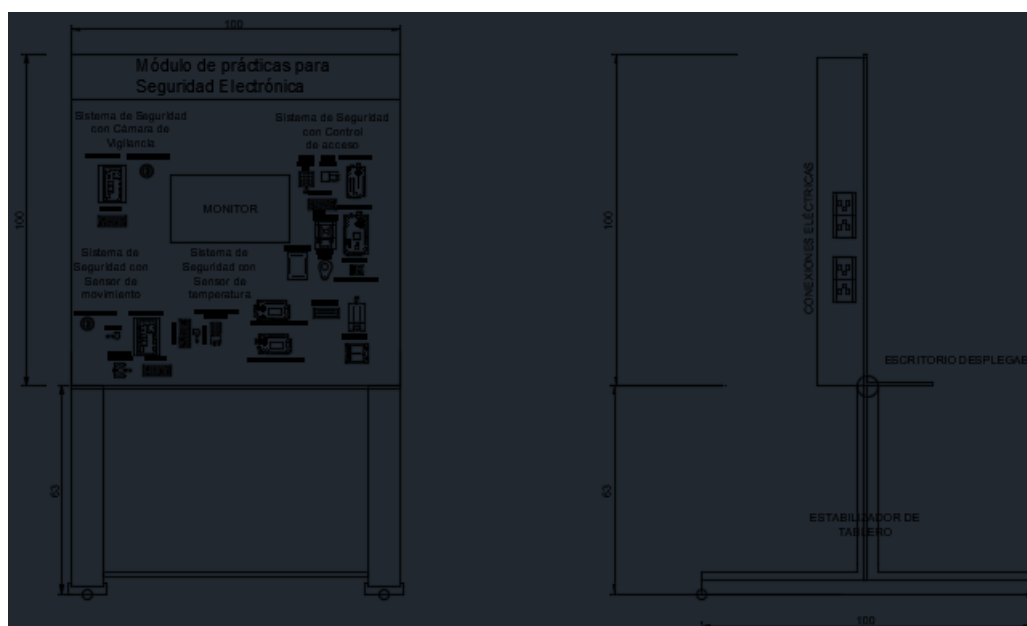


Figura 57. Dibujo AutoCAD módulo de seguridad electrónica. Elaborado por autores.

En la Figura 58, se muestra un renderizado con visualización en 3D realizado en el software Revit para la ejecución de la fabricación del módulo en físico.



Figura 58. Renderizado de módulo de prácticas de seguridad electrónica. Elaborado por autores.

Capítulo IV

Análisis de resultados.

4.1. Módulo de prácticas de seguridad electrónica en Data Center

En la Figura 59 se muestra el tablero didáctico para prácticas en el laboratorio de electrónica y telecomunicaciones el cual tiene su fabricación en estructura de madera, sujeta con pernos y tornillos, con pantalla posterior de MDF y una presentación en acrílica virgen de 2mm.

Para la exposición se añadieron identificadores adhesivos en vinil, y dispositivos electrónicos los cuales harán las funciones de sensores controlador y actuadores en la implementación de un módulo didáctico orientado a seguridad electrónica en data center basado en Raspberry Pi y Arduino para prácticas en el laboratorio de telecomunicaciones.



Figura 59. Módulo didáctico de prácticas de seguridad electrónica. Elaborado por autores.






















Número	Nombre	Gráfico
1	Raspberry PI	
2	Camara de Raspberry	
3	Protoboard	
4	Monitor	
5	Teclado	
6	Chapa Eléctrica	
7	Arduino Uno	
8	Arduino Mega	
9	Lector de Huella Digital	
10	Llave de acceso	
11	Tarjeta RFID	
12	Buzzer	
13	Lector Tarjeta RFID	
14	Sensor PIR	
15	Modulo GSM	
16	Modulo Relé	
17	Modulo wifi	
18	Pantalla LCD	
19	Sensor de Temperatura y Humedad	
20	Teclado inalámbrico	
21	Mouse	

Figura 60.Dispositivos electrónicos para utilizar. Elaborado por autores.

4.2. Listado de prácticas.

PRÁCTICA # 1: Programación de Raspberry Pi mediante el sistema Operativo Raspbian.

PRÁCTICA # 2: Programación de Raspberry Pi como servidor FTP y SMTP.

PRÁCTICA # 3: Configuración, Conexión y cableado de un sistema de circuito cerrado de televisión.

PRÁCTICA # 4: Configuración y Conexión de un sistema de control de acceso utilizando microcontrolador Arduino y tecnología RFID.

PRÁCTICA # 5: Configuración y Conexión de un sensor de humedad y temperatura.

PRÁCTICA # 6: Configuración y conexión de un sensor de movimiento.

PRÁCTICA # 7: Configuración de detector de movimiento en CCTV.

PRÁCTICA # 8: Configuración y Programación para funcionamiento de un módulo GPRS/GSM como alarma de tecnologías RFID, lector de huella, teclado matricial en Arduino.

PRÁCTICA # 9: Configuración de alertas mediante guardado de capturas en Raspberry Pi.

PRÁCTICA # 10: Configuración y visualización de alarmas mediante correo electrónico con tecnología RFID y sensor de movimiento.

4.3. Resultados de las prácticas.

- PRÁCTICA # 1: Programación de Raspberry Pi mediante el sistema Operativo Raspbian (*Véase en **¡Error! No se encuentra el origen de la referencia.***).

Mediante la Implementación de la práctica se logra instalar el sistema operativo Raspberry Pi Os en placa SBC Raspberry Pi 4B. Se comprende los protocolos para poder acceder al ambiente virtual sin necesidad de conexiones físicas a la placa SBC Raspberry Pi 4B.

- PRÁCTICA # 2: Programación de Raspberry Pi como servidor FTP y SMTP (*Véase en Anexo 2*).

En base a lo planteado en esta práctica se comprende el funcionamiento de un servidor FTP, conociendo que se basa en una relación Cliente-Servidor donde existe comunicación mediante la RED de forma unidireccional. Adicional se analiza y ejecuta el protocolo SMTP donde se explica de forma práctica cual es el proceso que atraviesa un correo en la red desde su origen hasta su destino.

- PRÁCTICA # 3: Configuración, Conexión y cableado de un sistema de circuito cerrado de televisión (*Véase en Anexo 2*).

Mediante la ejecución de esta práctica, se consiguió visualizar de una forma didáctica cómo funciona las cámaras de seguridad de Raspberry Pi. Al finalizar la práctica se han obtenido conocimientos en captura de imagen, grabación de video, y acceso remoto al sistema de vigilancia sin necesidad de programación si no mediante una app móvil.

- PRÁCTICA # 4: Configuración y Conexión de sistema de control de acceso (*Véase en Anexo 4*).

La importancia de realizar un sistema de vigilancia para el acceso a un área crítica es muy necesaria en el ámbito de seguridad electrónica. Los sensores RFID dan la

oportunidad de implementar una codificación que ha permitido controlar el acceso de las personas solo con autorización y monitoreo del personal de supervisión.

- PRÁCTICA # 5: Configuración y Conexión de un sensor de humedad y temperatura (*Véase en Anexo 5*).

Gracias a la ejecución de la práctica se muestra cómo las mediciones de las condiciones atmosféricas varían según su geolocalización y ambiente de trabajo, y cómo se puede reunir estos datos informáticos mediante el sensor DHT11/22.

- PRÁCTICA # 6: Configuración y conexión de un sensor de movimiento (*Véase en Anexo 6*).

La ejecución de esta práctica ha permitido comprender el funcionamiento de un sensor de movimiento PIR y una de sus aplicaciones en el área de seguridad como lo es en alarmas físicas. A través de esta práctica permite al usuario comprender cómo comunicar el sensor a un microcontrolador y programarlo de tal forma que pueda activarse sin la necesidad de que un usuario se encuentre permanente supervisando.

- PRÁCTICA # 7: Configuración de detector de movimiento en CCTV (*Véase en Anexo 7*).

La implementación de esta práctica ha permitido la comprensión del funcionamiento de un sensor de movimiento PIR vinculado a una cámara Raspberry Pi, y de estar forma crear un sistema de seguridad más completo, detectando tanto señal de movimiento como imagen de vulnerabilidad. Al existir esta comunicación exitosa entre Sensor PIR y el microcontrolador, facilita al usuario una supervisión remota sin este tener supervisión presencial, gracias al almacenamiento de imagen o video, que es de ayuda en caso de robo.

- PRÁCTICA # 8: Configuración y Programación para funcionamiento de un módulo GPRS/GSM como alarma de tecnologías RFID, lector de huella, teclado matricial en Arduino. (*Véase en Anexo 8*).

El método eficaz para las notificaciones de alarmas es la utilización de señales de radiofrecuencia GSM/GPRS debido a que su conmutación la realiza en menor tiempo que otros medios inalámbricos. La utilización de este módulo Sim800L permite mantener al usuario final informado de novedades en el área que tenga monitoreada.

- PRÁCTICA # 9 Configuración de alertas mediante guardado de capturas en Raspberry Pi. (*Véase en Anexo 9*).

Gracias a la culminación de esta práctica, se logra efectuar el envío de alertas por correo de una detección de intruso mediante un sensor de movimiento PIR vinculado a una cámara Raspberry Pi y con un actuador de envío por Protocolo SMTP de la clase

SMTPLib, y de esta forma crear el sistema de seguridad completo orientado a la tecnología IOT detectando tanto señal de movimiento como imagen de vulnerabilidad.

- PRÁCTICA # 10: Configuración y visualización de alarmas mediante correo electrónico con tecnología RFID y sensor de movimiento (*Véase en Anexo 10*).

Con la finalidad de implementar sensores que permitan alarmar remotamente en caso de anomalías y falsos accesos se puede encontrar al protocolo SMTPLib como la principal herramienta para notificaciones inalámbricas instaladas en el sistema de seguridad creado. Mediante su implementación se asegura que el sistema de seguridad tendrá un respaldo remoto en caso de que no exista personal de vigilancia en los puestos de monitoreo.

Conclusiones

- Para la realización del proyecto de titulación se encontraron conocimientos teóricos sólidos que permitieron el desarrollo y aplicación en las prácticas de manera exitosa.
- El uso de placa SBC Raspberry pi y Arduino como controlador en el desarrollo del sistema de seguridad para una Data Center, permite al estudiante familiarizarse y entender su manejo para la aplicación del conocimiento una vez se encuentre en el mercado laboral.
- La aplicación del lenguaje de programación Python para la ejecución del sistema de seguridad para un Data Center, genera todas las herramientas y es de fácil comprensión para el programador inicial, siendo relevante en el desarrollo del módulo didáctico desarrollado en el proyecto.
- Las prácticas desarrolladas y encontradas en los anexos exponen la funcionalidad del sistema de seguridad para un Data Center encontrado en el módulo didáctico elaborado para el continuo aprendizaje de los estudiantes de la carrera.

Recomendaciones

- En el proyecto se debe considerar la alimentación de los miniordenadores con las fuentes adecuadas debido a que estos dispositivos electrónicos funcionan en óptimas condiciones cuando se cumple el voltaje y la corriente inicial solicitada, garantizar la correcta conexión de fluido eléctrico permitirá trabajar en el tablero didáctica correctamente.
- Es aconsejable utilizar los diagramas de especificaciones de pines tanto del miniordenador Raspberry Pi con sus entradas GPIO como del miniordenador Arduino, debido a que una mala conexión intercambiando los pines no permitirá el cumplimiento a cabalidad de la programación realizada.
- Se debe tener en cuenta los factores externos en la toma de mediciones de datos de los sensores de movimiento, temperatura y humedad, su nivel de sensibilidad es elevado y cualquier cambio brusco en el medio ambiente “clima” puede ocasionar mala lectura de informaciones.
- Es aconsejable utilizar conectores con chaqueta o aislados para las conexiones físicas entre cada uno de los dispositivos electrónicos para evitar daños en las tarjetas integradas o en el peor de los casos un corto entre los pines de conexión.
- Considerar la configuración de seguridad en el protocolo SMTPlib para envío de correos, por formato los servidores de correos bloquean el acceso a dispositivos poco seguros y al implementar un servidor SMTP desde miniordenadores necesitan que exista esta vía habilitada para poder realizar envío y recepción de paquete de datos.
- Se recomienda trabajar de manera remota con el módulo didáctica lo cual facilitará el trabajo en equipo y aumentará el número de personas que podrán utilizar el módulo.

Referencias bibliográficas

- Alexander, C., David, J., & Armando, D. (2012). Piezoelectricidad en un BUZZER. *Tecnura*, 16(0), 111–119. <https://doi.org/10.14483/22487638.6818>
- Alfredo, Moreno, M. (2018). *Arduino. Curso Práctico* (Grupo Edit).
- Artero, O. T. (2013). *ARDUINO. Curso práctico de formación*. (RC Libros).
- Bahit, E., & ALCAZAR LUQUE, E. (2000). Aprenda a Pensar Como un Programador. In *Archivos españoles de urología* (Vol. 8, Issue 2). <http://www.unav.es/SI/manuales/Java/indice.html%0Ahttp://www.iaa.es/python/curso-python-para-principiantes.pdf>
- BeagleBoard. (n.d.). *beagleboard.org*. <http://beagleboard.org>
- Bergkvist, L., Crotty, P., & Hjalmarson, H. (2012). Effects of Green Seals of Approval on Brand Perceptions. *Advances in Advertising Research* (Vol. III), 81–92. https://doi.org/10.1007/978-3-8349-4291-3_7
- Bolaños, D. (2016). *Módulo de relés*. 8, 1–10. <http://www.bolanosdj.com.ar/MOVIL/ARDUINO2/moduloRele.pdf>
- Cajo, R., & Galarza, J. (2010). *Diseño e Implementación de un Sistema Inalámbrico de Alarma Domiciliaria con alerta Vía Celular*. 97. <https://www.dspace.espol.edu.ec/retrieve/129784/D-83509.pdf>
- Christian Cowan., & Gaskins, C. (2020). *Monitoreo de amenazas físicas en centros de datos*. 14. http://www.apc.com/salestools/JMON-5ZLP8M/JMON-5ZLP8M_R3_LS.pdf
- Coba Gibbs, A. J., & Samaniego González, E. (2017). *Implementation of a control system using the input and output RFID reader module with Arduino technology ; Implementación de un sistema de control de entrada y salida empleando el módulo de lectura RFID con la tecnología Arduino*. 2. <http://ridda2.utp.ac.pa/handle/123456789/2183>
- Cobo, A. L. (2014). Protocolo FTP (Protocolo de transferencia de archivos). *Kioskea.Net*, 5. <http://es.ccm.net/contents/263-protocolo-ftp-protocolo-de-transferencia-de-archivos>
- Diseño; Manufactura. (2019). *Diseño y Manufactura*. 1–17. <https://sites.google.com/site/temasdedisenoymanufactura/arduino#TOC-Ventajas-y-desventajas-del-Arduino>
- Edwin, Boza, G. (2012). *Servidor de escritorios remotos*. <http://creativecommons.org/licenses/by-sa/3.0/es/>


- EIA, E. I. A. (2005). *Ansi Tia-942*.
- ELECTRONICS, M. (1995). Temperature sensor. In *DHT11 Humidity & Temperature Sensor*. <https://doi.org/10.1117/3.1002910.ch11>
- ELECTRONILAB. (2011). *Especificaciones Sensor de movimiento PIR HC SR501*. <https://electronilab.co/wp-content/uploads/2013/12/HC-SR501.pdf>
- Enrich, G. G. (2007). El estándar TIA-942. *Ventas de Seguridad*.
- Espín Estévez, D. F., & Maldonado Tinizaray, E. F. (2008). Diseño De Un Sistema De Telemetría Basado En Transmisores De Corriente Y Tecnología Celular. In *Bibdigital.Epn.Edu.Ec*. <http://bibdigital.epn.edu.ec/dspace/handle/15000/842>
- Eugenio, López, A. (2017). *Raspberry Pi Fundamentos y aplicaciones* (RA-MA Edit).
- Federico Nicola, E. (2015). *Redes Celulares*. 01, 26. <https://www.dsi.fceia.unr.edu.ar/downloads/distribuidos/material/monografias/RedesGSM.pdf>
- FERNÁNDEZ, J. R. (2018). *Circuito cerrado de televisión y seguridad electrónica* (E. ParaninfoS.A. (ed.); 2da ed.).
- Ferretrónica. (2021). *Sensores de Temperatura Medida de Temperatura*.
- Garcia Cobo, J. (2017). ¿Qué es una placa SBC? *Hardwarelibre*, 1–2. <https://www.hwlibre.com/que-es-una-placa-sbc/>
- Gardoki, C. (n.d.-a). ¿Sabes cómo utilizar el protocolo FTP? *Hostalia*.
- Gardoki, C. (n.d.-b). *Qué es el protocolo SSH y cómo configurarlo para mejorar la seguridad de acceso a los servidores Linux*. 1, 8. http://pressroom.hostalia.com/wp-content/themes/hostalia_pressroom/images/WP-Hostalia-protocolo-SSH.pdf
- Gualberto Aguilar, Gabriel Sánchez, Karina Toscano, Mariko Nakano, H. P. (2008). Reconocimiento de huellas dactilares usando características locales. *Instituto Politécnico Nacional*, 40, 57–62. http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1870-90442009000200009&lng=en&tlng=en
- Herrero Herranz, J. C., & Sánchez Allende, J. (2015). Una mirada al mundo Arduino. *Revista Tecnológico y Desarrollo*, 13, 28.
- Joyanes Aguilar, L. (2003). *Fundamentos de programación: Libro de problemas. Algoritmos, estructuras de datos y objetos* (2a. ed.).
- Llorens, F. (2015). Dicen por ahí... que la nueva alfabetización pasa por la

- programación. *Revista Digital de Investigación En Docencia Universitaria*, 8(2), 8–11.
https://rua.ua.es/dspace/bitstream/10045/49092/1/2015_Llorens_ReVision.pdf
- López Eusebio, Victor Ernesto Gutiérrez Torres, J. (2017). *Desarrollo de un sistema de medición multipunto de Etileno, Dioxido de Carbono y Oxígeno*. 1(1), 22–28. www.ecorfan.org/republicofperu,
- Marzal Varó, A., Gracia Luengo, I., & García Sevilla, P. (2014). Introducción a la programación con Python 3. In *Introducción a la programación con Python 3*. <https://doi.org/10.6035/sapientia93>
- Michilena, J. R., & Torres, E. G. (2018). *Sistema Electrónico para Control de Acceso de Personas por Reconocimiento de Huella*. 1–6.
- NQA. (2017). *Iso 27001:2013 Guía De Implantación Para La Seguridad De La Información*. 1–30.
- Ochoa, R. (2017). Revista de Ingeniería Eléctrica. *Revista de Ingeniería Eléctrica ECORFAN-Perú*, 1(2), 24–36.
- PACIO, G. (2014). *Data centers hoy* (Alfa Omega).
- Patil, S. (2017). ! Что такое Raspberry Pi - Raspberry Pi энтузиаст. *Electronicwings*, V(Vi), 103–106. <https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access>
- Paz, A. J. C. (2009). Integración de la tecnología GPRS en redes GSM. *Revista Electrónica de Estudios Telemáticos*, 24–41.
- Pi Raspberry. (2019). *Raspberry Pi 4 Computer*. June, 6. www.raspberrypi.org
- Prieto, F. (2016). 3. Gprs. 63–81. <http://bibing.us.es/proyectos/abreproy/11372/fichero/Memoria%252F03+-+GPRS.pdf>
- RealVNC. (2021). *Real VNC*. <https://www.realvnc.com/es/connect/>
- Riabov, V. V. (2011). SMTP (Simple Mail Transfer Protocol). *Handbook of Computer Networks*, 2(December 2007), 388–406. <https://doi.org/10.1002/9781118256114.ch26>
- Rodríguez, N. (2011). Diseños experimentales en educación. *Revista de Pedagogía*, 32(91), 147–158.
- Rossum, G. Van. (2015). *El tutorial de Python*. 105. <http://www.python.org.ar>
- Ruben, G. (2006). VN Software General. *Observatorio Tecnológico*, 1–13.

<http://recursostic.educacion.es/observatorio/web/eu/software/software-general/377-ruben-gonzalez>

- Salcedo, M., & Cendrós, J. (2016). Uso del minicomputador de bajo costo “Raspberry Pi” en estaciones meteorológicas. *Télématique*, 15(1), 62–84.
- Sánchez, Á. G., Sotillo, Á. G., Sarmiento, L. E., & Rodríguez, J. S. (2015). *Servicios de Red e Internet 2/E*. 388. https://www.amazon.es/Servicios-Red-Internet-Texto-garceta/dp/8416228329/ref=sr_1_1?s=books&ie=UTF8&qid=1466631466&sr=1-1&keywords=servicios+networking
- Sandoval-gutiérrez, J., & García-sánchez, J. R. (2021). / *Visto: 1941 Regresar*. 67, 2017–2019.
- Silva, F. (2009). *Manual de Python*.
- Smaldone, J. (2004). *Introducción a Secure Shell*. 0.2, 1–15. http://es.tldp.org/Tutoriales/doc-ssh-intro/introduccion_ssh-0.2.pdf
- Tavil, J. P. (2012). Informe Central Calidad de imagen y transmisión a bajo costo. *Rnds*, 140. http://www.rnds.com.ar/articulos/014/RNDS_056W.pdf
- Trigo, V. (2004). Historia y evolución de los lenguajes de programación. *Autores Científico-Técnicos y Académicos*, 34, 85–95.
- Tuero Suárez, L., Hernández González, A., & Ramírez Beltrán, J. (2016). Sistema para la medición de la humedad relativa usando un DSSP. *Ingeniería Electrónica, Automática y Comunicaciones*, 37(1), 94-107–107. <https://doi.org/10.1234/rielac.v37i1.323>
- Valls, J. uson. (2003). *Introduccion Al Lenguaje*. September 2018.
- Vargas, A. M. (2013). Sistema biométrico de reconocimiento de Huellas dactilares. *ABB Review*, 84.
- VARGAS VERGARA, Z. V. (2013). *Sistema de Control de Acceso y Monitoreo con la Tecnología RFID para el Departamento de Sistemas de la Universidad Politécnica Salesiana Sede Guayaquil*. <http://dspace.ups.edu.ec/handle/123456789/5380>
- Velasco, N. (2005). Introducción al estándar GSM. *Universidad de Sevilla*, 145. <http://bibing.us.es/proyectos/abreproy/11141/fichero/PFC%252F4+Red+GSM.pdf>
- Victor Hugo Fillipo Rugeles, William Olarte Cortes, B. C. Z. (2009). Fundamentos De Diseño Para Un Circuito Cerrado De Television. *Scientia Et Technica*, XV(42), 46–50. <https://doi.org/10.22517/23447214.2583>

- Vinícius V. da Conceicao, Leandro de B. Francisco, B. A. S. J. (2003). Simple Mail Transfer Protocol , uma implementacao simplificada. *Facultad de Comunicación, Universidad Federal de Mato Grosso Do Sul*. <http://www.facom.ufms.br/wp-content/uploads/2015/11/Simple-Mail-Transfer.pdf>
- Weinstein, R. (2005). RFID: A technical overview and its application to the enterprise. *IT Professional*, 7(3), 27–33. <https://doi.org/10.1109/MITP.2005.69>
- Wevar, J. A. S. (2005). *Análisis y estudio de redes GPRS*. https://d1wqtxts1xzle7.cloudfront.net/34982475/Analisis_y_Estudio_de_Redes_GPRS.pdf?1412339359=&response-content-disposition=inline%3B+filename%3DAnalisis_y_Estudio_de_Redes_GPRS_Trabajo.pdf&Expires=1596514445&Signature=YHbkieyL4XuUHhqKfdcweNemKKHP22NC4iL
- Young, S., & Aitel, D. (2003). Simple Mail Transfer Protocol (SMTP). *The Hacker's Handbook*. <https://doi.org/10.1201/9780203490044.ch11>

		REVISION 1/1	Página 1 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Anexos

Anexo 1.Práctica #1realizada para programación de Raspberry Pi mediante el sistema operativo Raspbian

ELECTRÓNICA Y TELECOMUNICACIONES

PRÁCTICA # 1


DOCENTE: ING. LENIN CEVALLOS

TIEMPO ESTIMADO 2 HORAS

NÚMERO DE ESTUDIANTES: 20

TEMA: " PROGRAMACIÓN DE RASPBERRY PI MEDIANTE EL SISTEMA OPERATIVO RASPBIAN"

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 2 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

1) OBJETIVO GENERAL.

Aprender a instalar y programar el sistema operativo de una placa Raspberry Pi.

2) OBJETIVOS ESPECÍFICOS.

- Identificar las herramientas necesarias para los trabajos de programación de una placa de Raspberry Pi.
- Aprender a grabar el sistema operativo de Raspberry Pi en la microSD.
- Realizar una comunicación remota mediante protocolo SSH.
- Realizar una comunicación remota mediante aplicación VNC.
- Realizar test de actualización de paquetes.

3) MARCO TEÓRICO.


RASPBERRY PI 4 MODEL B

Es una placa del tamaño de una tarjeta de crédito considerado como un microcontrolador que soporta varios componentes necesarios en un computador de escritorio Común. Posee puertos de entrada y salida que permite conectar el microcontrolador a medios físicos externos tales como dispositivos de teclado, ratones, pantallas, cámaras, sensores y actuadores.

Especificaciones:

- Procesador central.
- Procesador Gráfico (GPU).
- Módulo de memoria RAM.
- Conector RJ45.
- 2 buses USB 2.0.
- 2 buses USB 3.0.
- Wifi y Bluetooth integrado.
- Interfaz para cámara y pantalla táctil.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

- Salida Analógica de audio/estéreo por Jack de 3,5 mm.
- Salida digital de video MicroHDMI y audio.
- Pines de entrada y salida de propósito General (GPIO).
- 2 puertos de conexión MicroHDMI.
- 1 conector de alimentación MicroUSB tipo C.
- Lector de Tarjeta SD.

SISTEMA OPERATIVO RASPBERRY PI OS

Para la manipulación y programación de las placas de la Raspberry Pi se utiliza sistemas operativos Basados en Linux, el recomendado por fabricantes es Raspberry Pi OS (anteriormente llamado Raspbian basado en Debian). Esta actualización permite optimizar al máximo el hardware de Raspberry Pi.

Es de descarga libre en la página web de la fundación Raspberry Pi y su instalación es mediante un aplicativo Raspberry Pi Imager y almacenado en una tarjeta SD.

PROTOCOLOSSH

El protocolo SSH es un medio seguro de administración remota que permite a los usuarios que la implementan inspeccionar y modificar sus servidores a través de un canal seguro tunelizado en el que toda la información está cifrada.


Mediante el protocolo SSH se puede copiar identificaciones de forma segura, gestionar claves e información prioritaria.

Su puerto TCP asignado es el puerto 22.

VNC

Es una aplicación de procedimientos libre, su arquitectura es cliente – servidor lo cual nos permite visualizar la información del ordenador servidor remotamente a través de un dispositivo cliente.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 4 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Accesorios a usar:



Figura 1: Accesorios Kit Raspberry Pi.

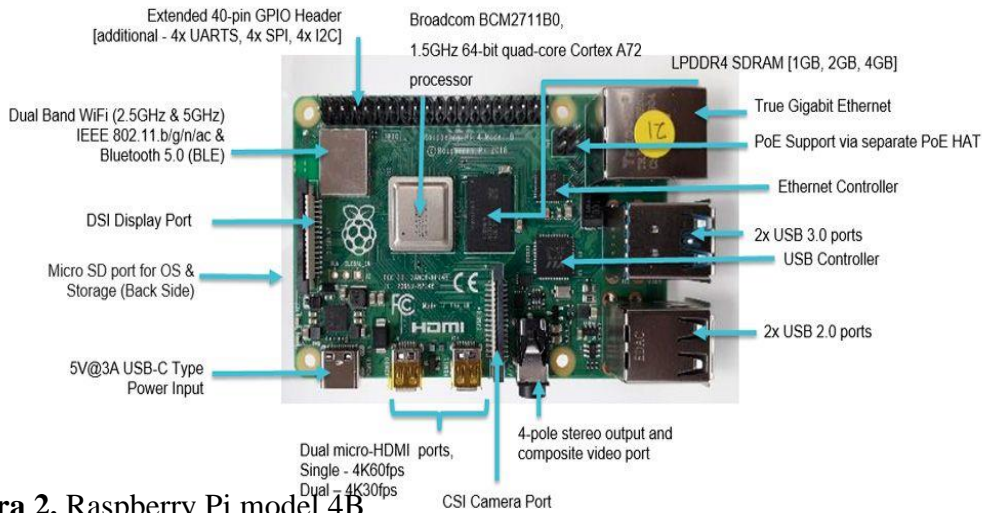



Figura 2. Raspberry Pi model 4B

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 5 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

4) MARCO PROCEDIMENTAL

PROCEDIMIENTO

En el procedimiento para realizar la instalación y programación básica del Raspberry Pi y sus conexiones remotas por SSH y VNC, se deben considerar la alimentación de la placa Raspberry Pi 4 modelo B que es 5V 3A conector tipo USB-C y realizar la programación y conexiones de la manera correcta, para una mejor operatividad.

Paso 1. Descarga de Sistema Operativo Raspberry Pi Imager.

Se inicia la navegación en internet desde el PC y accedemos a la página de la fundación Raspberry Pi.

La página de descarga es la siguiente: <https://www.raspberrypi.org/software/>

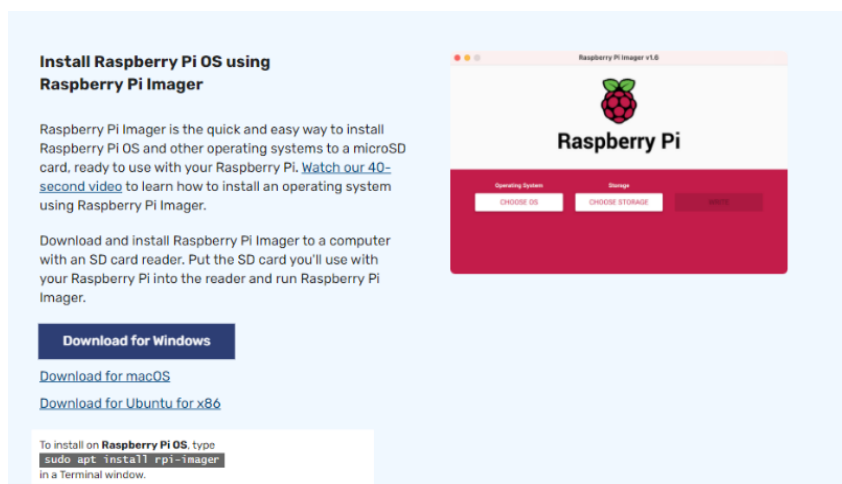



Figura 3. Página descarga SO Raspberry Pi.

Paso 2: Instalación de Raspberry Pi Imager.

Se procede a descargar la aplicación que este de acorde al SO de la PC, Ejecutamos y esperamos que inicie.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 6 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Una vez instalado, se realiza la búsqueda de Raspberry Pi Imager en los documentos. Se desplegará una ventana de la aplicación donde podremos descargar y cargar el sistema operativo que deseemos en nuestra tarjeta SD.




Figura 4. Ventana de aplicación Raspberry Pi Imager.

Paso 3: Conexión de Tarjeta SD y carga de Sistema Operativo.

Antes de seleccionar el sistema operativo que se desea instalar en la placa Raspberry Pi. Se debe conectar la tarjeta de almacenamiento donde se guardará nuestro sistema.

Se debe conectar la MicroSD en un adaptador para poder ingresarlo al puerto de la PC.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 7 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Es importante que la MicroSD tenga una capacidad de almacenamiento igual o superior a 16GB.



Figura 5. Conexión de Adaptador MicroSD a PC.


Paso 4: Cargar Raspberry Pi OS en MicroSD.

Seleccionar el sistema operativo a utilizar, en nuestro caso utilizaremos el SO Raspbian o Raspberry Pi OS, se selecciona la tarjeta de memoria y ejecutamos la sobrescritura del sistema operativo en la microSD.



Figura 6. Carga de sistema Operativo en Micro

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 8 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 5: Iniciación de placa Raspberry Pi.

Una vez realizado la carga del sistema Operativo se debe conectar los dispositivos de entrada y salida a la placa del Raspberry Pi, estos son el monitor o Televisor, Mouse, teclado, conectamos la alimentación de la placa de 5V y 3A e insertamos la tarjeta SD.

Al iniciarse el sistema por primera vez se solicitará ingresar el País, idioma y Zona Horaria.

Se desplegará una pantalla donde pedirá modificar la contraseña de root, se recomienda ubicar la contraseña raspberry. El usuario por defecto del root es pi.

Una vez configurado usuario y contraseña, se solicitará ingresar la conexión WI-FI y contraseña de acceso, esto ocurrirá si no conectamos la placa con el cable de ethernet.

Al final de la configuración nos pedirá si deseamos descargar las actualizaciones existentes.

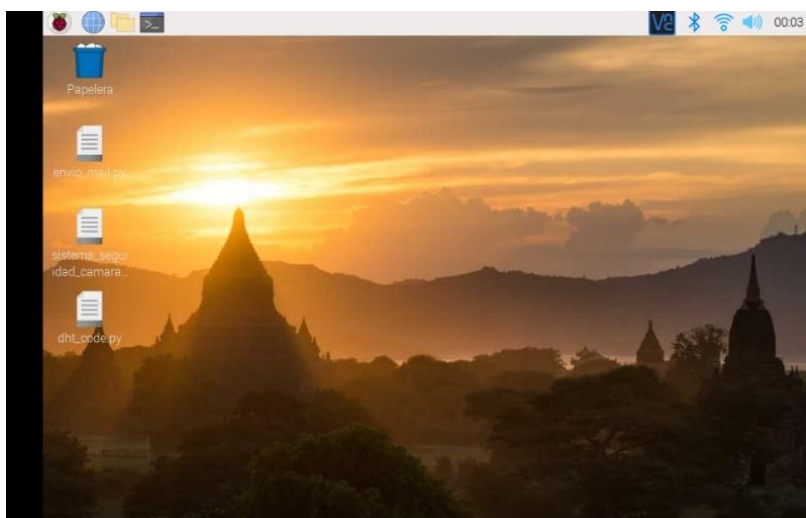



Figura 7. Pantalla de inicio Raspberry Pi OS.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 9 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 6: Instalación de acceso remoto a entorno Raspberry Pi mediante SSH y VNC.

Una vez realizada las configuraciones iniciales de la placa Raspberry Pi, el siguiente paso es habilitar las herramientas del protocolo SSH y VNC que vienen desactivadas por defecto.

Para habilitar se procede a ingresar en el ícono de la frambuesa o Raspberry Pi en la esquina superior izquierda, accedemos a preferencias y configuración de Raspberry Pi. Se desplegará una ventana donde podamos habilitar estas herramientas.

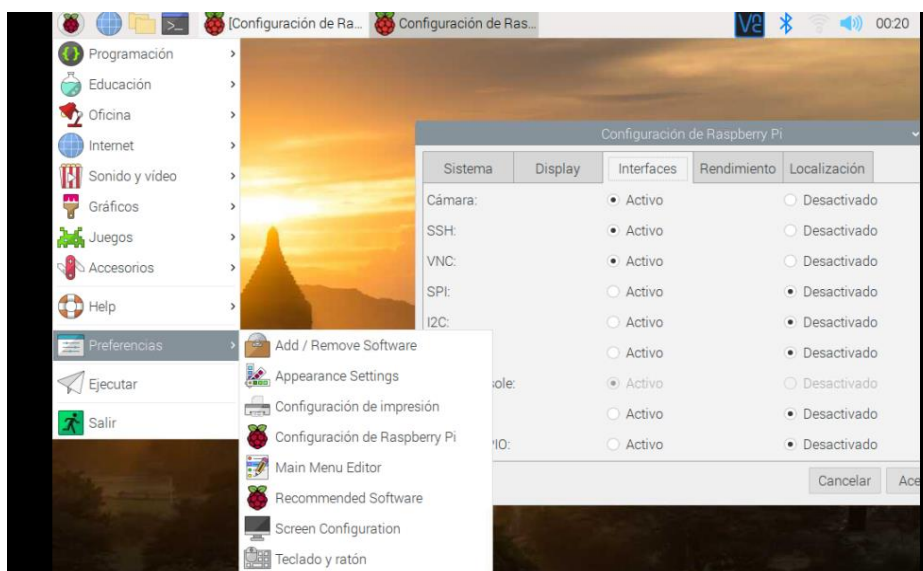



Figura 8. Habilitación de Herramientas SSH Y VNC.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

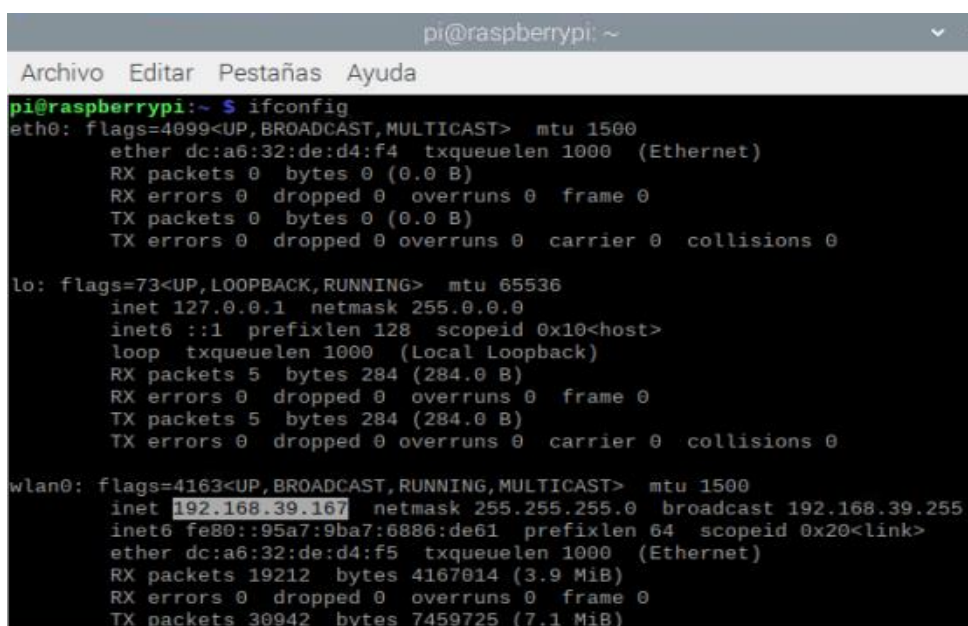
		REVISION 1/1	Página 10 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

A continuación, se despliega una ventana de terminal que nos mostrará nuestra dirección de IP que lo cual permitirá vincular el entorno de Raspberry Pi OS a el acceso remoto.

Para esto se debe ingresar por teclado en una pestaña de terminal que es el cuarto icono en la parte superior izquierda en forma de un cuadrado color negro con el símbolo de mayor que y un subguión.

Seleccionando la botonera se desplegará una ventana donde se debe escribir el siguiente comando:

Ifconfig
Enter



```

pi@raspberrypi:~ $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether dc:a6:32:de:d4:f4 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0


lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 5 bytes 284 (284.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5 bytes 284 (284.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.39.167 netmask 255.255.255.0 broadcast 192.168.39.255
    inet6 fe80::95a7:9ba7:6886:de61 prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:de:d4:f5 txqueuelen 1000 (Ethernet)
    RX packets 19212 bytes 4167014 (3.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 30942 bytes 7459725 (7.1 MiB)

```

Figura 9. Identificación de Dirección Ip.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 11 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

En el caso del raspberry a utilizar la dirección Ip será *192.168.39.167*. y usuario: *pi*; contraseña: *raspberrypi*.

Se debe ingresar una Ip estática, hay que tener en cuenta la dirección IP y si el proveedor está utilizando distribución DHCP, este caso solo se puede presentar si conectan la placa por medio de WI-FI.

Paso 7: Vinculación de acceso remoto con aplicaciones de PC VNC VIEWER

Y MOBAXTERM

Como siguiente paso se debe descargar las aplicaciones que nos permitirá la comunicación de forma remota.

VNC VIEWER: <https://www.realvnc.com/es/connect/download/viewer/>

MOBAXTERM: <https://mobaxterm.mobatek.net/download.html>

Una vez descargada las aplicaciones se ejecutan y habilitan los permisos que se les solicitan.

Ventana de VNC VIEWER


Para la inicialización de la aplicación, se debe acceder a una nueva conexión en la pestaña archivo nueva conexión.

Se desplegará una ventana donde se ingresa la dirección IP del Raspberry Pi y un nombre de cómo se lo quiere guardar, se brinda como sugerencia guardarlo como raspberry.

Una vez realizada la conexión nos pedirá autenticar las credenciales de acceso, se debe ingresar nuestro usuario y contraseña.

<p><i>Usuario: pi</i></p> <p><i>Contraseña: raspberrypi</i></p>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 12 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Ventana de MOBAXTERM

Se inicializa la aplicación y se logea nueva sesión de vinculo en la pestaña de sesión. Se desplegará una ventana donde debemos seleccionar el modo de acceso remoto que deseamos en nuestro caso SSH, ingresar la dirección IP del Raspberry Pi en el recuadro del host.

Una vez realizada la conexión, la aplicación solicitará autenticar las credenciales de acceso, debemos ingresar nuestro usuario y contraseña.

Usuario: <i>pi</i> Contraseña: <i>raspberrypi</i>
--

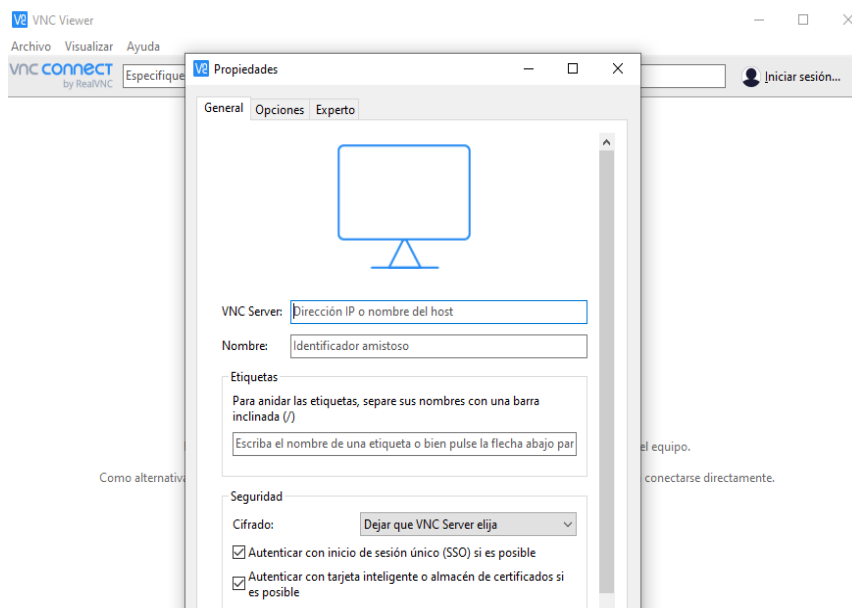



Figura 10. Conexión de acceso remoto VNC.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 13 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

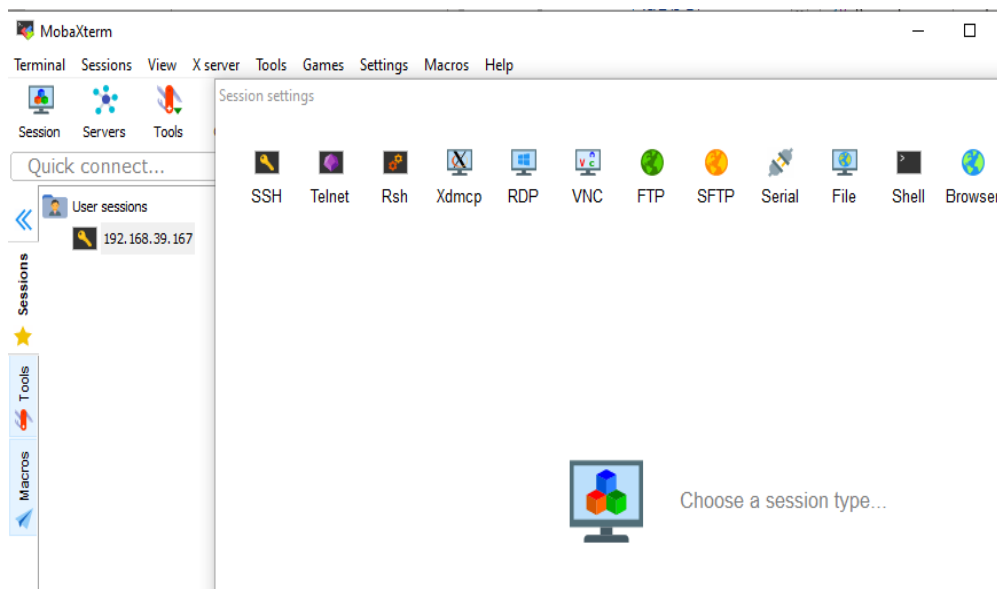


Figura 11. Conexión de acceso remoto SSH.

Paso 8: Iniciación de acceso remoto de placa Raspberry Pi.


Una vez realizada todas las configuraciones de la Raspberry Pi, se debe realizar un reinicio del sistema operativo para que los cambios se guarden y se inicie con las nuevas herramientas instaladas.

Paso siguiente se debe desconectar todos los dispositivos de entrada y salida de la placa debido a que estaremos trabajando bajo la modalidad de acceso remoto, alimentamos la placa con su respectivo cargador de 5V 3A, y accedemos a el ambiente de Raspberry Pi OS mediante las aplicaciones de VNC y Mobaxterm.

Una vez iniciado, se debe inicial una ventana de terminal para la actualización de paquetes.

Una vez dentro de la botonera de comandos Primero, se procede a actualizar la lista de paquetes de su sistema ingresando el siguiente comando:

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 14 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
sudo apt-get update
```

Paso siguiente, actualizar todos sus paquetes instalados a sus últimas versiones con el siguiente comando:

```
sudo apt-get upgrade
```

para finalizar ingresamos el comando *clear* para limpiar la ventana, y se recomienda reiniciar el raspberry con el comando:

```
sudo rebootnow.
```

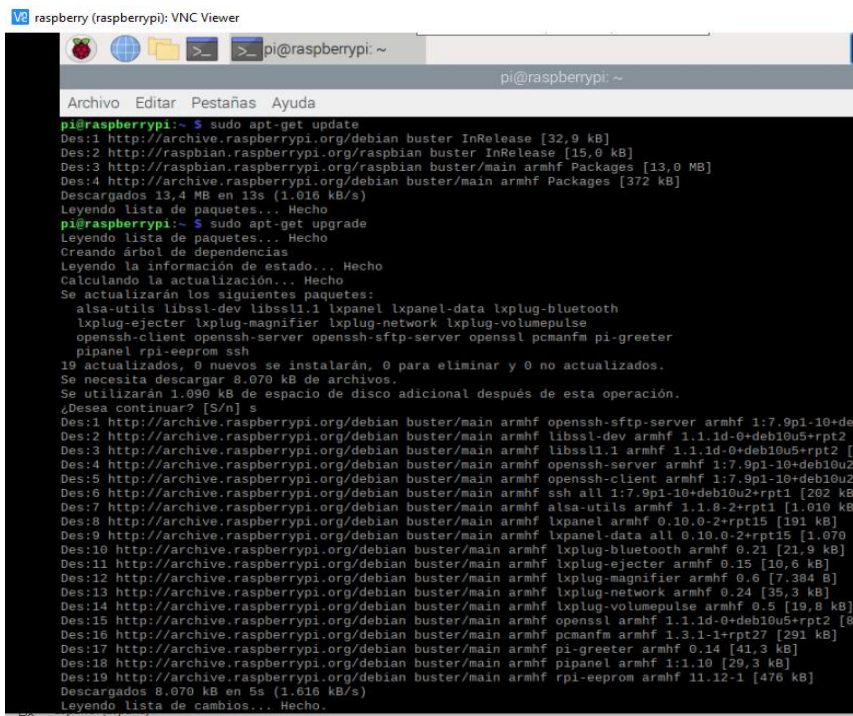



Figura 12. Realización de la prueba de actualización de paquetes.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 15 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

5) RECURSOS UTILIZADOS

- Kit Raspberry Pi.
- Adaptador MicroSD.
- Tarjeta MicroSD.
- Laptop.
- Monitor.
- Teclado.
- Mouse.
- Cable Ethernet.
- Cable HDMI – HDMI.
- Adaptador HDMI a MicroHDMI.

6) RESULTADOS OBTENIDOS

Mediante la Implementación de la práctica se logró instalar el sistema operativo Raspberry Pi Os en placa SBC Raspberry Pi 4B. Se comprendió protocolos para poder acceder al ambiente virtual sin necesidad de conexiones físicas a la placa SBC Raspberry Pi 4B.


7) CONCLUSIONES

Los microcontroladores Raspberry Pi necesita su configuración básica, su diseño de elaboración nos permite manipularlo a nuestro criterio y nos facilita la utilización de todas las entradas y salidas, así como programar su ambiente virtual de una forma ideal.

8) BIBLIOGRAFÍA UTILIZADA

<https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/1>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 1 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Anexo 2. Práctica desarrollada para la programación de Raspberry Pi como servidor FTP y SMTP.

ELECTRÓNICA Y TELECOMUNICACIONES

PRÁCTICA # 2


DOCENTE: ING. LENIN CEVALLOS

TIEMPO ESTIMADO 2 HORAS

NÚMERO DE ESTUDIANTES: 20

TEMA: “PROGRAMACIÓN DE RASPBERRY PI COMO SERVIDOR FTP Y SMTP”.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 2 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

1) OBJETIVO GENERAL.

Aprender a configurar al Raspberry Pi como un servidor FTP y SMTP.

2) OBJETIVOS ESPECÍFICOS.

- Identificar las herramientas necesarias para los trabajos de programación de una placa de Raspberry Pi.
- Realizar la programación para configurar al raspberry Pi como servidor FTP.
- Realizar transferencia de archivos Cliente – Servidor.
- Realizar la programación para configurar al raspberry Pi como servidor SMTP.
- Realizar transferencia de correo electrónico utilizando Raspberry Pi.

3) MARCO TEÓRICO.

RASPBERRY PI 4 MODEL B


Es una placa del tamaño de una tarjeta de crédito considerado como un microcontrolador que soporta varios componentes necesarios en un computador de escritorio Común.

Posee puertos de entrada y salida que permite conectar el microcontrolador a medios físicos externos tales como dispositivos de teclado, ratones, pantallas, cámaras, sensores y actuadores.

Especificaciones:

- Procesador central.
- Procesador Gráfico (GPU).
- Módulo de memoria RAM.
- Conector RJ45.
- 2 buses USB 2.0.
- 2 buses USB 3.0.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

- Wifi y Bluetooth integrado.
- Interfaz para cámara y pantalla táctil.
- Salida Analógica de audio/estéreo por Jack de 3,5 mm.
- Salida digital de video MicroHDMI y audio.
- Pines de entrada y salida de propósito General (GPIO).
- 2 puertos de conexión MicroHDMI.
- 1 conector de alimentación MicroUSB tipo C.
- Lector de Tarjeta SD.

PROTOCOLO FTP.

El protocolo de transferencia de archivos (FTP) es un medio por el cual se puede transferir archivos entre un cliente y un servidor de una red de computadoras.

Permite la comunicación de forma remota de dos computadoras mediante la red, Una de las computadoras servirá como cliente para enviar o solicitar archivos y la otra como servidor que permitirá almacenar información.

Utiliza el puerto 21 como su medio de comunicación con el mundo exterior. Por lo cual si deseas utilizar el protocolo FTP solo debes conectar el cliente en el puerto 21.

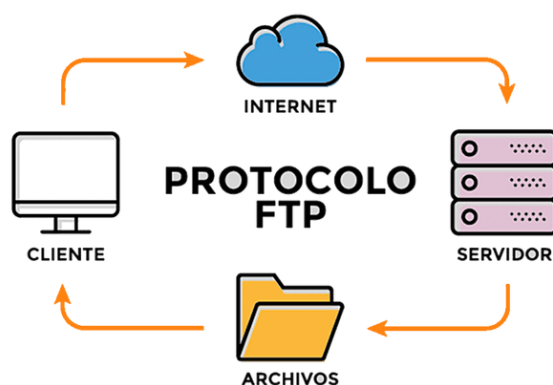



Figura 1. Diagrama de Protocolo FTP

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 4 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

PROCOLO VSFTPD

VSFTPD es un servicio FTP utilizado en ambiente virtual Linux que nos brinda servicio de registro mediante el protocolo FTP, se caracteriza por ser muy sencillo de configurar y a la vez por ser un sistema muy seguro.

FILEZILLA CLIENT

FileZilla es una aplicación FTP libre y de código abierto que puede funcionar como un cliente y un servidor. Es capaz de soportar los protocolos FTP, SFTP y FTP sobre SSL/TLS.

PROCOLO SMTP.

Protocolo SMTP (Simple Mail Transfer Protocol o protocolo para transferencia simple de correos), es un protocolo de red que permite la comunicación de mensajes vía correo electrónico entre dispositivos, computadoras, impresoras, teléfonos móviles entre otros. El servidor SMTP es el encargado del transporte del correo electrónico, como requisito nos solicita confirmar las identidades del remitente y del destinatario, una vez registrados nos permitirá el envío de este. Utiliza los puertos 25 o 587 para la serie de intercambios de información entre el servidor SMTP del cliente de un correo electrónico y el servidor SMTP del correo electrónico destino.

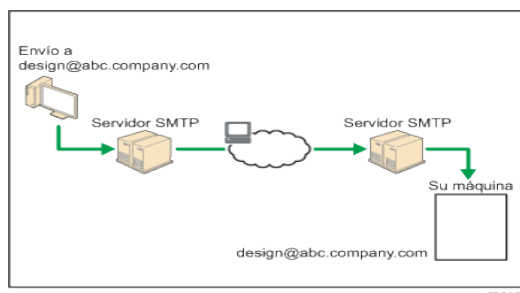



Figura 2. Diagrama de protocolo SMTP

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 5 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

VNC

Es una aplicación de procedimientos libre, su arquitectura es cliente – servidor lo cual nos permite visualizar la información del ordenador servidor remotamente a través de un dispositivo cliente.

SMTPLIB

El lenguaje de programación Python nos proporciona una librería de nombre SMTPlib que define un objeto cliente SMTP para poder utilizar, enviar y recibir correos electrónicos a cualquier dispositivo.

MIMETEXT

Es una clase del protocolo SMTP, Sus siglas (MIME) se traduce como extensión de multipropósito de correo e internet, son una serie de reglas de escritura y estructura del correo electrónico.

Accesorios a usar

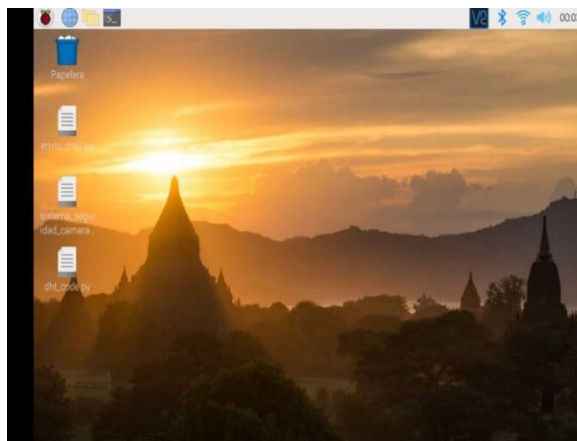
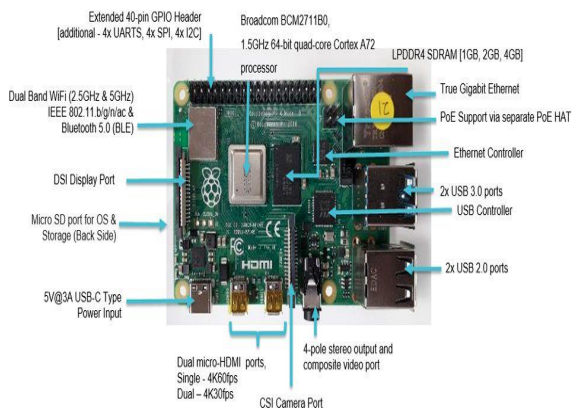



Figura 3. Raspberry Pi model 4B **Figura 4.** Entorno virtual Raspberry Pi OS

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 6 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

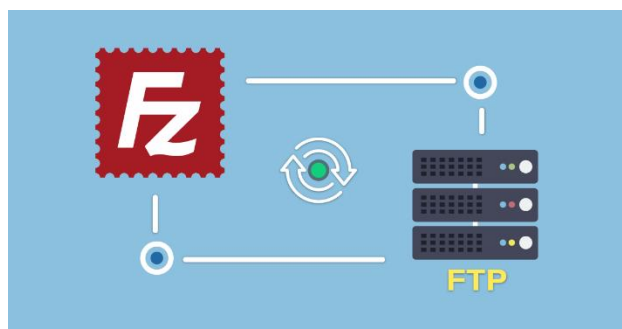


Figura 5. Cliente FTP Filezilla.

4) MARCO PROCEDIMENTAL

PROCEDIMIENTO

El procedimiento para realizar la configuración de la placa Raspberry Pi modelo 4B como servidor FTP y servidor SMTP se realiza mediante el ambiente virtual Raspberry pi OS por medio de comandos en la pestaña Terminal, la programación es manera remota conectándonos a la placa de raspberry Pi mediante VNC, como nota importante la placa debe tener una alimentación de 5V y 3^a para su correcto funcionamiento.

4.1 RASPBERRY PI COMO SERVIDOR FTP


Paso 1. Iniciamos Sesión en Raspberry Pi OS mediante VNC.

Para inicializar la práctica se debe energizar a la placa Raspberry Pi 4B con 5v y 3A, inmediatamente iniciamos sesión mediante nuestro protocolo VNC.

Recordandolas credenciales de acceso:

Dirección Ip del raspberry Pi: <i>172.18.63.250</i> . Usuario: pi Contraseña: raspberrypi

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 7 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

(NOTA IMPORTANTE), si no recuerda la dirección IP del Raspberry conecte la placa a un monitor mediante un cable HDMI y en la pestaña comandos realice la búsqueda mediante el código “ifconfig”.

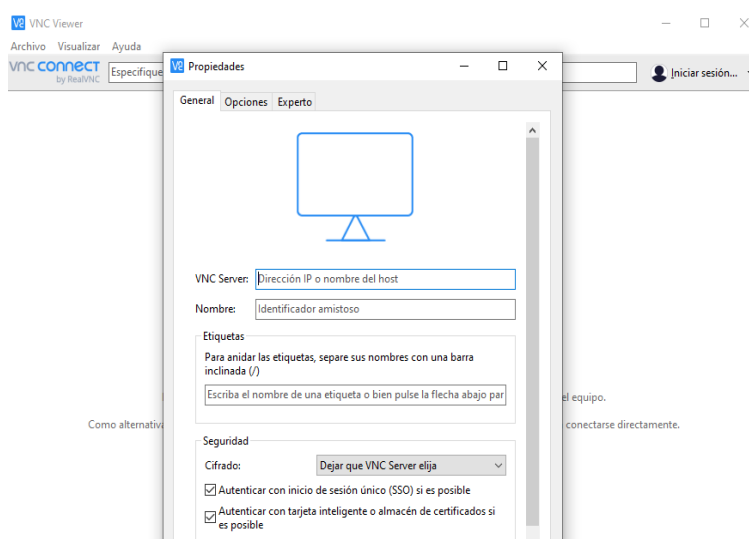


Figura 6. Conexión de acceso remoto VNC.

Paso 2. Actualización de paquetes Raspberry Pi.

Una vez iniciado la conexión con el ambiente virtual desde el protocolo “VNC”, se procede a iniciar la pestaña de Terminal.


Se recomienda actualizar los paquetes del sistema operativo con el siguiente comando:

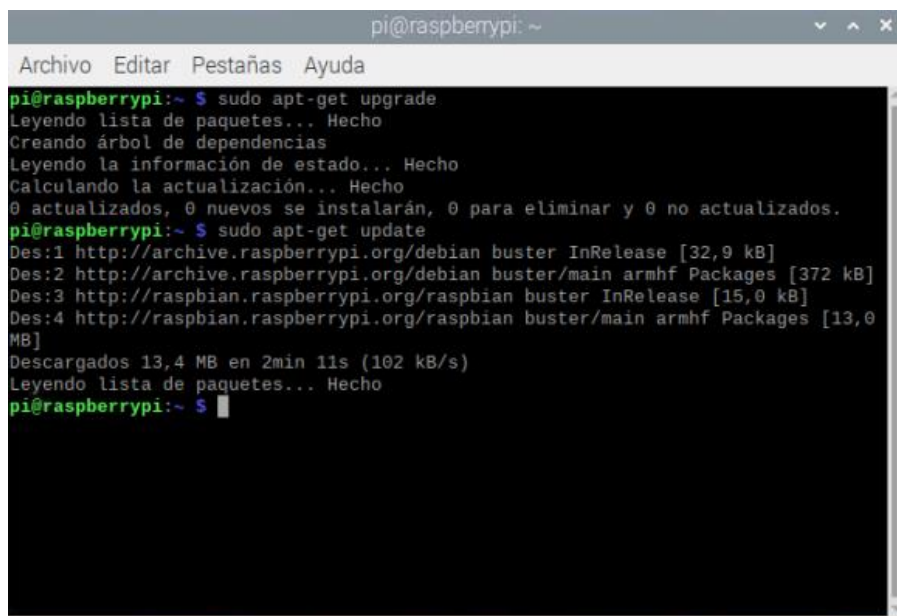
```
sudo apt-get update
```

Paso siguiente, se debe actualizar todos sus paquetes instalados a sus últimas versiones con el siguiente comando:

```
sudo apt-get upgrade
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 8 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



```

pi@raspberrypi:~ $ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
pi@raspberrypi:~ $ sudo apt-get update
Des:1 http://archive.raspberrypi.org/debian buster InRelease [32,9 kB]
Des:2 http://archive.raspberrypi.org/debian buster/main armhf Packages [372 kB]
Des:3 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Des:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0
MB]
Descargados 13,4 MB en 2min 11s (102 kB/s)
Leyendo lista de paquetes... Hecho
pi@raspberrypi:~ $

```


Figura 7. Actualización de paquetes.

Paso 3. Instalación de VSFTPD como servidor FTP.

Una vez terminado la actualización de paquetes, se debe proceder a instalar el Servidor FTP, en este caso se debe instalar uno compatible al lenguaje de programación Linux como lo es “VSFTPD” con el siguiente comando:

```
Sudo apt-get install vsftpd
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 9 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
Descargados 13,4 MB en 2min 11s (102 kB/s)
Leyendo lista de paquetes... Hecho
pi@raspberrypi:~ $ sudo apt-get install vsftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  vsftpd
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 3 no actualizados.
Se necesita descargar 137 kB de archivos.
Se utilizarán 306 kB de espacio de disco adicional después de esta operación.
Des:1 http://mirror.cedia.org.ec/raspbian/raspbian buster/main armhf vsftpd armhf
f 3.0.3-12 [137 kB]
Descargados 137 kB en 5s (28,0 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete vsftpd previamente no seleccionado.
(Leyendo la base de datos ... 163705 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar .../vsftpd_3.0.3-12_armhf.deb ...
Desempaquetando vsftpd (3.0.3-12) ...
Configurando vsftpd (3.0.3-12) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service - /li
b/systemd/system/vsftpd.service.
[vsftpd.conf:1] line references path below legacy directory /var/run/, updating
/var/run/vsftpd/empty -> /run/vsftpd/empty; please update the tmpfiles.d/ drop-in
file accordingly.
Procesando disparadores para man-db (2.8.5-2) ...
Procesando disparadores para systemd (241-7-deb10u6+rp1) ...
pi@raspberrypi:~ $

```

Figura 8. Instalación de protocolo Vsftpd.


Una vez termina la instalación del servidor “FTP” se necesita editar el archivo de configuración con el siguiente comando:

```
Sudo nano /etc/vsftpd.conf
```

Se busca la línea de comando que indique Local Enable y cambiar su estado a YES.

Se busca la línea de comando que indique Write Enable y cambiar su estado a YES.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 10 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

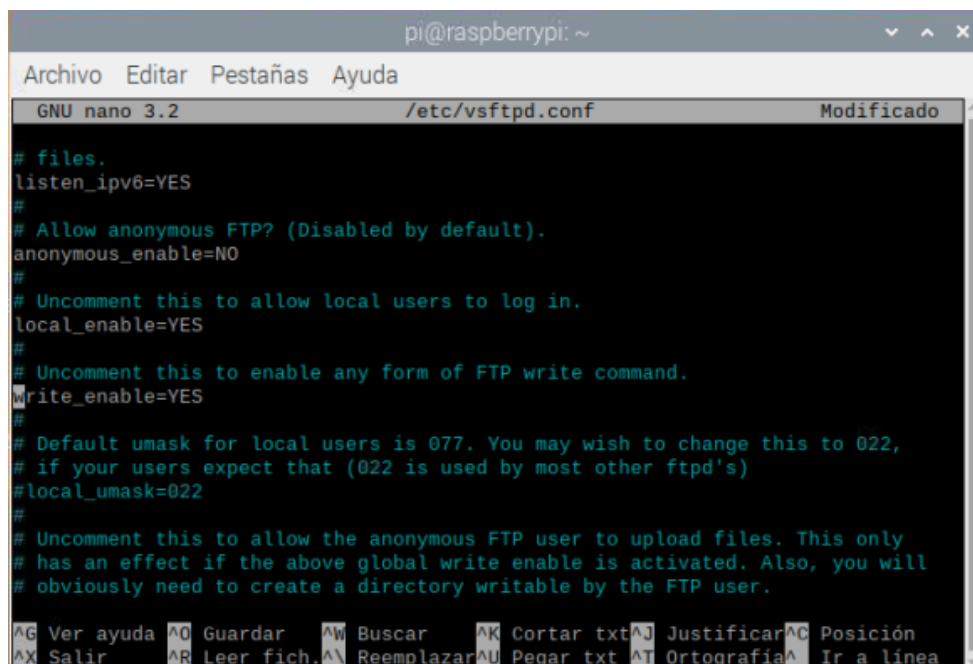



Figura 9. Edición de archivo de configuración de protocolo VSFTPD.

Una vez realizado estos comandos se procede a guardar los cambios con la combinación de las letras Ctrl + O y salimos con Ctrl + X.

Para guardar los cambios en la pestaña de terminal, debemos realizarlo por comando:

```
Sudo service vsftpd restart.
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 11 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

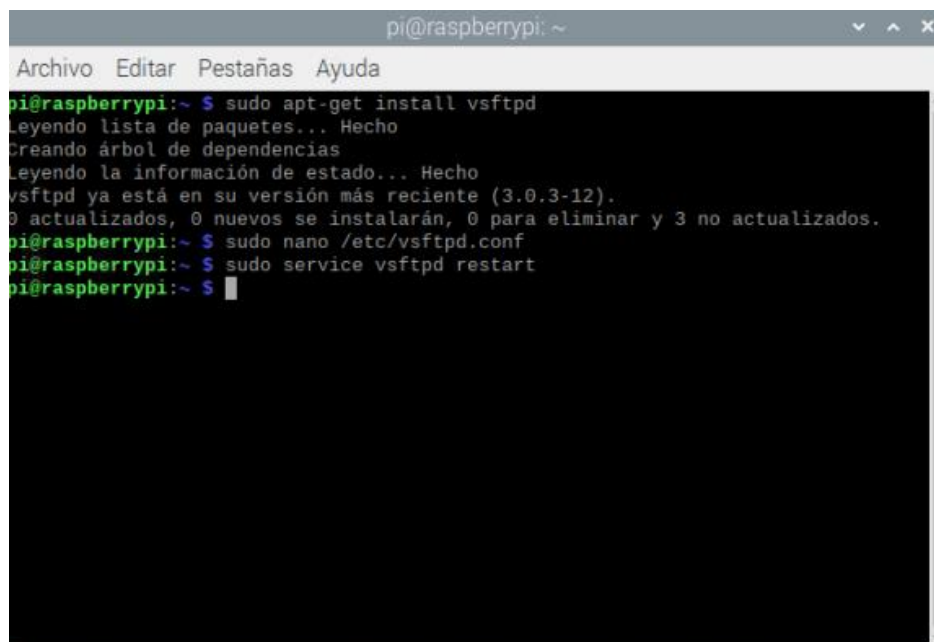



Figura 10. Comando de guardar cambios en servidor ftp vsftpd.

Paso 4. Instalación de Cliente FTP FileZilla en Raspberry Pi como servidor FTP.

Para instalar la aplicación “Filezilla” es nuestro Raspberry Pi, se debe ingresar por comando en la pestaña de Terminal el siguiente comando:

```
Sudo apt-get install filezilla
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 12 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda

pi@raspberrypi:~$ sudo apt-get install filezilla
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  filezilla-common libfilezilla0 libpugixmlv5 libwxbase3.0-0v5
  libwxgtk3.0-0v5
Se instalarán los siguientes paquetes NUEVOS:
  filezilla filezilla-common libfilezilla0 libpugixmlv5 libwxbase3.0-0v5
  libwxgtk3.0-0v5
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 3 no actualizados.
Se necesita descargar 8.256 kB de archivos.
Se utilizarán 32,2 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://mirror.cedia.org.ec/raspbian/raspbian buster/main armhf filezilla-c
common all 3.39.0-2+deb10u1 [2.307 kB]
Des:2 http://mirror.cedia.org.ec/raspbian/raspbian buster/main armhf libfilezill
a0 armhf 0.15.1-1 [57,1 kB]
Des:3 http://mirror.cedia.org.ec/raspbian/raspbian buster/main armhf libpugixml
v5 armhf 1.9-3 [76,3 kB]
Des:4 http://mirror.cedia.org.ec/raspbian/raspbian buster/main armhf libwxbase3.
0-0v5 armhf 3.0.4+dfsg-8 [891 kB]
Des:5 http://mirror.cedia.org.ec/raspbian/raspbian buster/main armhf libwxgtk3.0
-0v5 armhf 3.0.4+dfsg-8 [3.556 kB]

```

Figura 11. Instalación de cliente FTP Filezilla.

Una vez instalado el cliente “FTP Filezilla”, se lo podrá encontrar en la parte superior dando clic sobre el logo de Raspberry en la sección internet.

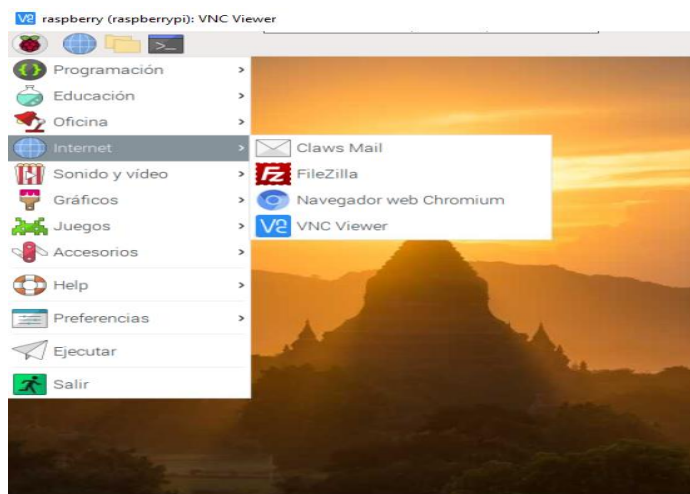



Figura 12. Visualización aplicación de Filezilla en entorno Raspberry Pi OS.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 13 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Si se procede a ejecutar la aplicación Filezilla, esta solicitará las credenciales de acceso al Raspberry Pi. Las cuales son:

Dirección IP:
Usuario: pi
Contraseña: raspberrypi

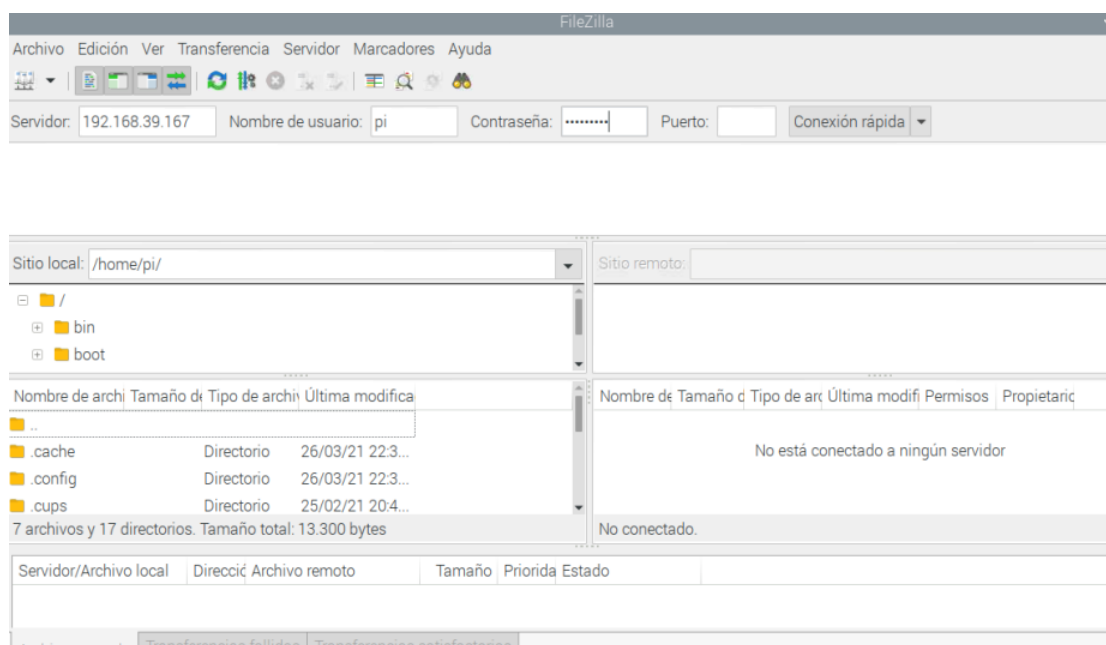



Figura 13. Conexión cliente servidor FTP mediante credenciales.

Una vez permitido el acceso, ya se puede utilizar el servidor FTP para la transferencia de archivos mediante la Red. Copiar y enviar archivos desde el servidor al cliente.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 14 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

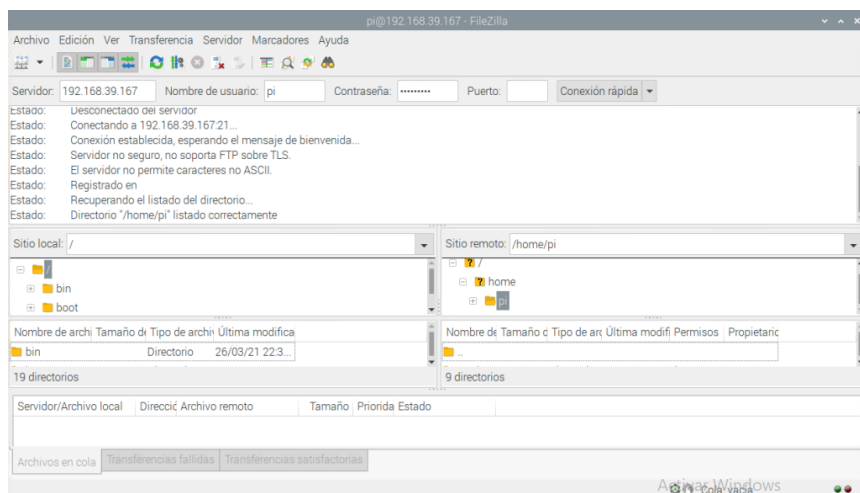


Figura 14. Conexión FTP entre dispositivos.

4.2 RASPBERRY PI COMO SERVIDOR SMTP

Paso 1. Iniciamos Sesión en Raspberry Pi OS mediante VNC.

Para inicializar la práctica se debe energizar a la placa Raspberry Pi 4B con 5v y 3A, inmediatamente iniciamos sesión mediante nuestro protocolo VNC.

Recordemos nuestras credenciales de acceso:


Dirección IP del Raspberry Pi: *192.68.39.168*.

Usuario: pi

Contraseña: raspberrypi

(NOTA IMPORTANTE), si no recuerda la dirección Ip del raspberry conecte la placa a un monitor mediante un cable HDMI y en la pestaña comandos realice la búsqueda mediante el código ifconfig.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 15 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

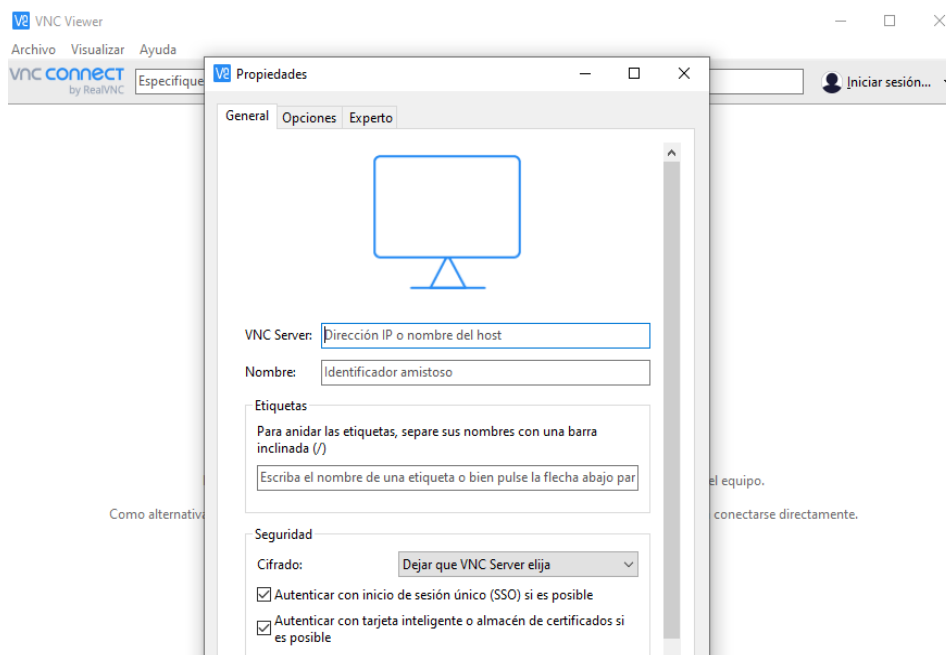


Figura 15. Conexión de acceso remoto VNC.


Paso 2. Actualización de paquetes Raspberry Pi.

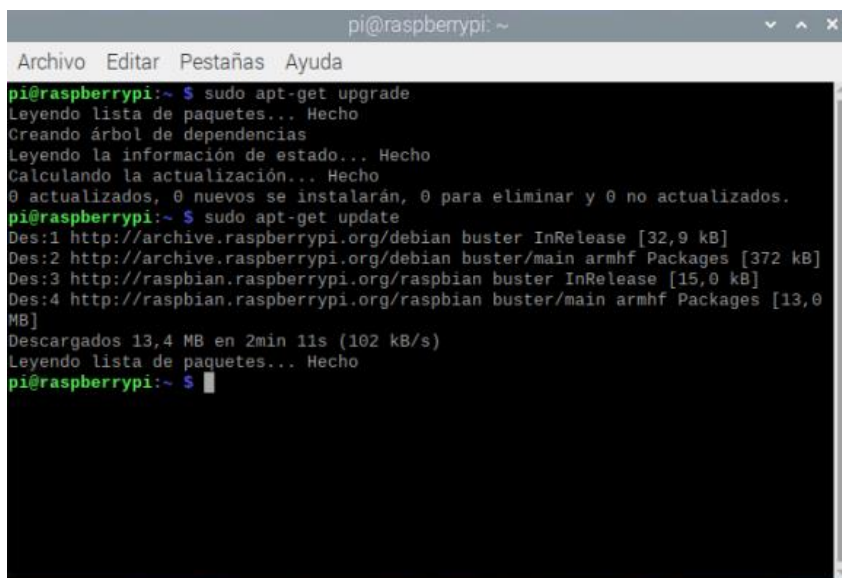
Una vez iniciado el ambiente virtual desde el protocolo VNC, se debe actualizar los paquetes desde la pestaña de Terminal.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 16 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



```

pi@raspberrypi:~ $ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
pi@raspberrypi:~ $ sudo apt-get update
Des:1 http://archive.raspberrypi.org/debian buster InRelease [32,9 kB]
Des:2 http://archive.raspberrypi.org/debian buster/main armhf Packages [372 kB]
Des:3 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Des:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0 MB]
Descargados 13,4 MB en 2min 11s (102 kB/s)
Leyendo lista de paquetes... Hecho
pi@raspberrypi:~ $

```

Figura 16. Actualización de paquetes.


Paso 3. Programación Raspberry Pi como servidor SMTP.

Se inicializa el entorno virtual de Thonny Python para la programación:

El protocolo SMTP es representado mediante la librería “SMTPlib” de Python, es decir dentro de la codificación debe ir importado SMTPlib para poder funcionar la clase “SMTP”. Para la realización de la programación de este protocolo se necesita tener activos correos electrónicos. Para esta práctica se suministrarán dos correos creados por autores:

avisosdatacenter@gmail.com
 Contraseña: raspberrypi4
repciondatacenter@gmail.com
 Contraseña: raspberrypi4

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 17 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

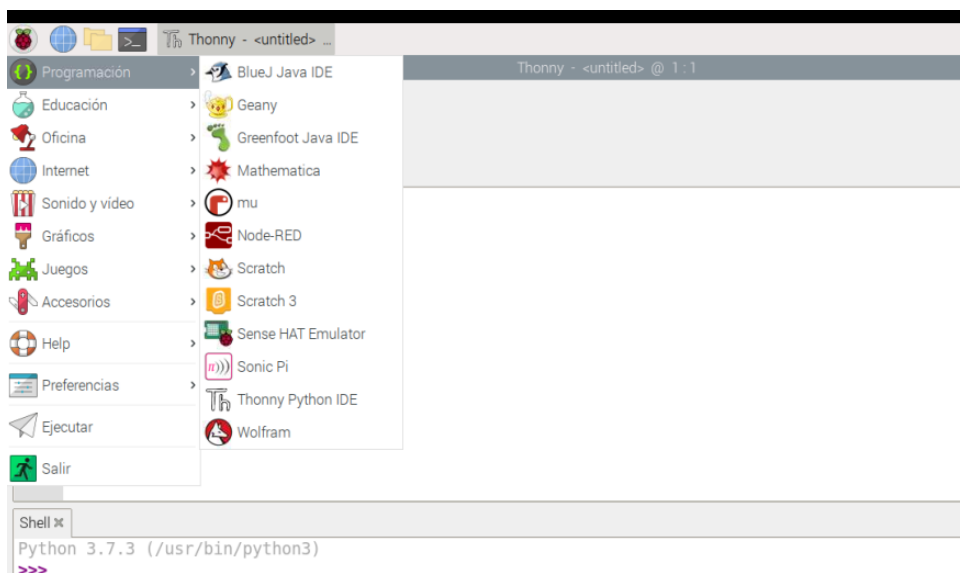


Figura 17. Entorno virtual Thonny Python

Una vez dentro de la plataforma Thonny Python se debe ingresar las siguientes líneas de código:

Cada línea de código tendrá su comentario para que sea entendible para los estudiantes.

CÓDIGO:


```
import smtplib
```

#lo primero es importar la librería de SMTP de Python la cual es Smtplib

```
from email.mime.text import MIMEText
```

#Posterior llamamos a la clase de Mimetext para que nos permita la comunicación de correos entre dispositivos mediante el formato MIME.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:


		REVISION 1/1	Página 18 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```

correo_envio = 'avisosdatacenter@gmail.com'
contraseña = 'raspberrypi4'
correo_destino = 'repciondatacenter@gmail.com'
msg = MIMEText("Hola Mundo, estoy usando mi raspberry Pi para enviar correos por SMTP")
#La variable msg corresponde a mensaje.
#Dentro de las comillas podemos escribir el mensaje que queremos enviar.
msg['Subject'] = 'PRÁCTICA 2'
#Subject significa Asunto, con qué asunto lo vas a enviar tu correo.
#Dentro de las comillas podrás editar el Asunto con el que desees que llegue el correo.
msg['From'] = correo_envio
#Declaramos de cual que correo lo vamos a enviar.
msg['To'] = correo_destino
#Declaramos cual es el correo que va a recibir.
#A partir de aquí debemos declarar los parámetros para utilizar el protocolo SMTP.
server = smtplib.SMTP('smtp.gmail.com',587)
#Declaramos el método SMTP de la clase Smtplib.
#Declaramos el nombre del servidor del correo electrónico que vamos a utilizar en nuestro
caso son correos de Gmail.
#Declaramos el puerto de comunicación del correo que estamos utilizando.
server.starttls()
server.login(correo_envio,contraseña)
server.sendmail(correo_envio,correo_destino,msg.as_string())
print("Su correo se ha procesado con éxito.")
server.quit

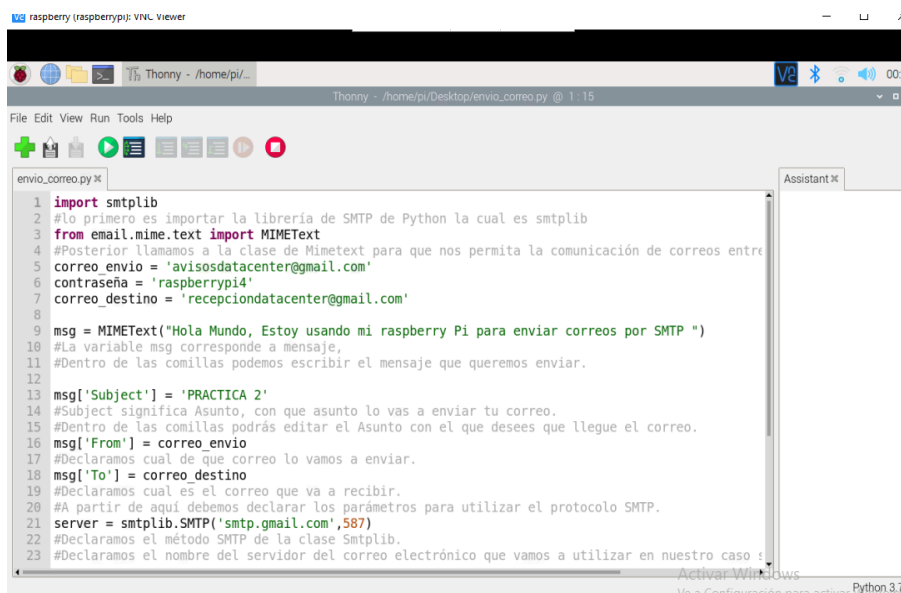
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 19 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

NOTA IMPORTANTE cada correo tiene su servidor de correo y su puerto de comunicación. Es decir, si usamos un correo distinto a Gmail debemos buscar el nombre del servidor.

#le adjunto link de búsqueda <https://support.microsoft.com/es-es/topic/c%C3%B3mo-configurar-una-cuenta-de-correo-electr%C3%B3nico-de-internet-en-outlook-2013-o-2016-3d900107-3c86-a326-6b8c-f214d10a6017>



```


1 import smtplib
2 #Lo primero es importar la librería de SMTP de Python la cual es smtplib
3 from email.mime.text import MIMEText
4 #Posterior llamamos a la clase de MimeText para que nos permita la comunicación de correos entre
5 correo_envio = 'avisosdatacenter@gmail.com'
6 contraseña = 'raspberrypi4'
7 correo_destino = 'repciondatacenter@gmail.com'
8
9 msg = MIMEText("Hola Mundo, Estoy usando mi raspberry Pi para enviar correos por SMTP ")
10 #La variable msg corresponde a mensaje,
11 #Dentro de las comillas podemos escribir el mensaje que queremos enviar.
12
13 msg['Subject'] = 'PRACTICA 2'
14 #Subject significa Asunto, con que asunto lo vas a enviar tu correo.
15 #Dentro de las comillas podrás editar el Asunto con el que desees que llegue el correo.
16 msg['From'] = correo_envio
17 #Declaramos cual de que correo lo vamos a enviar.
18 msg['To'] = correo_destino
19 #Declaramos cual es el correo que va a recibir.
20 #A partir de aquí debemos declarar los parámetros para utilizar el protocolo SMTP.
21 server = smtplib.SMTP('smtp.gmail.com',587)
22 #Declaramos el método SMTP de la clase Smtplib.
23 #Declaramos el nombre del servidor del correo electrónico que vamos a utilizar en nuestro caso s

```

Figura 18. Programación en Thonny Python.

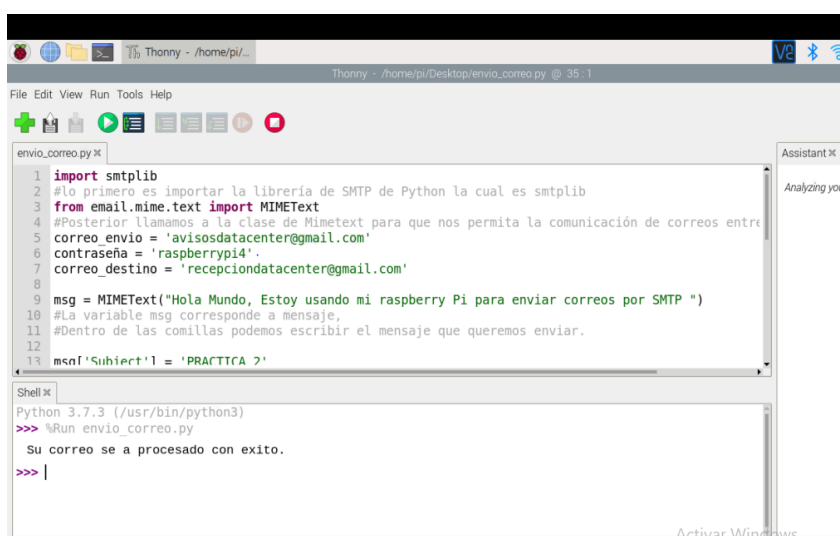
Nota Importante: Si usas tu correo personal de Gmail, se debe permitir unos accesos dentro de tu correo. En la pestaña de configuración, Seguridad, y activamos el acceso de APP menos seguras para que el raspberry pueda enviarte el correo. Este procedimiento solo se realiza en el correo electrónico que envía el mensaje.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 20 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 4. Ejecución del código ingresado.

Una vez cargado el código, nos solicitará guardar el documento con un nombre, en nuestro caso lo guardamos como envio_correo, podemos ejecutar seleccionando el botón de Play color verde, y el programa automáticamente leerá su codificación y les mostrará el mensaje “Su correo se ha procesado con éxito.” En una pestaña de nombre Sheel.



```

envio_correo.py x
1 import smtplib
2 #Lo primero es importar la librería de SMTP de Python la cual es smtplib
3 from email.mime.text import MIMEText
4 #Posterior llamamos a la clase de MimeText para que nos permita la comunicación de correos entre
5 correo_envio = 'avisosdatacenter@gmail.com'
6 contraseña = 'raspberrypi4'
7 correo_destino = 'repciondatacenter@gmail.com'
8
9 msg = MIMEText("Hola Mundo, Estoy usando mi raspberry Pi para enviar correos por SMTP ")
10 #La variable msg corresponde a mensaje,
11 #Dentro de las comillas podemos escribir el mensaje que queremos enviar.
12
13 msg['Subject'] = 'PRACTICA 2'

Shell x
Python 3.7.3 (/usr/bin/python3)
>>> %Run envio_correo.py
Su correo se a procesado con exito.
>>> |

```

Figura 18. Resultado de programación en Thonny Python envío de correo.

Paso 5. Comprobación de Mensaje recibido desde Gmail.


Para comprobar que el código ha sido procesado con éxito abrimos el correo de Gmail que está registrado como de receptor para esta codificación:

Recordemos las credenciales

repciondatacenter@gmail.com

Contraseña: raspberrypi4

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 21 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

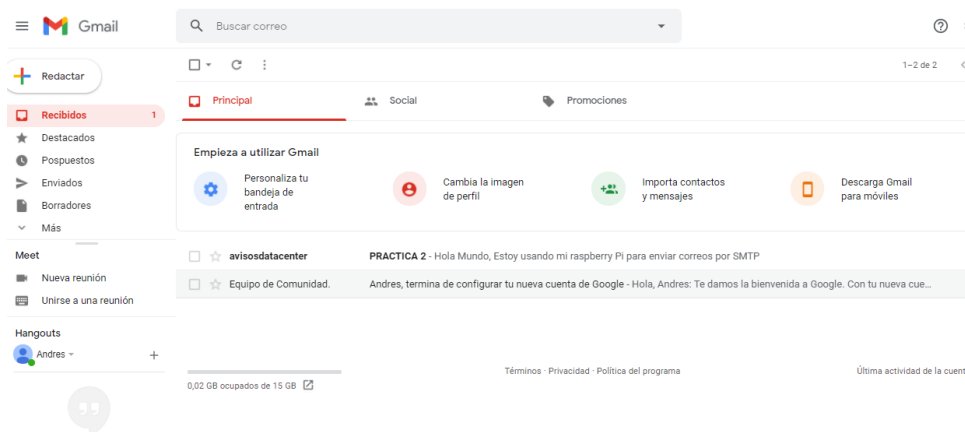


Figura 19. Comprobación del recibo del correo.


Como se puede visualizar el correo ha llegado con éxito.

El correo de origen es avisosdatacenter@gmail.com, el Asunto es Practica 2 y el mensaje es Hola Mundo, Estoy usando mi raspberry pi para enviar correos por SMTP, el correo de destino es recepciondatacenter@gmail.com.

RECURSOS UTILIZADOS

- Kit Raspberry Pi.
- Conector alimentación 5V 3A.
- Tarjeta MicroSD.
- Laptop.
- WI-FI.
- Protocolo VNC.
- Servidor VSFTPD.
- Cliente Filezilla.
- Correos electrónicos.
- Ambiente Virtual Thonny Python.
- Monitor

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 22 de 22
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

5) RESULTADOS OBTENIDOS

Mediante la implementación de esta práctica hemos comprendido el funcionamiento de un servidor FTP, conociendo que se basa en una relación Cliente-Servidor donde existe comunicación mediante la RED de forma unidireccional.

Adicional conocer sobre el protocolo SMTP nos explica de forma práctica cual es el proceso que atraviesa un correo en la red desde su origen hasta su destino.


6) CONCLUSIONES

Conocer las implementaciones de los protocolos FTP y SMTP nos abren el paso a el mundo del IOT permitiéndonos crear nubes de almacenamiento y recepción de alarmas de forma remota. Lo cual es sumamente útil para los proyectos de tecnología.

7) BIBLIOGRAFÍA UTILIZADA

<https://www.raspberrypi.org/forums/viewtopic.php?t=174479>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 1 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Anexo 3. Práctica realizada para la configuración y conexión de un sistema de circuito cerrado de televisión mediante Raspberry Pi.

ELECTRÓNICA Y TELECOMUNICACIONES

PRÁCTICA # 3


DOCENTE: ING. LENIN CEVALLOS

TIEMPO ESTIMADO 2 HORAS

NÚMERO DE ESTUDIANTES: 20

TEMA: “CONFIGURACIÓN Y CONEXIÓN DE UN SISTEMA DE CIRCUITO CERRADO DE TELEVISION MEDIANTE RASPBERRY PI”

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 2 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

1) OBJETIVO GENERAL.

Aprender a configurar al módulo Raspberry Pi como un sistema de circuito cerrado de televisión.

2) OBJETIVOS ESPECÍFICOS.

- Identificar las herramientas necesarias para los trabajos de programación de una placa de Raspberry Pi.
- Realizar la programación para configurar al raspberry Pi como CCTV.
- Realizar la programación para captura de imágenes y grabación de videos.
- Realizar la configuración para visualizar las transmisiones en vivo del CCTV mediante app móvil en la misma LAN.

3) MARCO TEÓRICO.

RASPBERRY PI 4 MODEL B


Es una placa del tamaño de una tarjeta de crédito considerado como un microcontrolador que soporta varios componentes necesarios en un computador de escritorio Común.

Posee puertos de entrada y salida que permite conectar el microcontrolador a medios físicos externos tales como dispositivos de teclado, ratones, pantallas, cámaras, sensores y actuadores.

Especificaciones:

- Procesador central.
- Procesador Gráfico (GPU).
- Módulo de memoria RAM.
- Conector RJ45.
- 2 buses USB 2.0.
- 2 buses USB 3.0.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

- Wifi y Bluetooth integrado.
- Interfaz para cámara y pantalla táctil.
- Salida Analógica de audio/estéreo por Jack de 3,5 mm.
- Salida digital de video MicroHDMI y audio.
- Pines de entrada y salida de propósito General (GPIO).
- 2 puertos de conexión MicroHDMI.
- 1 conector de alimentación MicroUSB tipo C.
- Lector de Tarjeta SD.

CCTV.


El circuito cerrado de televisión o CCTV es una tecnología para la supervisión y vigilancia a través de dispositivos como cámaras dirigidas a una variedad de ambientes y actividades y proyectadas en un computador o monitor. Los dispositivos de grabación están conectados en un circuito cerrado con una central común, las cuales crean una cadena de imágenes que pueden ser proyectadas para su control.

PYTHON 3

Python es un lenguaje de programación con un amplio campo de aplicación con énfasis en la legibilidad y ponderación de palabras claves estándar en inglés con fácil manejo al escribir y leer. Es un lenguaje interpretativo e interactivo con el usuario está orientado a objetos y ofrece estructuras de datos de alto nivel por un tipado dinámico y puntual.

Su principal plataforma IDE de visualización es Thonny, un entorno de desarrollo de Python3 y es capaz de presentar los programas de parte natural y de fácil comprensión para principiantes.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 4 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

CÁMARA

La cámara compatible con el microcontrolador Raspberry Pi son las cámaras 5MPX OV5647 la cual cuenta con un buzz de dato de 15 cm para la conexión con la placa Raspberry mediante el puerto de cámara. La cámara OV5647 funciona con un voltaje bajo, su sensor de imagen CMOS de 5 megapíxeles nos brinda un rendimiento alto en resolución y nos proporciona una salida de video de 2592x1944 mediante la tecnología OmniBSITM. El núcleo del sensor OV5647 transmite datos de píxeles generales a una velocidad constante de fotogramas, indicadas por VSYNC y HREF.

VNC

Es una aplicación de procedimientos libre, su arquitectura es cliente – servidor lo cual nos permite visualizar la información del ordenador servidor remotamente a través de un dispositivo cliente.

FLASK


Es un marco de micro Web Ligero que genera un gran puente entre las páginas WEB de Python y HTML.

RASPICAMREMOTE

Aplicación descargable para plataformas Android e IOS, compatible y configurable desde Raspberry Pi, fácil de usar, transmite lo que la cámara este enfocando con un retraso de 5 milisegundos.

Excelente para revisión remota sin estar en el IDE de raspberry Pi. Ideal para la vigilancia y control. Permite cambiar las tonalidades de enfoque, capturar imágenes y grabar videos.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 5 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Accesorios para usar:

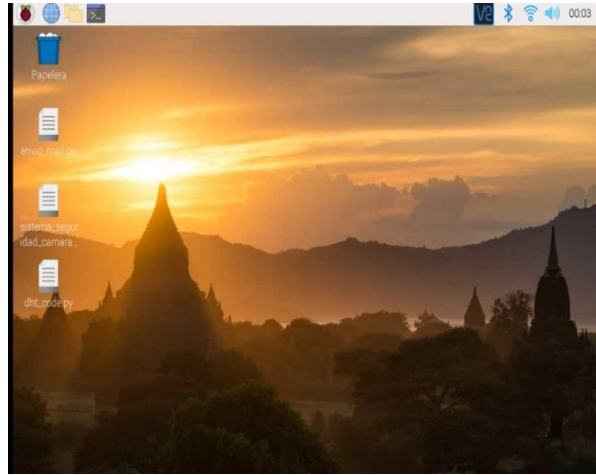
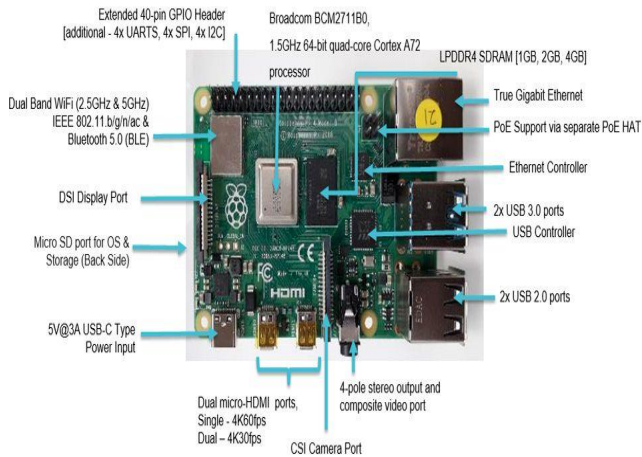


Figura 1. Raspberry Pi model 4B


Figura 2. Entorno virtual Raspberry Pi OS



Figura 3. Cámara Raspberry PI.

Figura4. App Raspicam Remote.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 6 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

4) MARCO PROCEDIMENTAL

PROCEDIMIENTO

El procedimiento para realizar la configuración de la placa Raspberry Pi modelo 4B como servidor CCTV es mediante librerías ejecutables en la pestaña de Terminal y Thonny Python, los archivos ejecutables lo debemos programar y ejecutar en el entorno virtual de Raspberry Pi, esto bien lo podemos hacer visualizando en un monitor e ingresando caracteres mediante un teclado o habilitando los accesos remotos de VNC y SSH para ingresar desde nuestros ordenadores personales PC.

Para la práctica se necesitará una cámara Raspberry Pi, la cual irá capturando mediante la ejecución del código podrá ir capturando imágenes en vivo, grabando videos en tiempo real y permitirá conectar mediante la aplicación móvil RaspicamRemote, los dispositivos conectados a la misma red que la placa de Raspberry Pi podrán ingresar mediante la app conociendo la dirección IP del Raspberry Pi y ver las imágenes en vivo de la transmisión.

Como declaración importante es tener en consideración las conexiones en los puertos y la alimentación correcta a la placa de Raspberry Pi.

RASPBERRY PI COMO CCTV

Paso 1. Iniciamos Sesión en Raspberry Pi OS mediante VNC.

Para inicializar la práctica se debe energizar a la placa Raspberry Pi 4B con 5v y 3A, inmediatamente iniciamos sesión mediante nuestro protocolo VNC.


Recordemos nuestras credenciales de acceso:

Dirección Ip del raspberry Pi: *192.168.39.167*.

Usuario: pi

Contraseña: raspberrypi

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 7 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

(NOTA IMPORTANTE), si no recuerda la dirección Ip del raspberry conecte la placa a un monitor mediante un cable HDMI y en la pestaña comandos realice la búsqueda mediante el código ifconfig ingresado por teclado.

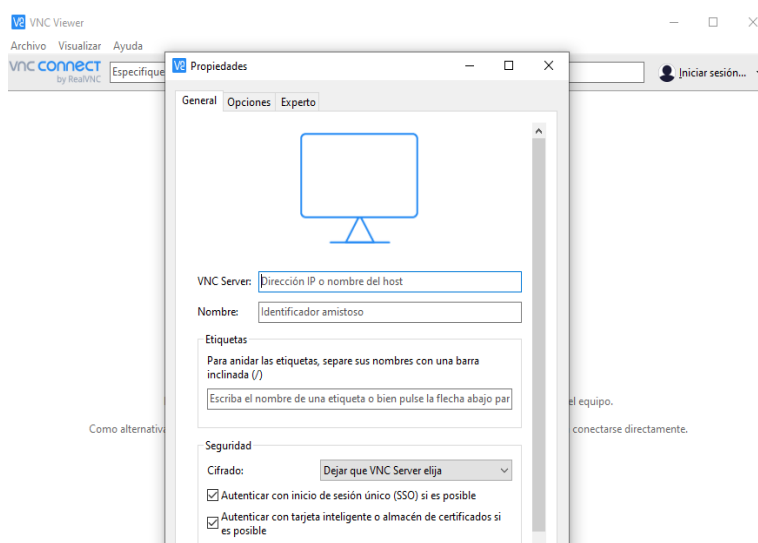


Figura 5. Conexión de acceso remoto VNC.

Paso 2. Actualización de paquetes Raspberry Pi.

Una vez iniciado la conexión con el ambiente virtual desde el protocolo “VNC”, se procede a iniciar la pestaña de Terminal.


Se recomienda actualizar los paquetes del sistema operativo con el siguiente comando:

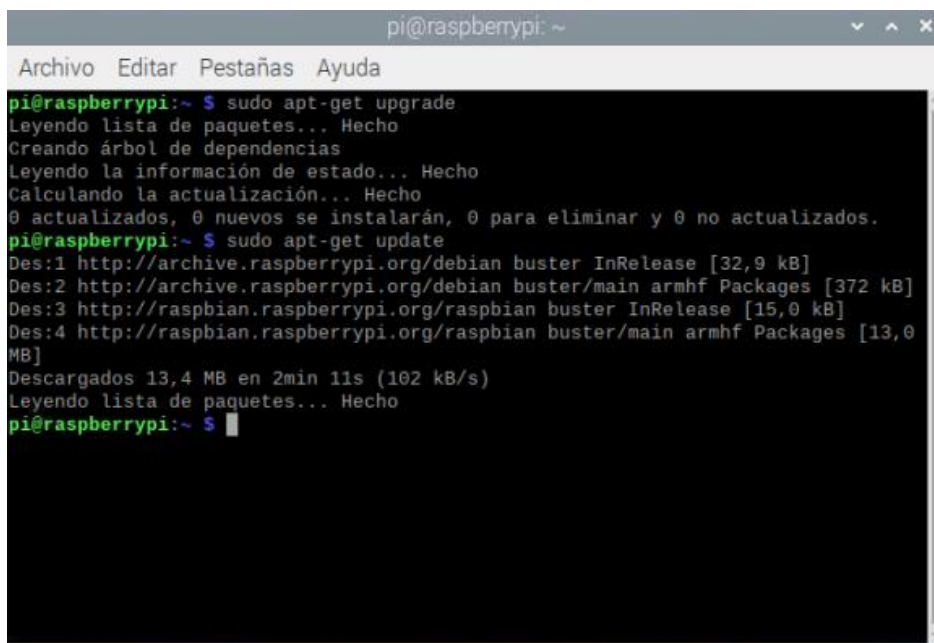
```
sudo apt-get update
```

Paso siguiente, se debe actualizar todos sus paquetes instalados a sus últimas versiones con el siguiente comando:

```
sudo apt-get upgrade
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 8 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
pi@raspberrypi:~ $ sudo apt-get update
Des:1 http://archive.raspberrypi.org/debian buster InRelease [32,9 kB]
Des:2 http://archive.raspberrypi.org/debian buster/main armhf Packages [372 kB]
Des:3 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Des:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0 MB]
Descargados 13,4 MB en 2min 11s (102 kB/s)
Leyendo lista de paquetes... Hecho
pi@raspberrypi:~ $

```

Figura 6.

Actualización de paquetes.

Paso 3. Habilitación de Puerto de Pi Cámara.


Después de realizar la conexión a través de VNC y actualizar los paquetes de raspberry Pi en la configuración de Pi OS, se debe habilitar los puertos Pi Cámara Port.

Para conseguirlo abrimos la ventana de la terminal y se escribe el siguiente código:

```
Sudo raspi-config
```

Seleccione la opción de interfaz, puerto de cámara y seleccione habilitar, guarde los cambios, si el programa le pide reiniciar “reboot” ejecute el sí.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 9 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

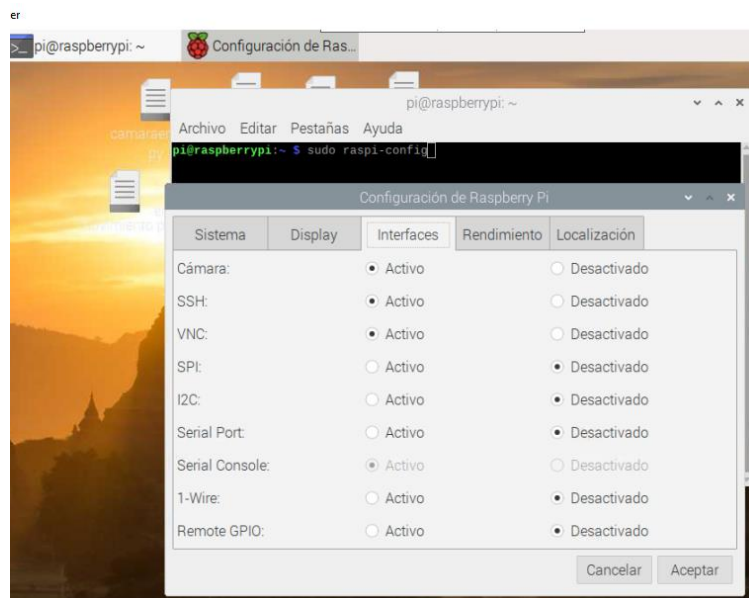


Figura 7. Habilitación de puerto de Cámara.

Para comprobar que se ha habilitado con éxito el puerto de la cámara, se debe la cámara a la ranura de la raspberry, abrimos la ventana de Terminal y se ingresa por comando una línea para captura una toma de fotos a través del módulo de la cámara:


```
Raspistill -o Desktop/image.jpg
```

En donde se debe reemplazar la palabra image con otro nombre que el usuario desee.

Para el ejemplo se utilizará: raspistill -o Desktop/prueba.jpg

Una vez ejecutado el comando el procesador de Thonny Python procesará el código y capturará imagen y la guardará en el escritorio del entorno Raspberry PI OS con el nombre que se ha seleccionado.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 10 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

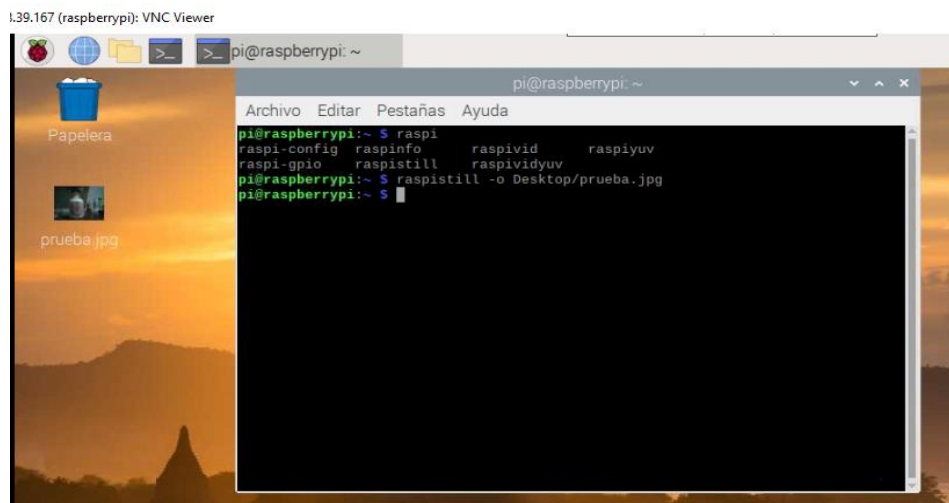



Figura 8. Captura de foto por comando.



Figura 9. Cámara Raspberry Pi Capturando imagen.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 11 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Una vez capturada la imagen se podrá visualizar accionando sobre el icono doble clic con el mouse.

El entorno virtual mostrará la imagen captada, la cámara podrá capturar fotos cuantas veces se ejecute el código, si se lo realiza sin cambiar el nombre el microcontrolador sobrescribirá la imagen con el mismo nombre sobre la anterior.

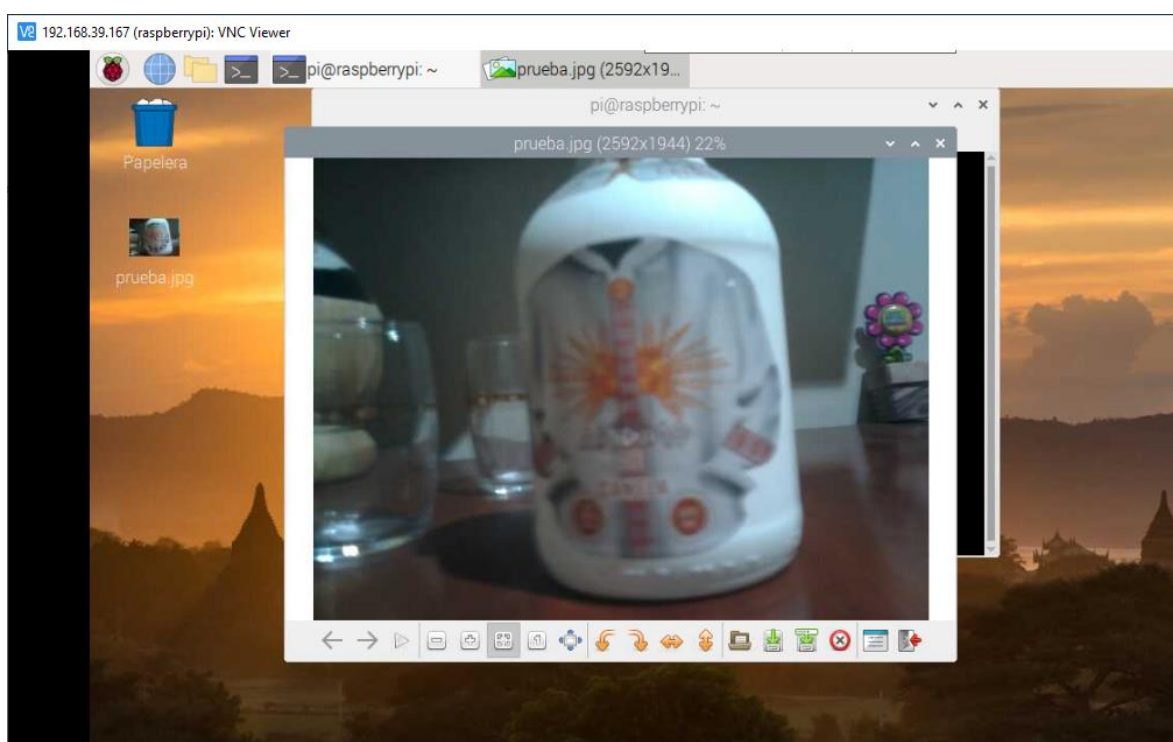



Figura 10. Imagen de nombre prueba.jpg.

Paso 4. Creación de Código para Captura de pantalla en cámara HD.

El tema principal de esta práctica es poder utilizar la cámara de Raspberry Pi como un visualizador de imagen en vivo de lo que transmite la lente del dispositivo.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 12 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Como se presentó en el Paso 3, la cámara de Raspberry Pi funciona con comandos para poder capturar fotos, entonces debemos crear líneas de código que permitan al microcontrolador utilizar la cámara para según nosotros queramos. En este paso se utilizará las siguientes líneas de código para poder ver en vivo la imagen y que se muestren en el monitor o ventana virtual mientras se está ejecutando.

Para ello, se inicia el ambiente de programación virtual Thonny Python y se ingresa las siguientes líneas de código.

Códigos:

```
importpicamera
from time importsleep
```

#importamos librería de Cámara, librería de tiempo, librería de sleep

```
camera = picamera.PiCamera()
camera.vflip = None
```

#vflip es un comando que permite ajustar la imagen para que no salga volteada

```
camera.start_preview(fullscreen=True)
```

#muestra imagen en vivo en pantalla de monitor através de hdmi

```
sleep(30)
```

#muestra la imagen por 30 segundos


```
camera.stop_preview()
```

#la proyección se detiene

Los comentarios en el ambiente de programación Thonny Python se escriben anticipando el signo de numeral (#).

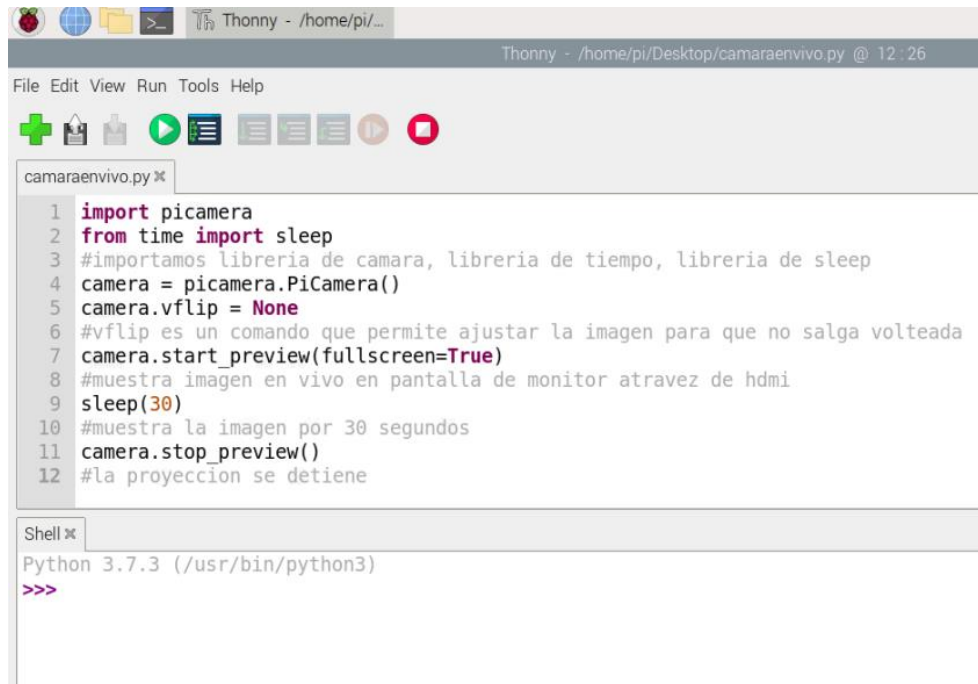
Se guarda el código en formato (xxx.py), en el caso de esta práctica se guardó el documento con el nombre camaraenvivo.py. Una vez guardado ejecutamos mediante el

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 13 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

botón Run del ambiente virtual que está representado por un círculo verde en la barra de herramientas.

Figura
Captura
pantalla
códigos



```

1 import picamera
2 from time import sleep
3 #importamos libreria de camara, libreria de tiempo, libreria de sleep
4 camera = picamera.PiCamera()
5 camera.vflip = None
6 #vflip es un comando que permite ajustar la imagen para que no salga volteada
7 camera.start_preview(fullscreen=True)
8 #muestra imagen en vivo en pantalla de monitor atravez de hdmi
9 sleep(30)
10 #muestra la imagen por 30 segundos
11 camera.stop_preview()
12 #la proyeccion se detiene

```

```

Shell x
Python 3.7.3 (/usr/bin/python3)
>>>

```

10.
de
del

documento camaraenvivo.py.

Luego de ejecutar el microcontrolador ordenará a la cámara Raspberry a visualizar la imagen que tenga en frente de su lente y enviar la información mediante su buzz de conexión hasta el centro de procesamiento que a su vez lo proyectará en la pantalla ya sea HDMI o monitor que se tiene conectado en el puerto de la Placa SBC de la Raspberry Pi 4B por un tiempo de 30 segundos.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:


		REVISION 1/1	Página 14 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



Figura 11. Monitor proyectando el código de camaraenvivo.py.

Paso 5. Creación de código para grabación de video mediante cámara HD.


Para la ejecución de grabación de un video en vivo, se debe modificar y añadir ciertas líneas de código.

Códigos:

```
importpicamera
from time importsleep
camera = picamera.PiCamera()
camera.vflip = None
camera.resolution = (800,600)
camera.start_recording('output-video.h264')
sleep(30)
camera.stop_preview()
```

Paso siguiente, se guarda el código en formato (xxx.py), en el caso de esta práctica se guardó el documento con el nombre grabacionenvivo.py. Una vez guardado se ejecutará

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 15 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

mediante el botón Run del ambiente virtual que está representado por un círculo verde en la barra de herramientas.

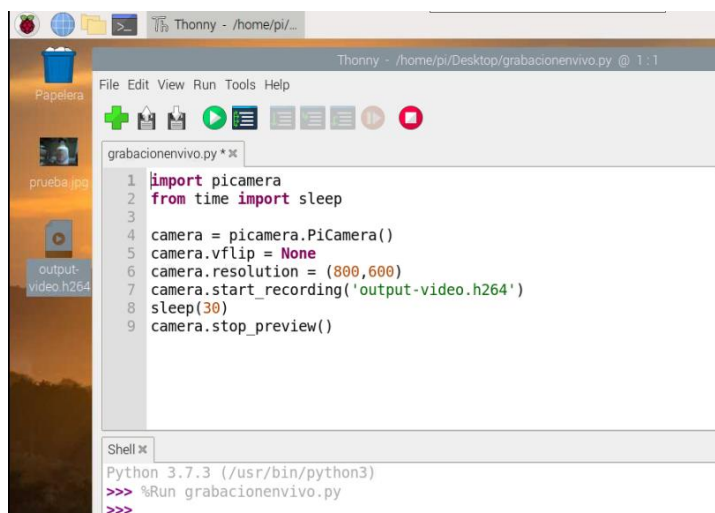


Figura 12. Captura de pantalla códigos del documento grabacionenvivo.py.

Nota importante: Raspberry Pi guarda todos los videos en un formato que no puede ser leído normalmente por nuestras PC, esto debido a que contiene un reproductor por defecto y sus grabaciones se guardan en formato (.h264), para ello debemos convertir estos videos en un formato que si se pueda reproducir como lo es mp4.

Para poder convertir del formato h264 a mp4, se lo puede realizar mediante una aplicación de nombre “gpac”.


Para iniciar la instalación debemos abrir la ventana de comando y escribir:

```
Sudo apt-get install -y gpac
```

Luego de esto podemos convertir nuestros videos con la línea de código siguiente:

```
MP4Box -fps 30 -add output-video.h264 prueba.mp4
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 16 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Se debe exportar la librería de “mp4box”, con la cantidad de cuadros por segundo (30), se añade el archivo a convertir (output-video.h264) y se nombra el nombre de cómo queremos que se llame el archivo en mp4 (prueba.mp4).

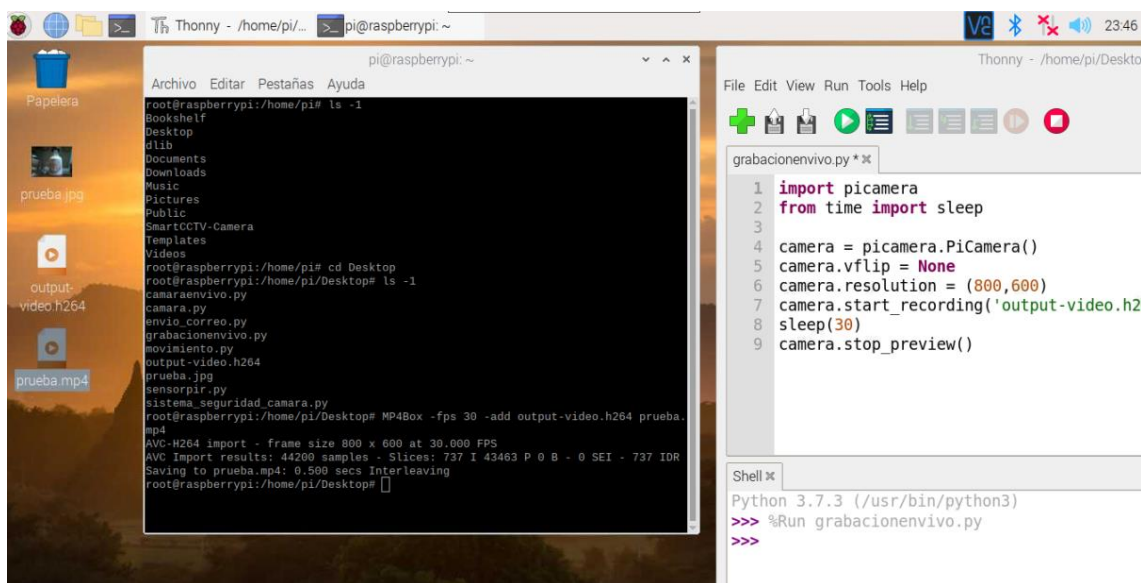



Figura 13. Captura de pantalla conversión de video prueba.mp4.

Paso 6. Visualización de CCCTV mediante App Móvil.

Si bien los pasos 4 y 5 muestran la capacidad de una placa Raspberry Pi para crear una cámara espía con una oportunidad de que pueda grabar y proyectar en pantalla lo que su lente toma del ambiente, sería ideal que nos permita conectar al sistema de vigilancia sin necesidad de escribir líneas de comando.

Para ello se debe utilizar la aplicación disponible para Android e IOS de nombre RasiCam Remote. El cual nos permite conectarnos remotamente a nuestro microcontrolador Raspberry Pi 4B únicamente conociendo la dirección IP que este tenga registrado en la Red LAN a la cual esté conectado.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 17 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

En nuestro caso:

Dirección Ip: 192.168.39.167
Nombre del dispositivo: pi (nombre por defecto)
Password: raspberrypi
Puerto de conexión: 22.

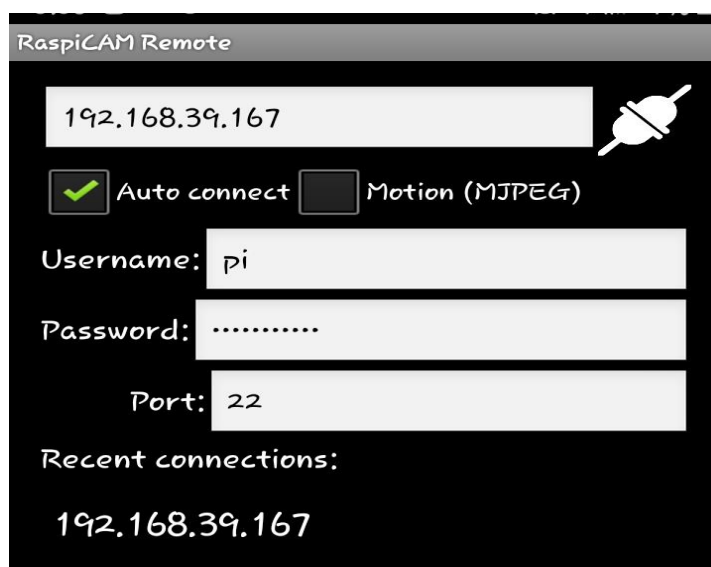



Figura 14. Ambiente virtual de aplicación Raspicam Remote.

Una vez establecida la conexión remota mediante la aplicación se obtendrá visualización en el dispositivo móvil de que la cámara este mostrando.

La proyección tendrá un pequeño retraso dependiendo de la señal de internet, y podrá capturar imágenes mediante la aplicación de lo mostrado.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 18 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

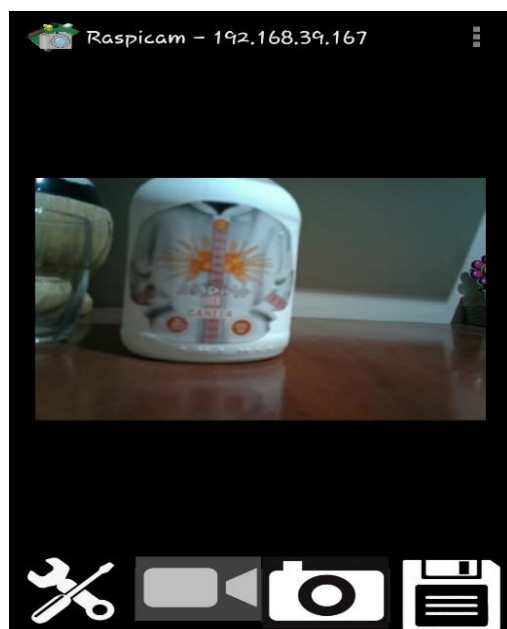



Figura 15. Captura de pantalla acceso remoto a cámara Raspberry Pi mediante App.

5) RECURSOS UTILIZADOS

- Kit Raspberry Pi.
- Conector alimentación 5V 3Amp.
- Tarjeta MicroSD.
- Laptop.
- WI-FI.
- Protocolo VNC.
- Servidor VSFTP.
- Ambiente Virtual Thonny Python.
- App Móvil RaspiCamRemote.
- Monitor.
- Conector HDMI.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 19 de 19
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

6) RESULTADOS OBTENIDOS

Mediante la implementación de esta práctica, se pudo comprender y visualizar de una forma didáctica cómo funciona las cámaras de seguridad de Raspberry Pi.

Al finalizar la práctica se han obtenido conocimientos en captura de imagen, grabación de video, y acceso remoto al sistema de vigilancia sin necesidad de programación si no mediante una app móvil.


7) CONCLUSIONES

Con la práctica realizada se busca generar conocimiento a los usuarios de cómo poder crear un sistema de vigilancia sin necesidad de contratar una implementación con costos elevados y dispositivos robustos resaltando la importancia de estos para la seguridad personal.

8) BIBLIOGRAFÍA UTILIZADA

<https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/7>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 1 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Anexo 4. Práctica realizada para el control de acceso mediante uso de microcontrolador Arduino.

ELECTRÓNICA Y TELECOMUNICACIONES

PRÁCTICA # 4


DOCENTE: ING. LENIN CEVALLOS

TIEMPO ESTIMADO 2 HORAS

NÚMERO DE ESTUDIANTES: 20

TEMA: “CONFIGURACIÓN Y CONEXIÓN DE UN SISTEMA DE CONTROL DE ACCESO UTILIZANDO MICROCONTROLADOR ARDUINO Y TECNOLOGIA RFID”

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 2 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

1) OBJETIVO GENERAL.

Aprender a instalar y programar un control de acceso mediante el uso de un microcontrolador Arduino.

2) OBJETIVOS ESPECÍFICOS.

- Identificar las herramientas necesarias para las conexiones de un control de acceso
- Realizar programación mediante el uso de la plataforma de Arduino.
- Aprender sobre el uso de este sistema como seguridad de ingresos a áreas restringidas.
- Realizar un control de acceso mediante uso de sensor RFID.
- Realizar un control de acceso mediante uso de lector de huella.
- Realizar un control de acceso mediante uso de teclado matricial.

3) MARCO TEÓRICO.


Arduino Uno

Arduino Uno es una placa basada en un microcontrolador, con código abierto orientado al microchip Atmega328p y desarrollado por arduino.cc. La placa contiene entradas/salidas digitales y analógicas, en total tiene 14 pines digitales. 6 pines analógicos y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B

Especificaciones:

- Microcontrolador: ATmega328
- Voltaje de operación: 5V
- Voltaje de entrada (recomendado): 7-12V
- Voltaje de entrada (límites): 6-20V
- Pines de E/S digitales: 14 (de los cuales 6 proporcionan salida PWM)

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

- Pines de entrada analógica: 6
- Corriente DC por pin de E/S: 40 mA
- Corriente DC para 3.3V Pin: 50 mA
- Memoria Flash: 32 KB de los cuales 0,5 KB utilizados por el bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Velocidad de reloj: 16 MHz


Arduino Mega

Es una placa basada en un microcontrolador, tiene 54 entradas/salidas digitales, pueden ser usadas 15 como salidas PWM, 16 entradas analógicas, 4UARTs y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B

Especificaciones:

- Microcontrolador: ATmega2560
- Voltaje de operación: 5V
- Voltaje de entrada (recomendado): 7-12V
- Voltaje de entrada (límites): 6-20V
- Pines de E/S digitales: 54 (de los cuales 15 proporcionan salida PWM)
- Pines de entrada analógica: 16
- Corriente DC por pin de E/S: 40 mA
- Corriente DC para 3.3V Pin: 50 mA
- Memoria Flash: 256 KB de los cuales 8 KB utilizados por el bootloader
- SRAM: 8 KB (ATmega2560)
- EEPROM: 4 KB (ATmega2560)
- Velocidad de reloj: 16 MHz

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 4 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Modulo Lector RC522

El módulo RC522 es Lector-Grabador RFID 13.56MHz, posee comunicación SPI lo que permite trabajar fácilmente con la mayoría de los microcontroladores. Utiliza un sistema de modulación y demodulación para todo tipo de dispositivos pasivos RFID de 13.56MHz. El dispositivo maneja el ISO14443-A y soporta el algoritmo de encriptación Quick CRYPTO1 y MIFARE. El rango de detección de tags RFID es de aprox. 5-7cm. Compatible con Arduino, Pic, Raspberry Pi y más.

Los sistemas RFID son muy útiles para sistemas de control de acceso, seguridad electrónica, trazabilidad.

Tarjeta RFID

Es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio.

Lector de huella dactilar

El Lector de huella dactilar es un sensor biométrico que me permite leer, guardar e identificar huellas de todos los dedos, este dispositivo contiene un conector para alimentación de 5V y comunicación serie.


LCD Display

Es un dispositivo con pantalla de cristal liquida, conformado por pixeles de color o monocromos, el cual permite mostrar información al usuario

Teclado Matricial

Es un dispositivo formado por una matriz de pulsadores, ordenados en filas y columnas para que se pueda reducir el número de pines a usar para la conexión.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 5 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Modulo Relé

Es un dispositivo electromecánico que permite a un procesador como Arduino controlar cargas a un nivel tensión o intensidad muy superior a las que su electrónica puede soportar.

Solenoides

Se trata básicamente de una cerradura electrónica, diseñada para un gabinete de base o de seguridad o de puerta. Normalmente, el bloqueo está activo por lo que no puede abrir la puerta porque el lingote del solenoide está en el camino (no utiliza ningún poder en este estado). Cuando se aplica 9 ~ 12 VDC, el lingote se contrae para que no sobresalga más y la puerta se pueda abrir.

Buzzer

Es dispositivo que produce sonido de un mismo tono, es usado como señalización o alarma y es usado en diversos sistemas de seguridad

Accesorios para usar:

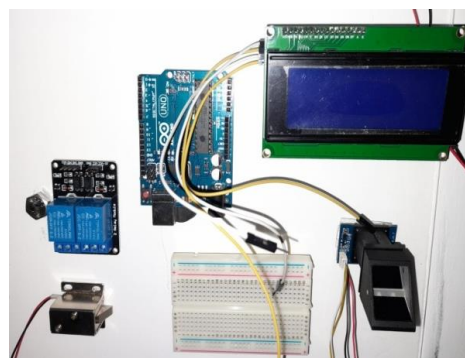



Figura 1. Kit de práctica con tarjetarfid **Figura2.**Kit de práctica con lector de huellas

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 6 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

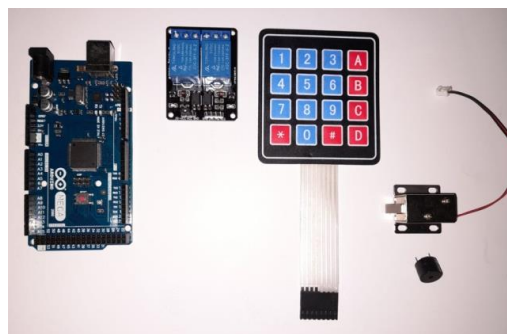


Figura 3. Kit de práctica con teclado matricial

4) MARCO PROCEDIMENTAL

4.1. CONTROL DE ACCESO CON TARJETA RFID

El procedimiento para realizar un control de acceso mediante uso de Arduino es reconocer los funcionamientos de que cada equipo, a que voltaje trabaja para no producir daños en ellos, y realizar la programación y conexiones de la manera correcta, para una mejor operatividad.


Paso 1: Verificar las entradas y salidas del Arduino.

Es fundamental identificar para una correcta programación en Arduino. En el Arduino UNO existen entradas y salidas analógicas (6) y digitales (16). Por un lado, las analógicas son utilizadas, generalmente, para dispositivos de entrada y nos permiten leer un rango de valores. En cambio, las digitales pueden tener a la entrada o a la salida 0V o 5V, esto nos indica que ese pin está en estado bajo (Low) o alto (High).

Paso 2: Conexiones de los dispositivos.

Procedemos a verificar que entradas se va a utilizar, cablear con jumpers hembra-macho la tarjeta RFID hacia el Arduino, luego conectar como alarma de sonido y visión un buzzer y leds como indicadores, para seguir con el uso de un solenoide que trabaja con una

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 7 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

fuente de 12V y un módulo relé, el cual se activa al pasar la tarjeta o llavero de identificación que podrá brindar el acceso y la chapa eléctrica se abrirá.

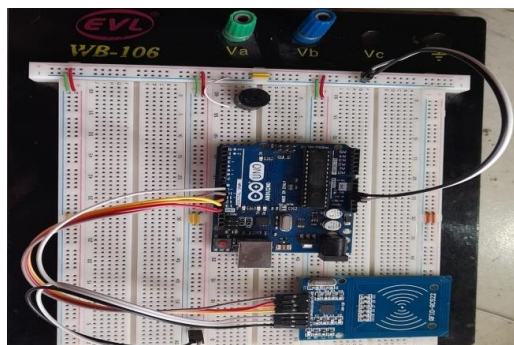


Figura 4. Conexión del módulo RFID con Arduino.

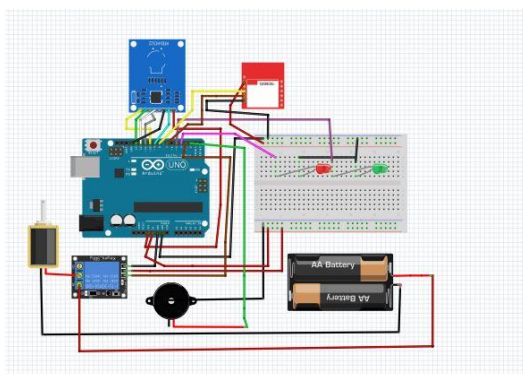



Figura 5. Conexión completa del control de acceso

Paso 3: Programación

Después de las conexiones, proceder a realizar la programación mediante el uso del programa de Arduino, para comenzar se debe descargar la librería MFRC522, seguir con la codificación con uso de parámetros establecidos por parte de cliente como por ejemplos números de tarjetas a usar, tiempo de espera para que no refleje error e intruso.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 8 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3.1. Iniciamos en incluir las librerías a usar.

Para iniciar la programación se debe definir las librerías de los dispositivos a usar, para que se pueda efectuar el proceso.

```
#include <SPI.h> // Include: permite añadir las librerías para poder hacer uso de
diferentes dispositivos

#include <MFRC522.h>

#include <SoftwareSerial.h>
```

Paso 3.2. Definir entradas a utilizar


```
#define SS_PIN 10
#define RST_PIN 9
#define LED_G 5
#define LED_R 4
#define RELAY 3
#define BUZZER 2
#define ACCESS_DELAY 2000
#define DENIED_DELAY 1000
```

Paso 3.3. Declarar variables para la ejecución del programa

```
void setup() {
  Serial.begin(9600);
  _buffer.reserve(50);
  sim.begin(9600);
  SPI.begin();
  mfr522.PCD_Init();
```

```
  pinMode(LED_G, OUTPUT);
  pinMode(LED_R, OUTPUT);
  pinMode(RELAY, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  noTone(BUZZER);
  digitalWrite(RELAY, LOW);
  Serial.println("Bienvenidos");
  Serial.println("Porfavor ingrese tarjeta");
  Serial.println();}
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 9 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		


Paso 3.4. Ejecutamos los comandos de los resultados a obtener

```

voidloop() {
if(! mfr522.PICC_IsNewCardPresent()){
return;}
if(! mfr522.PICC_ReadCardSerial()){
return;}
Serial.print("Tag");
Stringcontent="";
byte letter;
for(byte i=0; i < mfr522.uid.size; i++){
Serial.print(mfr522.uid.uidByte[i] < 0x10 ? "0:" " ");
Serial.print(mfr522.uid.uidByte[i], HEX);
content.concat(String(mfr522.uid.uidByte[i]<0x10 ? "0" : " "));
content.concat(String(mfr522.uid.uidByte[i], HEX));}
Serial.println();
Serial.print("Mensaje:");
content.toUpperCase();
if (content.substring(1) == "29 74 C2 98"){

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:


		REVISION 1/1	Página 10 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```

else {
Serial.println("Acceso denegado");
digitalWrite(LED_R, HIGH);
tone(BUZZER, 300);
delay(DENIED_DELAY);
digitalWrite(LED_R, LOW);
noTone(BUZZER);

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 11 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

4.2. CONTROL DE ACCESO CON LECTOR DE HUELLAS

El procedimiento para realizar un control de acceso mediante uso de Arduino, es reconocer los funcionamientos de que cada equipo, a que voltaje trabaja para no producir daños en ellos, y realizar la programación y conexiones de la manera correcta, para una mejor operatividad.

Paso 1: Verificar las entradas y salidas del Arduino.

Es fundamental conocerlas para empezar a programar Arduino. En el Arduino UNO existen entradas y salidas analógicas (6) y digitales (16). Por un lado, las analógicas son utilizadas, generalmente, para dispositivos de entrada y nos permiten leer un rango de valores. En cambio, las digitales pueden tener a la entrada o a la salida 0V o 5V, esto nos indica que ese pin está en estado bajo (Low) o alto (High).

Paso 2: Conexiones de los dispositivos.

Se procede a verificar qué entradas se va a utilizar, cablear con jumpers al lector de huellas hacia el Arduino, luego conectar como alarma de sonido un buzzer, para seguir con el uso de un solenoide que trabaja con una fuente de 12V.y un módulo relé el cual se activa al poner el dedo de identificación al lector de huellas que podrá brindar el acceso y la chapa eléctrica se abrirá

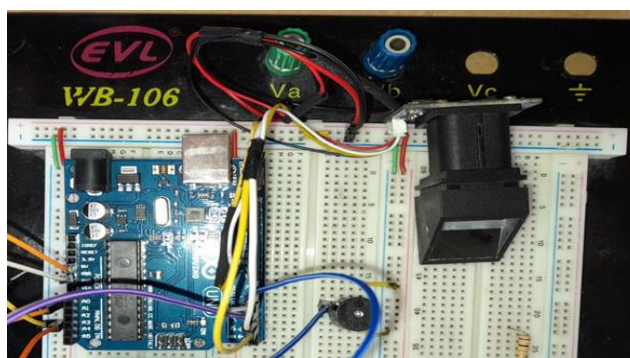



Figura 6. Conexión del lector de huella con Arduino.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 12 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

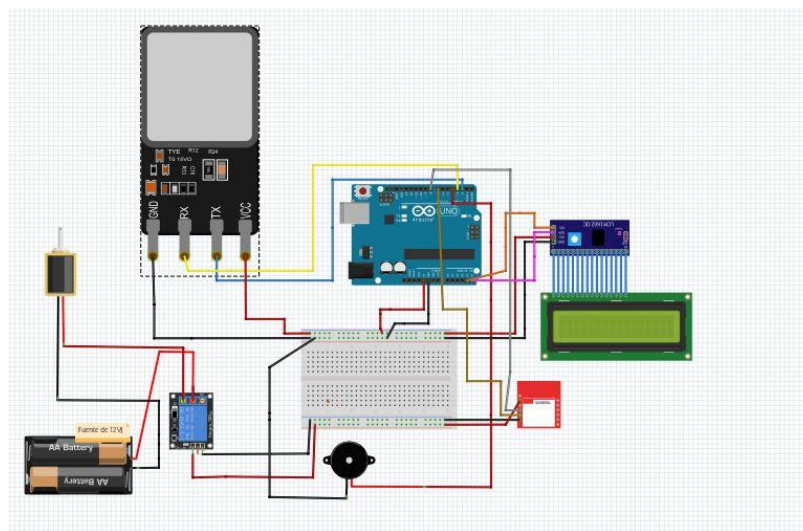


Figura
completa del control de acceso.

7. Conexión

Paso 3: Programación


Después de las conexiones, Se debe proceder a realizar la programación mediante el uso del entorno virtual de Arduino, para comenzar se debe descargar la librería “adafruit-fingerprint”, seguir con la codificación con uso de parámetros establecidos por parte del cliente como por ejemplo números de huellas a usar, tiempo de espera para que no refleje error e intruso.

Paso 3.1. Iniciamos en incluir las librerías a usar.

Como paso inicial, se debe definir las librerías de los dispositivos a usar, para que se pueda efectuar el proceso:

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 13 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3.2. Definir entradas a utilizar

```

intgetFingerprintIDez();
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
SoftwareSerialmySerial(2, 3);
SoftwareSerialsim(7, 8);
Adafruit_Fingerprintfinger = Adafruit_Fingerprint(&mySerial);
LiquidCrystal_I2C lcd(0x27, 20,4);
int ID_reg1 = 2;
int ID_reg2 = 8;
intgetFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) return -1;
  p = finger.image2Tz();
  if (p != FINGERPRINT_OK) return -1;
  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK) return -1;

  // found a match!

  Serial.print("Found ID #");


  Serial.print(finger.fingerID);

  Serial.print(" with confidence of ");

  Serial.println(finger.confidence);}

```


Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 14 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3.3. Declarar variables para la ejecución del programa

```
voidsetup() {
Serial.begin(9600);
Serial.println("Prueba de seguridad");
finger.begin(57600);
lcd.init();
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Bienvenidos");
lcd.setCursor(0,1);
lcd.print("Ingrese Huella");
//lcd.clear();
//if (finger.verifyPassword()) {
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 15 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		


Paso 3.4. Ejecutamos los comandos de los resultados a obtener

```

voidloop() {
  getFingerprintIDez();
  if(finger.fingerID==ID_reg1)
  { lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Registro Encontrado");
  lcd.setCursor(0,1);
  lcd.print("Welcome");
  lcd.setCursor(0,2);
  lcd.print("R1");
}

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 16 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

CONTROL DE ACCESO CON TECLADO MATRICIAL

El procedimiento para realizar un control de acceso mediante uso de Arduino es reconocer los funcionamientos de que cada equipo, a que voltaje trabaja para no producir daños en ellos, y realizar la programación y conexiones de la manera correcta, para una mejor operatividad.

Paso 1: Verificar las entradas y salidas del Arduino.

Es fundamental reconocer que pin es de entrada y salida para empezar la programación en Arduino. En el Arduino Mega existen entradas y salidas analógicas (6) y digitales (16). Por un lado, las analógicas son utilizadas, generalmente, para dispositivos de entrada y nos permiten leer un rango de valores. En cambio, las digitales pueden tener a la entrada o a la salida 0V o 5V, esto nos indica que ese pin está en estado bajo (Low) o alto (High).

Paso 2: Conexiones de los dispositivos.

Se debe verificar que entradas se va a utilizar, cablear con jumpers el teclado matricial tipo membrana hacia el Arduino, luego conectar como alarma de sonido un buzzer, para seguir con el uso de un solenoide que trabaja con una fuente de 12V y un módulo relé, el cual se activa al poner la clave de identificación el cual podrá brindar el acceso y la chapa eléctrica se abrirá.

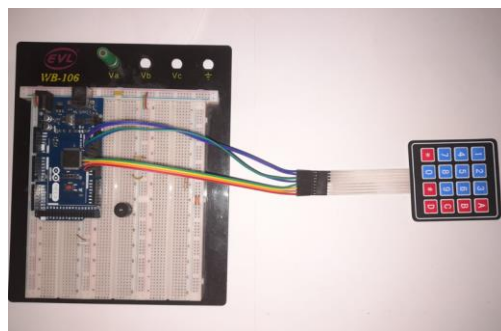



Figura 8. Conexión del teclado matricial con Arduino.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 17 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

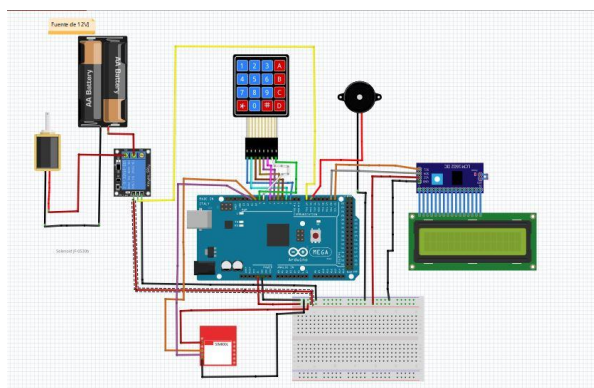


Figura 9. Conexión completa del control de acceso.

Paso 3: Programación


Después de las conexiones, Se debe realizar la programación mediante el uso del ambiente virtual de Arduino, para comenzar se debe descargar la librería “Keypad”, seguir con la codificación con uso de parámetros establecidos por parte de cliente como por ejemplos números de claves a usar, tiempo de espera para que no refleje error e intruso.

Paso 3.1. Iniciamos en incluir las librerías a usar.

Se inicia la programación exportando las librerías de los dispositivos a usar, para que se pueda efectuar el proceso.

```
#include <LiquidCrystal_I2C.h> //Incluir librería de dispositivos
a usar
#include <Wire.h>
#include <Keypad.h>
#include <SoftwareSerial.h>
LiquidCrystal_I2C lcd(0x27, 20,4);
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 18 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3.2. Definir entradas a utilizar

```

int _timeout;

String _buffer;

const byte ROWS = 4;

const byte COLS = 4;

charkeys[ROWS][COLS] = {

  {'1','2','3','A'},

  {'4','5','6','B'},

  {'7','8','9','C'},

  {'*','0','#','D'}};


charpin[5];

intposicion = 0;

byte rowPins[ROWS] = { 9,8,7,6 };

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 19 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3.3. Declarar variables para la ejecución del programa

```

voidsetup(){
Serial.begin(9600);
  _buffer.reserve(50);
sim.begin(9600);

lcd.init();

lcd.backlight();

lcd.setCursor(0,0);


lcd.print("Bienvenido ");

lcd.setCursor(0,1);

lcd.print("Introduce tu pin");

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 20 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		


Paso 3.4. Ejecutamos los comandos de los resultados a obtener

```

voidloop(){
pin[posicion] = teclado.getKey();
if (pin[posicion]){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Introduciendo");
for(int i = 0; i <= posicion; i++){
lcd.setCursor(i,1);
lcd.print("*");}
posicion++;
if(posicion == 5) posicion = 0;}
if(pin[4] == 'A'){
pin[4] = 'a';
if(pin[0] == '7' && pin[1] == '8' && pin[2] == '9' && pin[3] == 'c'&& pin[3] == '*'){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Pin correcto ");
lcd.setCursor(0,1);
lcd.print("Acceso Autorizado ");
digitalWrite(22,LOW);
delay(4000);
digitalWrite(22,HIGH);}
else {
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Pin incorrecto ");
lcd.setCursor(0,1);
lcd.print("Error, Acceso Denegado");
digitalWrite(43,HIGH);
delay(500);
digitalWrite(43,LOW);
delay(1500);
}
}
}

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 21 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```

else {

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Pin incorrecto ");

lcd.setCursor(0,1);

lcd.print("Error, Acceso Denegado");

digitalWrite(43,HIGH);


delay(500);

```

5) RECURSOS UTILIZADOS

- Cables Jumper M-M Y M-H
- Arduino Mega
- Arduino Uno
- Lector de huellas
- Tarjeta rfid
- Pantalla lcd
- Modulo Relé
- Teclado 4x4
- Protoboard
- Buzzer
- Solenoide de 12V

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 22 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

5) RESULTADOS OBTENIDOS

5.1. CONTROL DE ACCESO CON TARJETA RFID

Mediante esta práctica se obtiene como resultado principal el uso y manejo de dispositivos como seguridad, que permiten usar diferentes parámetros de acuerdo a la necesidad del área que lo requiera, mediante la implementación de acceso por tarjeta RFID.

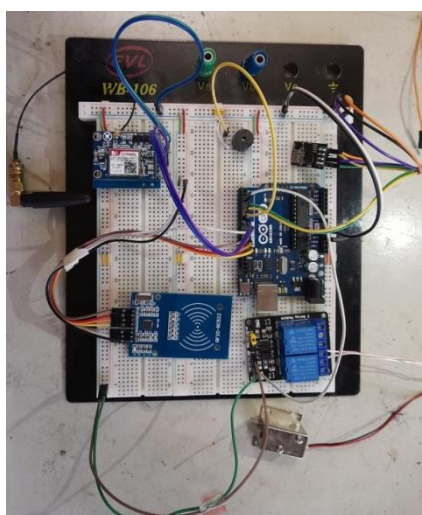



Figura 10. Conexión completa.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 23 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

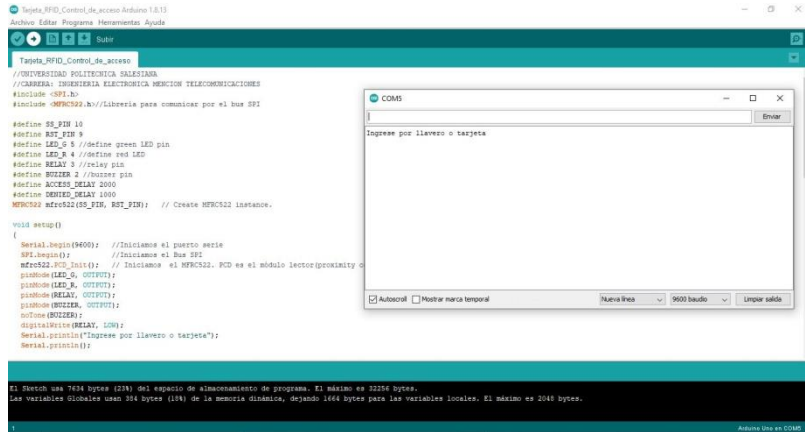


Figura 11. Inicio de programa.

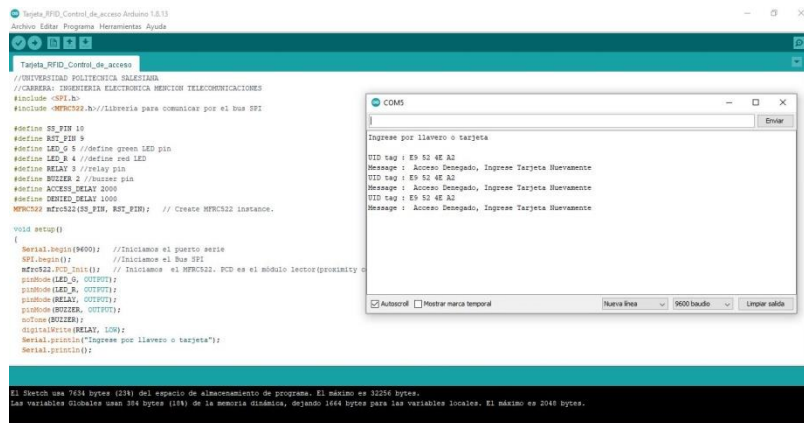



Figura 12. Error de acceso.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 24 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

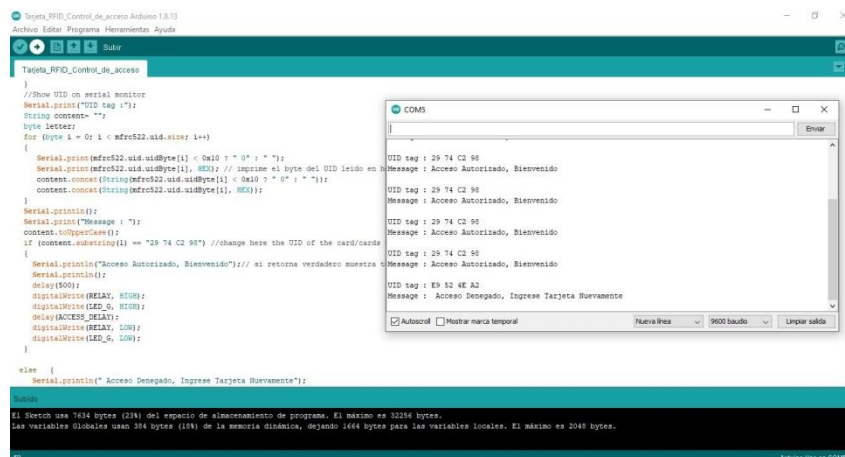


Figura 13. Acceso Autorizado

5.2. CONTROL DE ACCESO CON LECTOR DE HUELLAS

Mediante esta práctica se obtiene como resultado principal el uso y manejo de dispositivos como seguridad, que permiten usar diferentes parámetros de acuerdo con la necesidad del área que lo requiera, teniendo dos tipos de alarmas mediante mensaje de texto y correo electrónico.

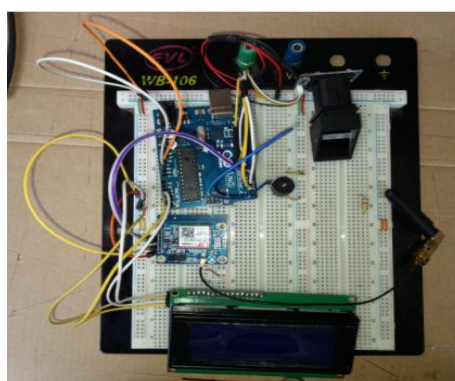


Figura 14. Inicio de programa.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:


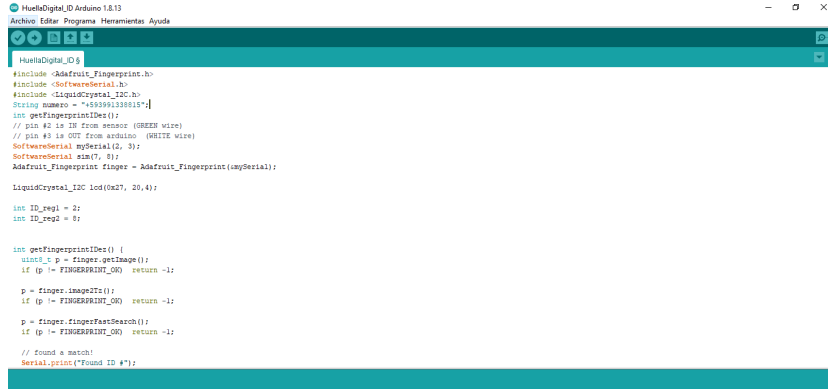
		REVISION 1/1	Página 25 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Figura 15.
Inicio de programa.



```

HuellDigital_ID Arduino 1.8.13
Archivo Editor Programa Herramientas Ayuda

HuellDigital_ID_0
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
String numero = "45899133015";
int getFingerprintIDe();
// pin #5 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint fmgp = Adafruit_Fingerprint(mySerial);
LiquidCrystal_I2C lcd(0x27, 20, 4);

int ID_req1 = 2;
int ID_req2 = 8;

int getFingerprintIDe() {
  uint8_t p = fmgp.getImage();
  if (p != FINGERPRINT_OK) return -1;

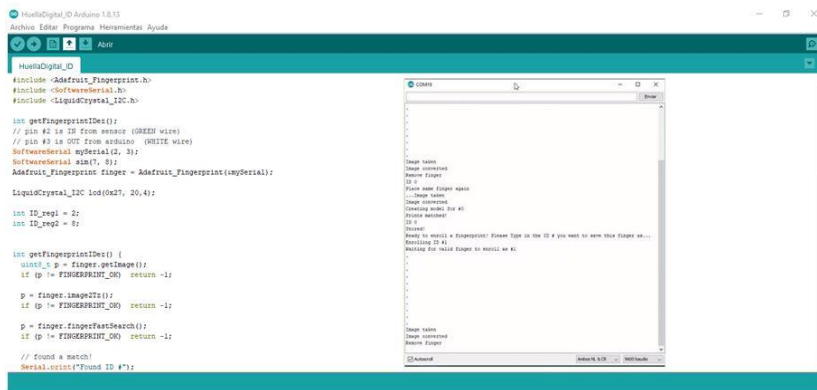
  p = fmgp.image2Tz();
  if (p != FINGERPRINT_OK) return -1;

  p = fmgp.fingerFastSearch();
  if (p != FINGERPRINT_OK) return -1;

  // found a match!
  Serial.println("Found ID #");
}

```

Inicio de



```

HuellDigital_ID Arduino 1.8.13
Archivo Editor Programa Herramientas Ayuda

HuellDigital_ID
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>

int getFingerprintIDe();
// pin #5 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint fmgp = Adafruit_Fingerprint(mySerial);
LiquidCrystal_I2C lcd(0x27, 20, 4);

int ID_req1 = 2;
int ID_req2 = 8;

int getFingerprintIDe() {
  uint8_t p = fmgp.getImage();
  if (p != FINGERPRINT_OK) return -1;

  p = fmgp.image2Tz();
  if (p != FINGERPRINT_OK) return -1;

  p = fmgp.fingerFastSearch();
  if (p != FINGERPRINT_OK) return -1;

  // found a match!
  Serial.println("Found ID #");
}

```


```

Change value
Change orientation
Place new finger
OK
Place new finger again
Change value
Change orientation
Detect a match for #0
Place number!
OK
Match
Ready to enroll a fingerprint? Please type in the ID # you want to save this finger as...
Enrolling ID #0
Waiting for this finger to enroll as #1
Change value
Change orientation
Place finger
OK
Match

```

Figura 16. Lectura de huella para acceso.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 26 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

5.3. CONTROL DE ACCESO CON TECLADO MATRICIAL

Mediante esta práctica se obtiene como resultado principal el uso y manejo de dispositivos como seguridad, que permiten usar diferentes parámetros de acuerdo con la necesidad del área que lo requiera, mediante la implementación de un acceso por clave.

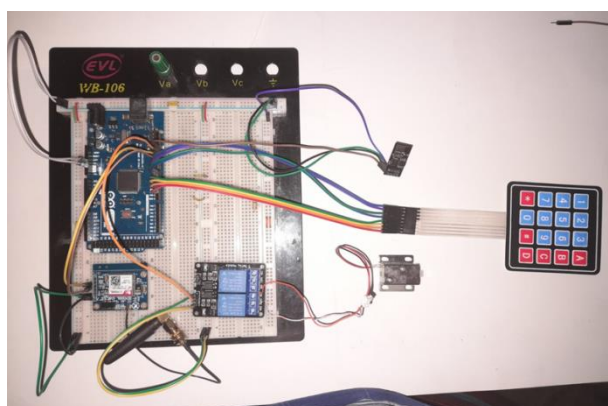


Figura 17. Conexión completa.



```

TESS2 Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

TESS2.g
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <Keypad.h>
#include <SoftwareSerial.h>
SoftwareSerial serial10(12);
LiquidCrystal_I2C lcd(0x27, 20, 4);

int _timeouts;
String _buffer;
String numero = "+593991330010";

const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','*'},
  {'4','5','6','#'},
  {'7','8','9','0'},
  {'*','0','#','*'}
};
char pin[5];
int position = 0;
byte rowPins[ROWS] = { 9,8,7,6 };
byte colPins[COLS] = { 5,4,3,2 };


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

void setup() {
  Serial.begin(9600);
  _buffer.reserve(50);

```

Figura 18. Inicio de programa.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 27 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

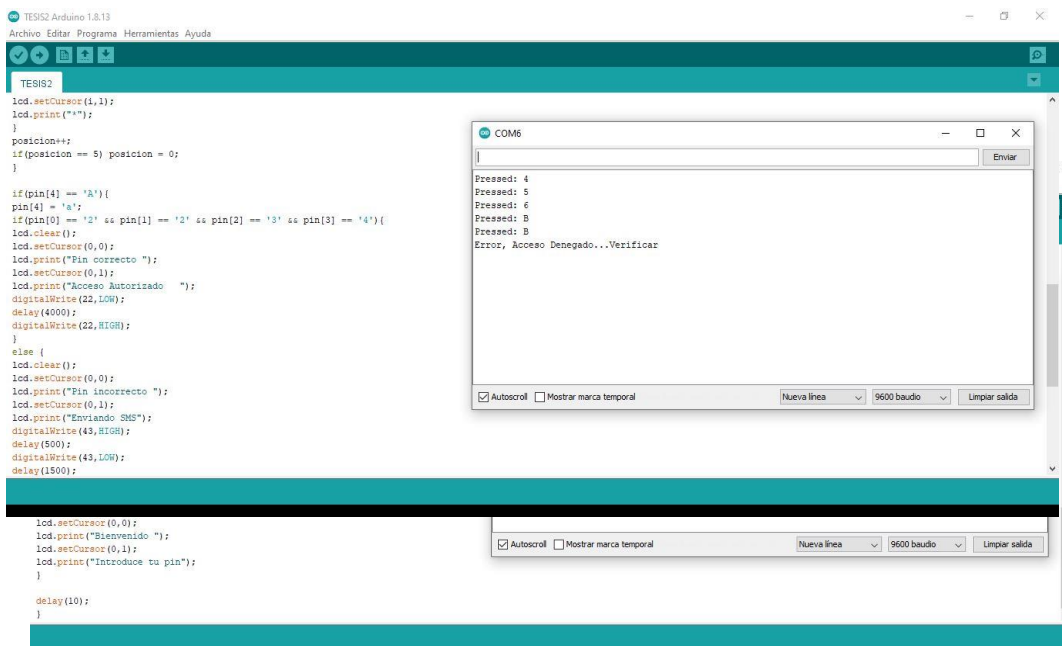



Figura 19. Error de clave.

Figura 20. Acceso Autorizado.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 28 de 28
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		


6) CONCLUSIONES

Con las prácticas realizadas, se busca que los usuarios tengan conocimientos de cómo funciona y se maneja un sistema de control de acceso en el área de seguridad electrónica, sin necesidad de una implementación costosa, y así puedan efectuar futuros proyectos basados en seguridad electrónica a pequeños precios.

7) BIBLIOGRAFÍA UTILIZADA

<https://www.arduino.cc/education/explore-iot-kit>
<https://www.arduino.cc/education/engineering-kit>
<https://search.arduino.cc/search?tab=&q=access+control>
<https://blog.arduino.cc/2018/05/16/an-arduino-based-rfid-tag-system-perfect-for-escape-rooms/?queryID=4df2f369cb6f9d98209488c32538b9c1>
<https://www.rinconingenieril.es/control-de-acceso-con-teclado/>
<https://programarfacil.com/blog/arduino-blog/lector-rfid-rc522-con-arduino/>
<https://www.instructables.com/Lector-De-Huella-Digital-Arduino/>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 1 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Anexo 5. Práctica desarrollada para la configuración y conexión de un sensor de humedad y temperatura.

ELECTRÓNICA Y TELECOMUNICACIONES

PRÁCTICA # 5


DOCENTE: ING. LENIN CEVALLOS

TIEMPO ESTIMADO 2 HORAS

NÚMERO DE ESTUDIANTES: 20

TEMA: “CONFIGURACIÓN Y CONEXIÓN DE UN SENSOR DE HUMEDAD Y TEMPERATURA.”

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 2 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

1) OBJETIVO GENERAL.

Aprender a conectar y configurar un sensor de temperatura y humedad DHT11/22 al microcontrolador Raspberry Pi como un sistema de seguridad electrónica.

2) OBJETIVOS ESPECÍFICOS.

- Identificar las herramientas necesarias para los trabajos de programación de una placa de Raspberry Pi.
- Realizar la programación para configurar al Raspberry Pi como microcontrolador para sistema de medición de temperatura y humedad.
- Realizar la correcta conexión entre el módulo DHT11 y el raspberry Pi mediante Conectores Jumper para establecer un sistema medición y control de temperatura.
- Realizar la ejecución y verificación de funcionamiento del sistema de medición de temperatura y humedad.

3) MARCO TEÓRICO.


RASPBERRY PI 4 MODEL B

Es una placa del tamaño de una tarjeta de crédito considerado como un microcontrolador que soporta varios componentes necesarios en un computador de escritorio Común. Posee puertos de entrada y salida que permite conectar el microcontrolador a medios físicos externos tales como dispositivos de teclado, ratones, pantallas, cámaras, sensores y actuadores.

Especificaciones:

- Procesador central.
- Procesador Gráfico (GPU).
- Módulo de memoria RAM.
- Conector RJ45.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

- 2 buses USB 2.0.
- 2 buses USB 3.0.
- Wifi y Bluetooth integrado.
- Interfaz para cámara y pantalla táctil.
- Salida Analógica de audio/estéreo por Jack de 3,5 mm.
- Salida digital de video MicroHDMI y audio.
- Pines de entrada y salida de propósito General (GPIO).
- 2 puertos de conexión MicroHDMI.
- 1 conector de alimentación MicroUSB tipo C.
- Lector de Tarjeta SD.

SENSOR DE TEMPERATURA Y HUMEDAD

El dispositivo Sensor de temperatura y humedad, con salida de señal digital, de tamaño reducido y de una precisión tan confiable que demuestra que es el indicado para un sistema de seguridad electrónica, cuenta con tres pines que sobresalen de su SBC interna, los cuales son diseñados para su alimentación de voltaje +Vcc, tierra y salida de señal captada.

Este sensor digital lleva un microcontrolador para realizar el tratamiento de la señal y este compuesto de un sensor capacitivo para medir la humedad y de un termistor.

El sensor DHT tiene dos modelos de fabricación DHT11 Y DHT


22 se puede apreciar en la Figura 513.

PYTHON 3

Python es un lenguaje de programación con un amplio campo de aplicación con énfasis en la legibilidad y ponderación de palabras claves estándar en inglés con fácil manejo al escribir y leer.

Es un lenguaje interpretativo e interactivo con el usuario está orientado a objetos y ofrece estructuras de datos de alto nivel por un tipado dinámico y puntual.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 4 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Su principal plataforma IDE de visualización es Thonny, un entorno de desarrollo de Python3 y es capaz de presentar los programas de parte natural y de fácil comprensión para principiantes.

PIP3

Es el sistema de gestión de paquetes que es el permite la administración e instalación de software escritos en Python.

VNC

Es una aplicación de procedimientos libre, su arquitectura es cliente – servidor lo cual nos permite visualizar la información del ordenador servidor remotamente a través de un dispositivo cliente.


BUZZER

Es un dispositivo capaz de convertir las señales eléctricas en sonido, tiene dos cables de conexión que sobresalen de su gabinete de fabricación, estos cables representan la polaridad del dispositivo los cuales son GND y VCC que se conectan a cualquier microcontrolador ya sea Raspberry o Arduino.

LUCES LED

Es un elemento capaz de absorber una corriente eléctrica moderada y convertirla en radiación electromagnética transformada en luz visible.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 5 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Accesorios a usar:

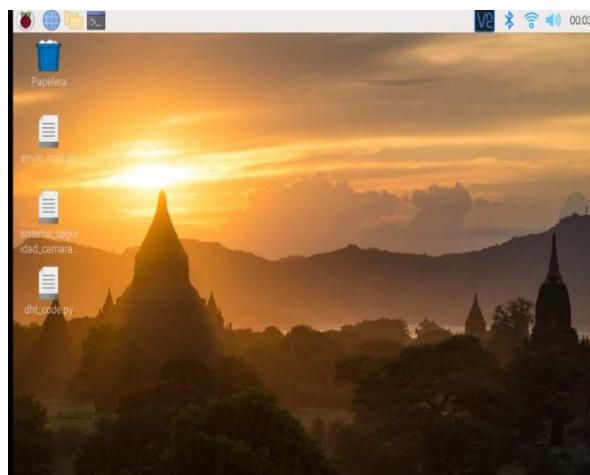
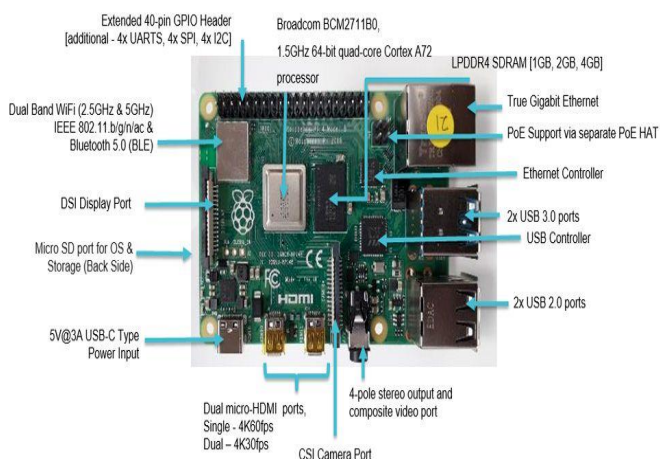
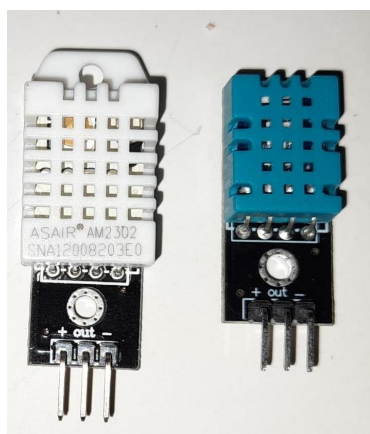


Figura 1. Raspberry Pi model 4B

Figura 2. Entorno virtual Raspberry Pi OS



Buzzer.

Figura 3. Sensor DHT11.

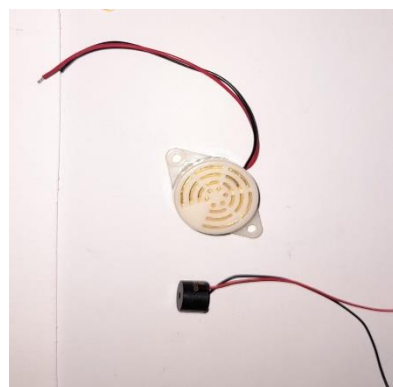



Figura 4.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 6 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

4) MARCO PROCEDIMENTAL

PROCEDIMIENTO

El procedimiento para realizar un sistema de medición de temperatura y humedad en una placa Raspberry Pi modelo 4B es mediante comandos programables en la pestaña de Terminal y Thonny Python, para esto necesitamos un módulo sensorial que sea compatible con el Raspberry Pi, en el proyecto será el sensor DHT11 la cual ira tomando lecturas del ambiente mediante su sensor capacitivo para medir la humedad y de un termistor para medir la temperatura.

La conexión entre estos dispositivos de entrada y procesamiento se realiza físicamente entre sus pines, el sensor DHT11 cuenta con tres pines de conexión y la placa Raspberry cuenta con sus entradas y salidas de propósito general GPIO.

RASPBERRY PI COMO SISTEMA DE DETECCIÓN DE MOVIMIENTO

Paso 1. Iniciamos Sesión en Raspberry Pi OS mediante VNC.


Para inicializar la práctica se debe energizar a la placa Raspberry Pi 4B con 5v y 3A, inmediatamente iniciamos sesión mediante nuestro protocolo VNC.

Recordandolas credenciales de acceso:

Dirección Ip del raspberry Pi: 192.168.39.167.
Usuario: pi
Contraseña: raspberrypi

(NOTA IMPORTANTE): Si no recuerda la dirección Ip del raspberry conecte la placa a un monitor mediante un cable HDMI y en la pestaña comandos realice la búsqueda mediante el código ifconfig.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 7 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

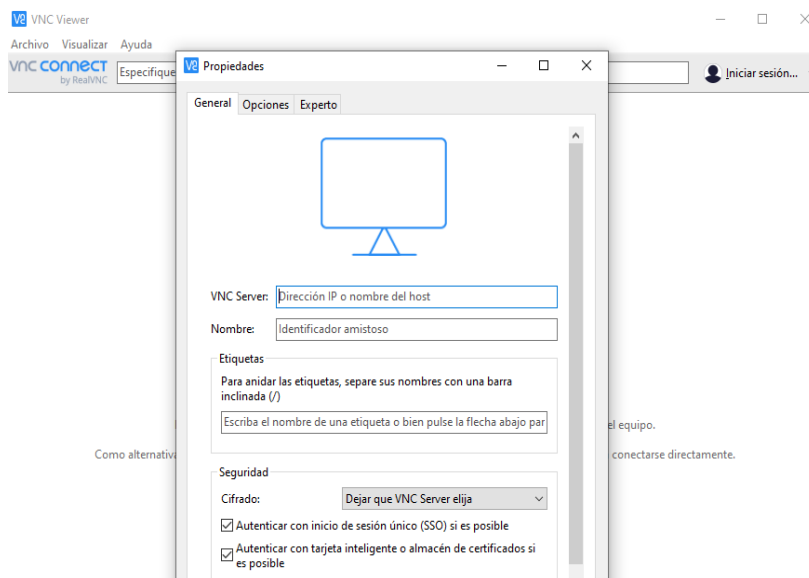


Figura 5. Conexión de acceso remoto VNC.

Paso 2. Actualización de paquetes Raspberry Pi.

Una vez iniciado la conexión con el ambiente virtual desde el protocolo “VNC”, se procede a iniciar la pestaña de Terminal.


Se recomienda actualizar los paquetes del sistema operativo con el siguiente comando:

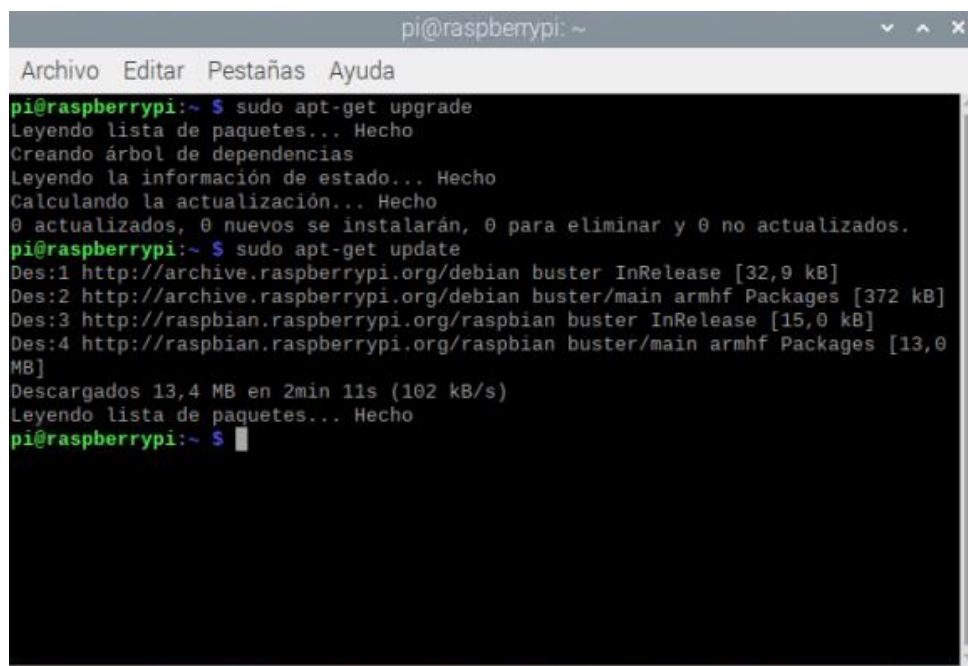
```
sudo apt-get update
```

Paso siguiente, se debe actualizar todos sus paquetes instalados a sus últimas versiones con el siguiente comando:

```
sudo apt-get upgrade
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 8 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
pi@raspberrypi:~ $ sudo apt-get update
Des:1 http://archive.raspberrypi.org/debian buster InRelease [32,9 kB]
Des:2 http://archive.raspberrypi.org/debian buster/main armhf Packages [372 kB]
Des:3 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Des:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0 MB]
Descargados 13,4 MB en 2min 11s (102 kB/s)
Leyendo lista de paquetes... Hecho
pi@raspberrypi:~ $

```

Figura 6. Actualización de paquetes.

Paso 3. Importación de librerías para activación de sensor DHT11 o DHT22.


Para continuar con la implementación de este sistema de medición se necesitará importar librerías de la biblioteca “CircuitPython_DHT” para la vinculación de los sensores ya sea DHT11 o DHT22 con la placa RaspberryPi.

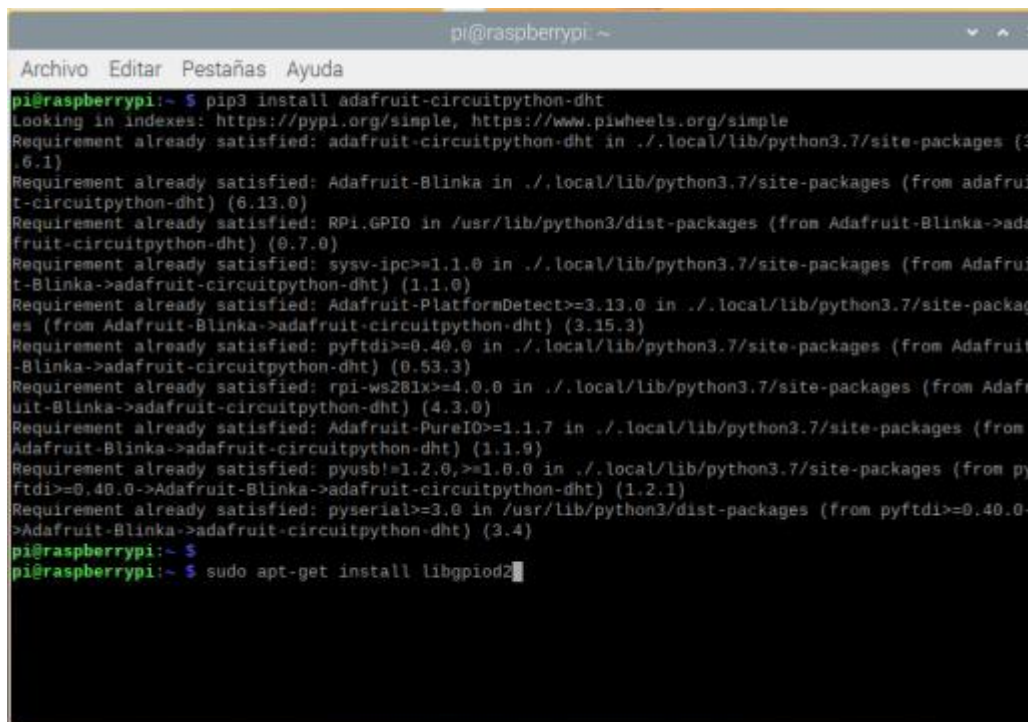
```

pip3 install adafruit-circuitpython-dht
sudo apt-get install libgpod2

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 9 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



```

pi@raspberrypi:~$ pip3 install adafruit-circuitpython-dht
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: adafruit-circuitpython-dht in ./local/lib/python3.7/site-packages (3.0.6.1)
Requirement already satisfied: Adafruit-Blinka in ./local/lib/python3.7/site-packages (from adafruit-circuitpython-dht) (6.13.0)
Requirement already satisfied: RPi.GPIO in /usr/lib/python3/dist-packages (from Adafruit-Blinka->adafruit-circuitpython-dht) (0.7.0)
Requirement already satisfied: sysv-ipc>=1.1.0 in ./local/lib/python3.7/site-packages (from Adafruit-Blinka->adafruit-circuitpython-dht) (1.1.0)
Requirement already satisfied: Adafruit-PlatformDetect>=3.13.0 in ./local/lib/python3.7/site-packages (from Adafruit-Blinka->adafruit-circuitpython-dht) (3.15.3)
Requirement already satisfied: pyftdi>=0.40.0 in ./local/lib/python3.7/site-packages (from Adafruit-Blinka->adafruit-circuitpython-dht) (0.53.3)
Requirement already satisfied: rpi-ws281x>=4.0.0 in ./local/lib/python3.7/site-packages (from Adafruit-Blinka->adafruit-circuitpython-dht) (4.3.0)
Requirement already satisfied: Adafruit-PureIO>=1.1.7 in ./local/lib/python3.7/site-packages (from Adafruit-Blinka->adafruit-circuitpython-dht) (1.1.9)
Requirement already satisfied: pyusb!>=1.2.0,>=1.0.0 in ./local/lib/python3.7/site-packages (from pyftdi>=0.40.0->Adafruit-Blinka->adafruit-circuitpython-dht) (1.2.1)
Requirement already satisfied: pyserial>=3.0 in /usr/lib/python3/dist-packages (from pyftdi>=0.40.0->Adafruit-Blinka->adafruit-circuitpython-dht) (3.4)
pi@raspberrypi:~$
pi@raspberrypi:~$ sudo apt-get install libgpiod2

```


Figura 7. importación de biblioteca “CircuitPython_DHT”.

Paso 4. Creación de código ejecutable en formato (xxx.py) para activación de sensor DHT11/22.

Una vez actualizado los paquetes de microcontrolador Raspberry Pi, se debe crear un código ejecutable en un formato que una al sensor DHT11 y al controlador mediante la combinación de conexiones físicas y presentación digital.

Para ello se debe inicializar la aplicación de programación virtual Thonny Python y escribir las siguientes líneas de códigos:

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 10 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

CÓDIGO:

```
import time
import board
import adafruit_dht
```

#Declaramos tiempo,

#eclaramos board o tablero, el cual es un módulo integrado a Circuitpython

#Importamos librería “Adafruit_dht” para los sensores DHT11


```
dhtDevice = adafruit_dht.DHT11(board.D4)
```

#Declaramos una variable que contenga la librería de adafruit y los datos tomados

```
while True:
    try:
        temperature_c = dhtDevice.temperature
        temperature_f = temperature_c * (9 / 5) + 32
        humidity = dhtDevice.humidity
        print(
            "Temp: {:.1f} F / {:.1f} C Humidity: {}% ".format(
                temperature_f, temperature_c, humidity
            )
        )
    )
```

#Creamos una condicional que lea las librerías específicas y muestre por pantalla la temperatura y humedad

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 11 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
except RuntimeError as error:
print(error.args[0])
time.sleep(2.0)
continue
except Exception as error:
dhtDevice.exit()
raise error
time.sleep(2.0)
```

#Creamos una excepción para cuando no tome medidas el sensor envíe un mensaje de error y una puerta secreta para salir del programa.

#El programa tomara lectura cada 2 segundos.

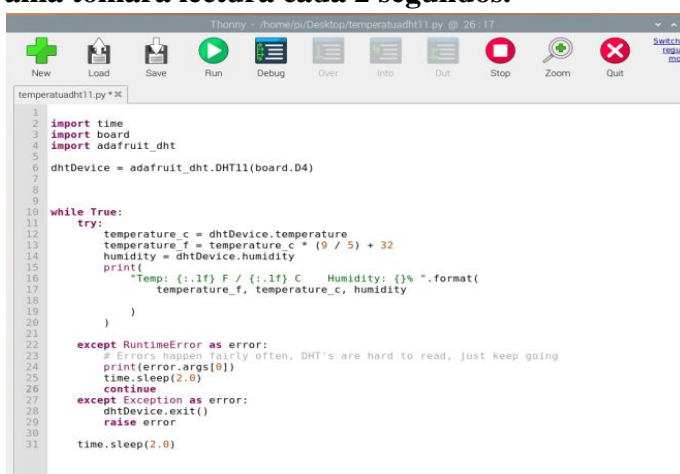



Figura 8. Código fuente de sistema de medición de temperatura y humedad.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 12 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 4. Instalación de sensor de movimiento en Raspberry Pi.

Para la conexión del sensor DHT11 y la Raspberry Pi, se debe utilizar cables jumper para comunicar los pines GPIO de la placa y los pines del sensor.

Para ello se debe identificar los pines del sensor, los cuales son tres, que sobresalen de su arquitectura de fabricación, los cuales son:

+VCC

OUT

GND

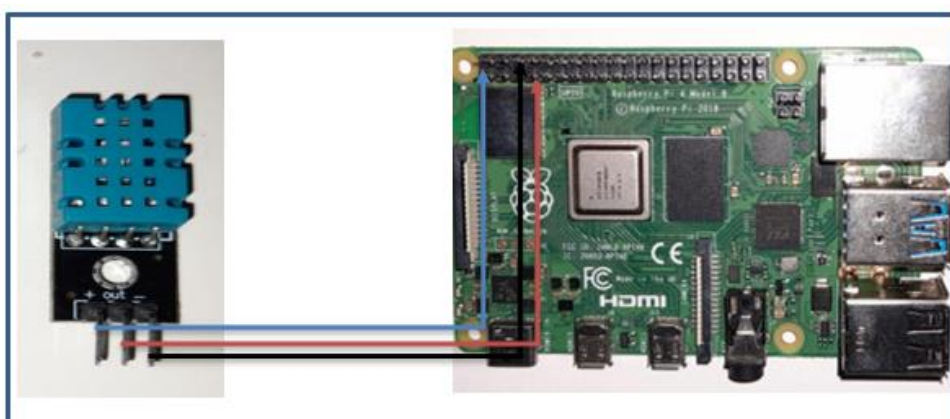



Figura 9. Pines de Sensor de temperatura y humedad DHT11.

Luego de identificar los pines del sensor DHT11 se podrá conectar a los Pines de la GPIO, para ello es necesario conocer su identificación ya que estos pueden ser Board (físicos) o BCM (digitales) según se determine en la programación.

Para ello se tendrá una tabla impresa en formato A4 para que el programador pueda acceder y realizar las conexiones sin temor a equivocarse.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 13 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Según el código ejecutable del paso 3, se define que los pines GPIO de la placa de Raspberry serán utilizados como entradas digitales (BCM). Por lo cual se debe conectar con la ayuda de un jumper la salida de la señal del sensor DHT (pin central) con la entrada digital 4.

Con la ayuda de la tabla impresa se podrá identificar los pines del GPIO de la placa de raspberry las cuales funcionan como VCC y GND.

Se debe recordar que el Sensor DHT dependiendo del modelo seleccionado trabaja con voltaje desde 3.3 hasta 5V el modelo DHT11 y de 4 hasta 6.5V el modelo DHT22.

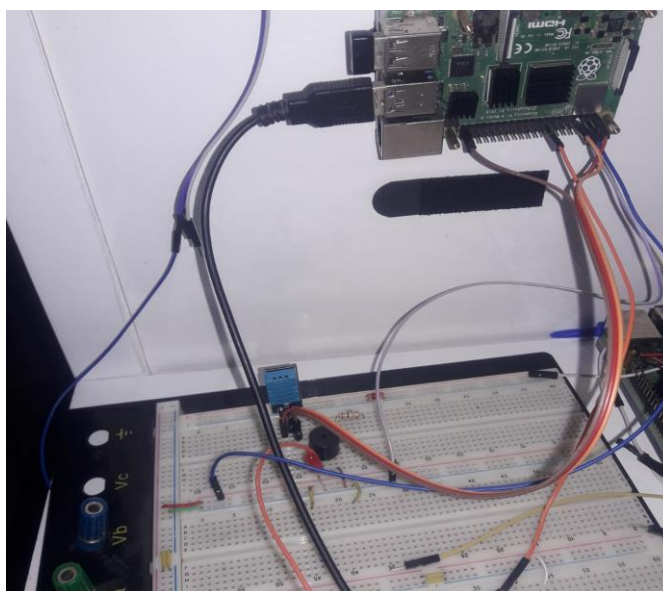



Figura 10. Conexión de Sensor DHT11 en una placa de pruebas y Raspberry Pi.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 14 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 5. Ejecución y visualización de resultados en pantalla.

Luego de realizar las conexiones físicas y crear el código ejecutable en el ambiente virtual de la Raspberry Pi, se debe guardar el archivo y ejecutarlo.


Para el ejemplo se ha guardado el código de ejecución como Temp.py el cual lo podemos ejecutar ingresando con la aplicación Thonny Python o por la línea de comandos.

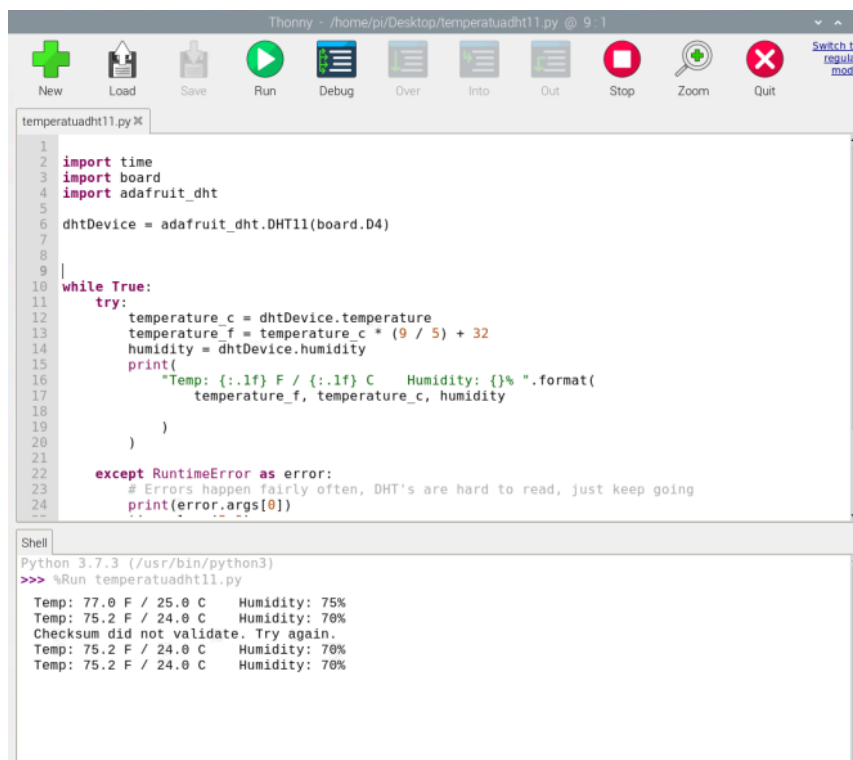
En el ambiente de programación Thonny Python solo debemos dar doble clic sobre el ícono verde de Play, inmediatamente deberá mostrarse el mensaje de Run en la ventana que se despliega en la parte inferior del monitor.

Como código inicial el sistema creado se activa y tomará medición de temperatura y humedad cada cierto rango de tiempo.

Para el segundo programa el código tomará lectura de la temperatura y humedad y en caso de no estar dentro de los parámetros establecidos enviara una señal al microcontrolador el cual hará que active la señal física del buzzer convirtiendo el uno lógico o señal eléctrica en sonido.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 15 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



```

1
2 import time
3 import board
4 import adafruit_dht
5
6 dhtDevice = adafruit_dht.DHT11(board.D4)
7
8
9
10 while True:
11     try:
12         temperature_c = dhtDevice.temperature
13         temperature_f = temperature_c * (9 / 5) + 32
14         humidity = dhtDevice.humidity
15         print(
16             "Temp: {:.1f} F / {:.1f} C Humidity: {}% ".format(
17                 temperature_f, temperature_c, humidity
18             )
19         )
20     except RuntimeError as error:
21         # Errors happen fairly often, DHT's are hard to read, just keep going
22         print(error.args[0])
23
24
--

```

```

Python 3.7.3 (/usr/bin/python3)
>>> %Run temperuadht11.py
Temp: 77.0 F / 25.0 C Humidity: 75%
Temp: 75.2 F / 24.0 C Humidity: 70%
Checksum did not validate. Try again.
Temp: 75.2 F / 24.0 C Humidity: 70%
Temp: 75.2 F / 24.0 C Humidity: 70%

```

Figura 11. Resultado del sistema de medición y control de temperatura leyendo señales del ambiente.

Para comparar las mediciones tomamos la temperatura que da el navegador de Google.

Y nos damos cuenta el sistema de medición si funciona correctamente.

En caso de que el sistema no quiera ejecutar dos veces debemos reiniciar la placa SBC de la raspberry Pi.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:


		REVISION 1/1	Página 16 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		




Figura 12. Datos de temperatura y humedad tomados de internet.

5) RECURSOS UTILIZADOS

- Kit Raspberry Pi.
- Conector alimentación 5V 3^a.
- Tarjeta MicroSD.
- Laptop.
- WI-FI.
- Protocolo VNC.
- Sensor DHT11
- Buzzer.
- Luces Led.
- Thonny Python.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 17 de 17
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

6) RESULTADOS OBTENIDOS

Como se ha demostrado en esta práctica, el funcionamiento de un sensor de temperatura y humedad es funcional para la toma de lectura de estos factores ambientales.

Se puede utilizar para crear una base de datos de los cambios del sector atmosférico dentro de un área específica y poder gracias a esto generar una tendencia y generar.

Enfocado netamente a la práctica, se demostró como comunica el sensor a un microcontrolador y programarlo de tal forma que pueda activarse sin la necesidad de que un usuario se encuentre permanente supervisando.

7) CONCLUSIONES

En virtud de lo estudiado, ahora se sabe cómo crear un sistema de medición y control de temperatura y humedad. La lectura de estos datos ambientales es utilizada en la seguridad electrónica cuando está orientada a control y mantenimiento, el poder conocer y programar un sistema que nos alerte cuando recepte medidas tomadas que estén fuera del rango establecido nos ayuda a evitar un daño mayor en los dispositivos que estemos vigilando.


8) BIBLIOGRAFÍA UTILIZADA

<https://projects.raspberrypi.org/en/projects/physical-computing/11>

<https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/python-setup>

<https://www.internetdelascosas.cl/2017/05/19/raspberry-pi-conectando-un-sensor-de-temperatura-y-humedad-dht11/>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 1 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Anexo 6. Práctica realizada para la configuración y conexión de un sensor de movimiento mediante Raspberry Pi.

ELECTRÓNICA Y TELECOMUNICACIONES

PRÁCTICA # 6


DOCENTE: ING. LENIN CEVALLOS

TIEMPO ESTIMADO 2 HORAS

NÚMERO DE ESTUDIANTES: 20

TEMA: “CONFIGURACIÓN Y CONEXIÓN DE UN SENSOR DE MOVIMIENTO MEDIANTE RASPBERRY PI”

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 2 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

1) OBJETIVO GENERAL.

Aprender a conectar y configurar un sensor de movimiento PIR HC - SR501 al Raspberry Pi como un sistema de seguridad electrónica.

2) OBJETIVOS ESPECÍFICOS.

- Identificar las herramientas necesarias para los trabajos de programación de una placa de Raspberry Pi.
- Realizar la programación para configurar al Raspberry Pi como microcontrolador para sistema de detección de movimiento.
- Realizar la correcta conexión entre el dispositivo PIR HC - SR501 y el raspberry Pi mediante Conectores Jumper para establecer un sistema detección de movimiento.
- Realizar la ejecución y verificación de funcionamiento del sistema de detección de movimiento.

3) MARCO TEÓRICO.


RASPBERRY PI 4 MODEL B

Es una placa del tamaño de una tarjeta de crédito considerado como un microcontrolador que soporta varios componentes necesarios en un computador de escritorio Común. Posee puertos de entrada y salida que permite conectar el microcontrolador a medios físicos externos tales como dispositivos de teclado, ratones, pantallas, cámaras, sensores y actuadores.

Especificaciones:

- Procesador central.
- Procesador Gráfico (GPU).
- Módulo de memoria RAM.
- Conector RJ45.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

- 2 buses USB 2.0.
- 2 buses USB 3.0.
- Wifi y Bluetooth integrado.
- Interfaz para cámara y pantalla táctil.
- Salida Analógica de audio/estéreo por Jack de 3,5 mm.
- Salida digital de video MicroHDMI y audio.
- Pines de entrada y salida de propósito General (GPIO).
- 2 puertos de conexión MicroHDMI.
- 1 conector de alimentación MicroUSB tipo C.
- Lector de Tarjeta SD.

SENSOR DE MOVIMIENTO.

El sensor de movimiento (PIR) “Pasive Infra Red” modelo HC-SR501 es un sensor piro eléctrico capaz de medir los niveles de radiación infrarroja emitida por cada cuerpo y objeto en un rango determinado de Máximo 5 metros.

Es un sensor pasivo ideal para la detección de dentro de su franja de barrido.

Su rango de operacional es de 4,5V – 20V.


Tiene 3 pines, que se denominan VCC, OUTPUT y GND.

Tiene 2 potenciómetro para ajustar los parámetros de sensibilidad y retardo. El cual puede configurarse entre 5 y 300 segundos (retardo), y 2 a 5 metros (sensibilidad).

PYTHON 3

Python es un lenguaje de programación con un amplio campo de aplicación con énfasis en la legibilidad y ponderación de palabras claves estándar en inglés con fácil manejo al escribir y leer. Es un lenguaje interpretativo e interactivo con el usuario está orientado a objetos y ofrece estructuras de datos de alto nivel por un tipado dinámico y puntual.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 4 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Su principal plataforma IDE de visualización es Thonny, un entorno de desarrollo de Python3 y es capaz de presentar los programas de parte natural y de fácil comprensión para principiantes.

PIP3

Es el sistema de gestión de paquetes que es el permite la administración e instalación de software escritos en Python.

VNC

Es una aplicación de procedimientos libre, su arquitectura es cliente – servidor lo cual nos permite visualizar la información del ordenador servidor remotamente a través de un dispositivo cliente.


BUZZER

Es un dispositivo capaz de convertir las señales eléctricas en sonido, tiene dos cables de conexión que sobresalen de su gabinete de fabricación, estos cables representan la polaridad del dispositivo los cuales son GND y VCC que se conectan a cualquier microcontrolador ya sea Raspberry o Arduino.

LUCES LED

Es un elemento capaz de absorber una corriente eléctrica moderada y convertirla en radiación electromagnética transformada en luz visible.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 5 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Accesorios a usar:

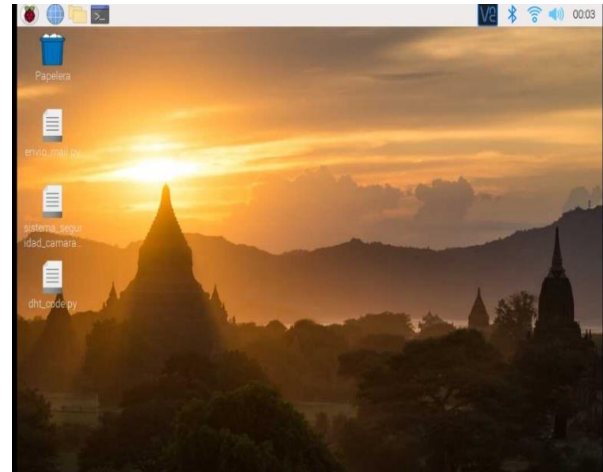
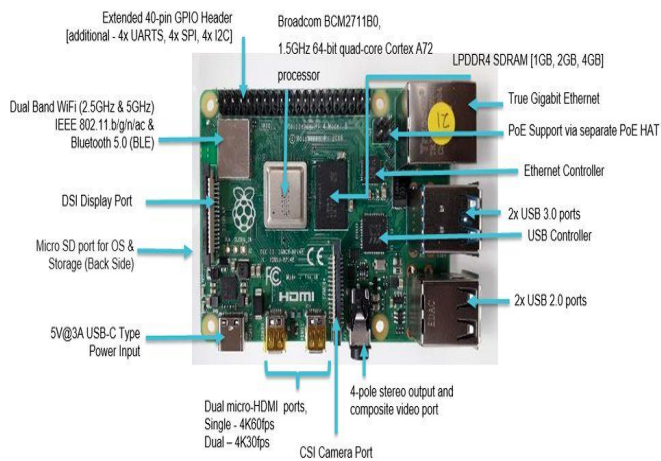


Figura 1. Raspberry Pi model 4B

Figura 2. Entorno virtual Raspberry Pi OS



Figura 3. Sensor PIR HC-SR501.

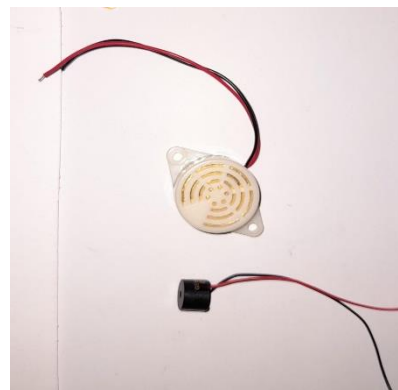



Figura 4. Buzzer.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 6 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

4) MARCO PROCEDIMENTAL

PROCEDIMIENTO

El procedimiento para realizar un sistema de detección de movimiento en una placa Raspberry Pi modelo 4B es mediante comandos programables en la pestaña de Terminal y Thonny Python, para esto necesitamos un sensor compatible con el Raspberry Pi, en el proyecto será el sensor PIR HC-SR501 la cual ira capturando señales infrarrojas que estén atravesando su rango radiación en un determinado tiempo.

La conexión entre estos dispositivos de entrada y procesamiento se realiza físicamente entre sus pines, el sensor PIR cuenta con tres pines de conexión y la placa Raspberry cuenta con sus entradas y salidas de propósito general GPIO.

RASPBERRY PI COMO SISTEMA DE DETECCIÓN DE MOVIMIENTO

Paso 1. Iniciamos Sesión en Raspberry Pi OS mediante VNC.

Para inicializar la práctica se debe energizar a la placa Raspberry Pi 4B con 5v y 3A, inmediatamente iniciamos sesión mediante nuestro protocolo VNC.


Recordandolas credenciales de acceso:

Dirección Ip del raspberry Pi: 192.168.39.167.
Usuario: pi

Contraseña: raspberrypi

(NOTA IMPORTANTE): Si no recuerda la dirección Ip del raspberry conecte la placa a un monitor mediante un cable HDMI y en la pestaña comandos realice la búsqueda mediante el código ifconfig.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 7 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

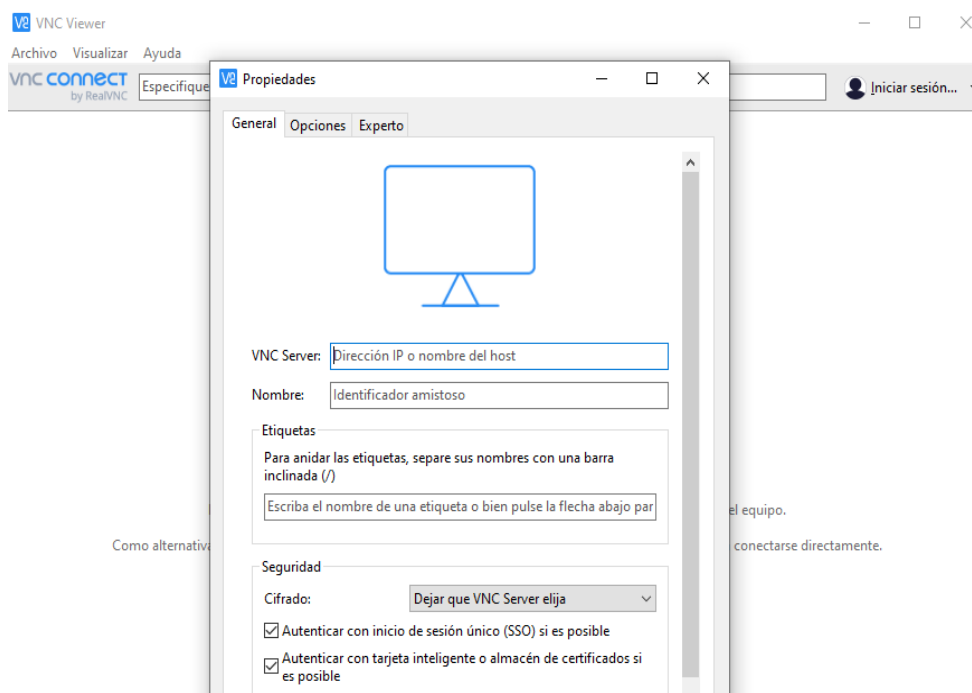


Figura 5. Conexión de acceso remoto VNC.

Paso 2. Actualización de paquetes Raspberry Pi.

Una vez iniciado la conexión con el ambiente virtual desde el protocolo “VNC”, se procede a iniciar la pestaña de Terminal.


Se recomienda actualizar los paquetes del sistema operativo con el siguiente comando:

```
sudo apt-get update
```

Paso siguiente, se debe actualizar todos sus paquetes instalados a sus últimas versiones con el siguiente comando:

```
sudo apt-get upgrade
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 8 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

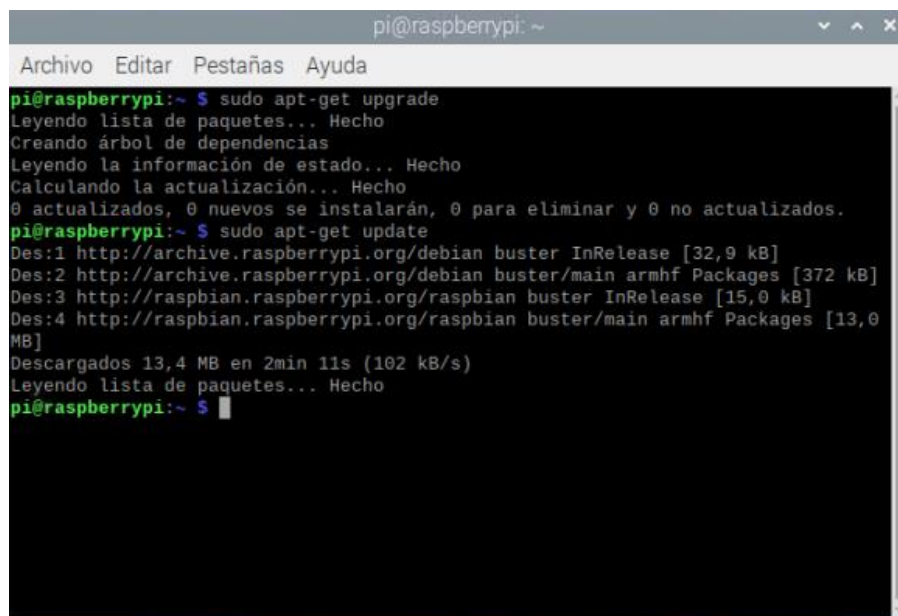


Figura 6. Actualización de paquetes.

Paso 3. Creación de código ejecutable en formato (xxx.py) para activación de sensor PIR.

Una vez actualizado los paquetes de microcontrolador Raspberry Pi, se debe crear un código ejecutable en un formato que una al sensor PIR y al controlador mediante la combinación de conexiones físicas y presentación digital.


Para ello se debe inicializar la aplicación de programación virtual Thonny Python y escribir las siguientes líneas de códigos:

CÓDIGO:

```
importRPi.GPIO as GPIO
import time
```

#importamos la librería para configuración de pines de la raspberry

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 9 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

#importamos la librería para el uso del módulo de tiempo

```
detector = 11
alarma = 7
```

#Declaramos dos variables independientes detector, alarma

```
GPIO.setmode(GPIO.BOARD)
```

#Declaramos los pines de GPIO como Board es decir entradas físicas no lógicas

```
GPIO.setup(alarma, GPIO.OUT)
GPIO.setup(detector, GPIO.IN)
```

#Declaramos la variable del sensor Pir (detector) como entrada en la GPIO

#Declaramos la variable del Buzzer (alarma) como salida en la GPIO

```
current_state = 0
```

#Declaramos un estado inicial valor lógico 0

```
print('Esperando a que se establezca el Sensor de Movimiento')
```


#Imprimimos por pantalla el establecimiento del sensor

```
time.sleep(3)
print('sensor encendido')
```

#Imprimimos por pantalla el sensor encendido

```
try:
while True:
time.sleep(0.2)
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 10 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
current_state = GPIO.input(detector)
```

#Declaramos a nuestra variable inicial como la entrada del GPIO detector

```
ifcurrent_state == 1:
print("alerta movimiento detectado señal del pin %s encendido igual a %s" %
(detector,current_state))
print("intruso detectado")
print("alarma encendida")
```

#Si el sensor detectado movimiento enviará una señal lógica 1 e imprimirá la alarma.

```
GPIO.output(alarma,True)
time.sleep(1)
```

#Detectado el movimiento por el sensor la raspberry activará la alarma del buzzer por 1 segundo


```
GPIO.output(alarma,False)
time.sleep(5)
```

#Luego de que la raspberry active el buzzer lo va a apagar por 5 segundos

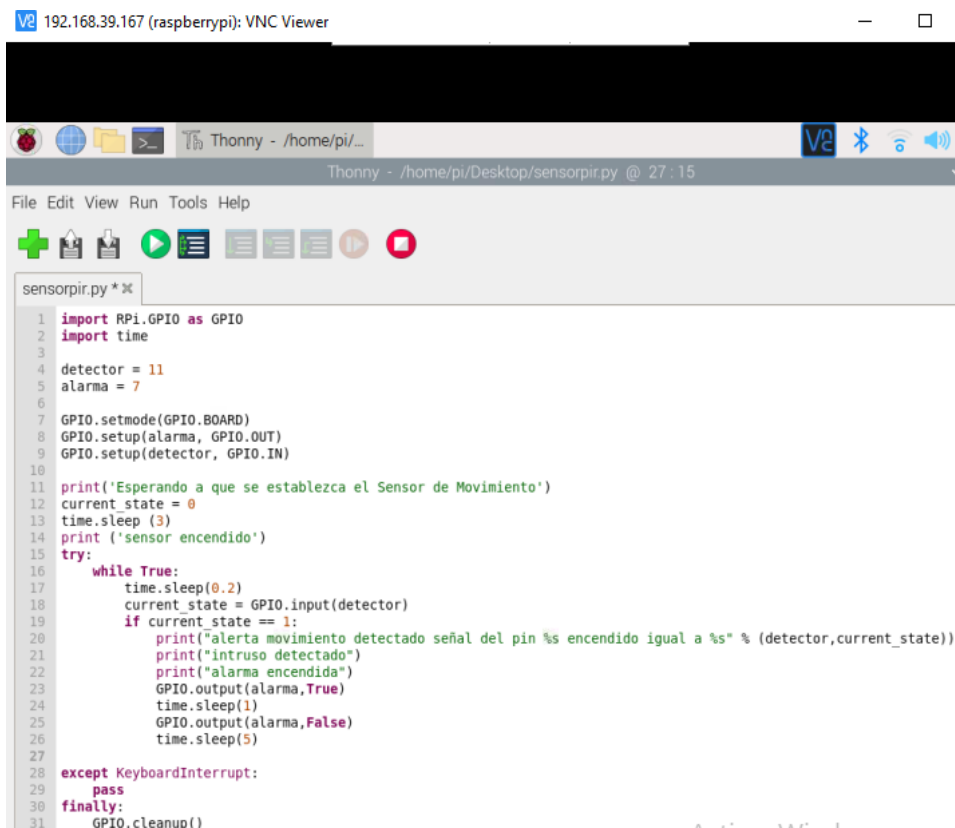
```
exceptKeyboardInterrupt:
pass
finally:
GPIO.cleanup()
```

#Declaramos un comando para detener la ejecución el cual es CTRL + C.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 11 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

#Finalmente limpiamos el while true.



```

1 import RPi.GPIO as GPIO
2 import time
3
4 detector = 11
5 alarma = 7
6
7 GPIO.setmode(GPIO.BOARD)
8 GPIO.setup(alarma, GPIO.OUT)
9 GPIO.setup(detector, GPIO.IN)
10
11 print('Esperando a que se establezca el Sensor de Movimiento')
12 current_state = 0
13 time.sleep(3)
14 print('sensor encendido')
15 try:
16     while True:
17         time.sleep(0.2)
18         current_state = GPIO.input(detector)
19         if current_state == 1:
20             print("alerta movimiento detectado señal del pin %s encendido igual a %s" % (detector,current_state))
21             print("intruso detectado")
22             print("alarma encendida")
23             GPIO.output(alarma,True)
24             time.sleep(1)
25             GPIO.output(alarma,False)
26             time.sleep(5)
27
28 except KeyboardInterrupt:
29     pass
30 finally:
31     GPIO.cleanup()

```


Figura 7. Código fuente de sistema de detección de movimiento.

Paso 4. Instalación de sensor de movimiento en Raspberry Pi.

Para la conexión del sensor PIR y la Raspberry Pi, se utilizará cables jumper para comunicar los pines GPIO de la placa y los pines del sensor.

Para ello se debe identificar las entradas y salidas de un Sensor PIR, el cual cuenta con tres pines que sobresalen de su arquitectura de fabricación, los cuales son:

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 12 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

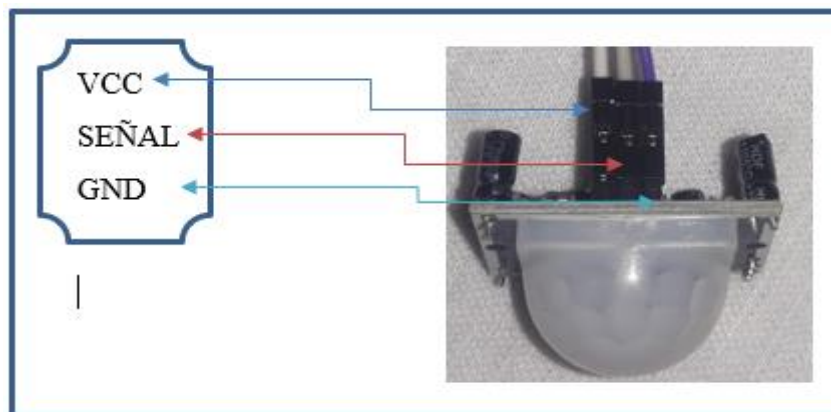


Figura 8. Pines de Sensor de movimiento PIR.

Luego de identificar los pines del sensor PIR se podrá conectar a los Pines de la GPIO, para ello es necesario conocer su identificación ya que estos pueden ser Board (físicos) o BCM (digitales) según se determine en la programación.


Para ello se tendrá una tabla impresa en formato A4 para que el programador pueda acceder y realizar las conexiones sin temor a equivocarse.

Según nuestro código del paso 3 declaramos que utilizaremos nuestros GPIO de la placa de Raspberry como entradas físicas (BOARD). Por lo cual se debe conectar con la ayuda de un jumper la salida de la señal del sensor PIR (pin central) con la entrada física 11 y el pin de encendido del buzzer con la entrada física 7. Al igual de los pines de señal del buzzer y del sensor, estos dispositivos deben ser conectados con voltaje moderado VCC y GND.

Con la ayuda de la tabla impresa se podrá identificar los pines del GPIO de la placa de raspberry las cuales funcionan como VCC y GND.

Se debe recordar que el Sensor PIR trabaja con voltaje desde 3.3 hasta 20 V.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 13 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

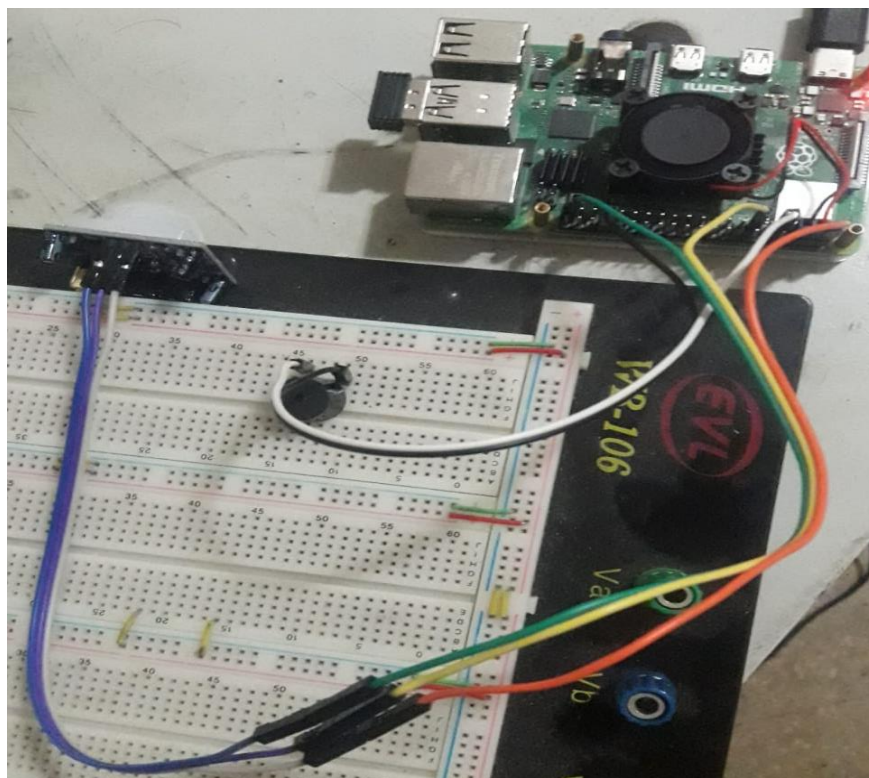


Figura 9. Conexión de Sensor PIR y Buzzer en una placa de pruebas y Raspberry Pi.


Paso 5. Ejecución y visualización de resultados en pantalla.

Luego de realizar las conexiones físicas y crear el código ejecutable en el ambiente virtual de la Raspberry Pi, se debe guardar el archivo y ejecutarlo.

Para el ejemplo se ha guardado el código de ejecución como `sensorpir.py` el cual lo podemos ejecutar ingresando con la aplicación Thonny Python o por la línea de comandos.

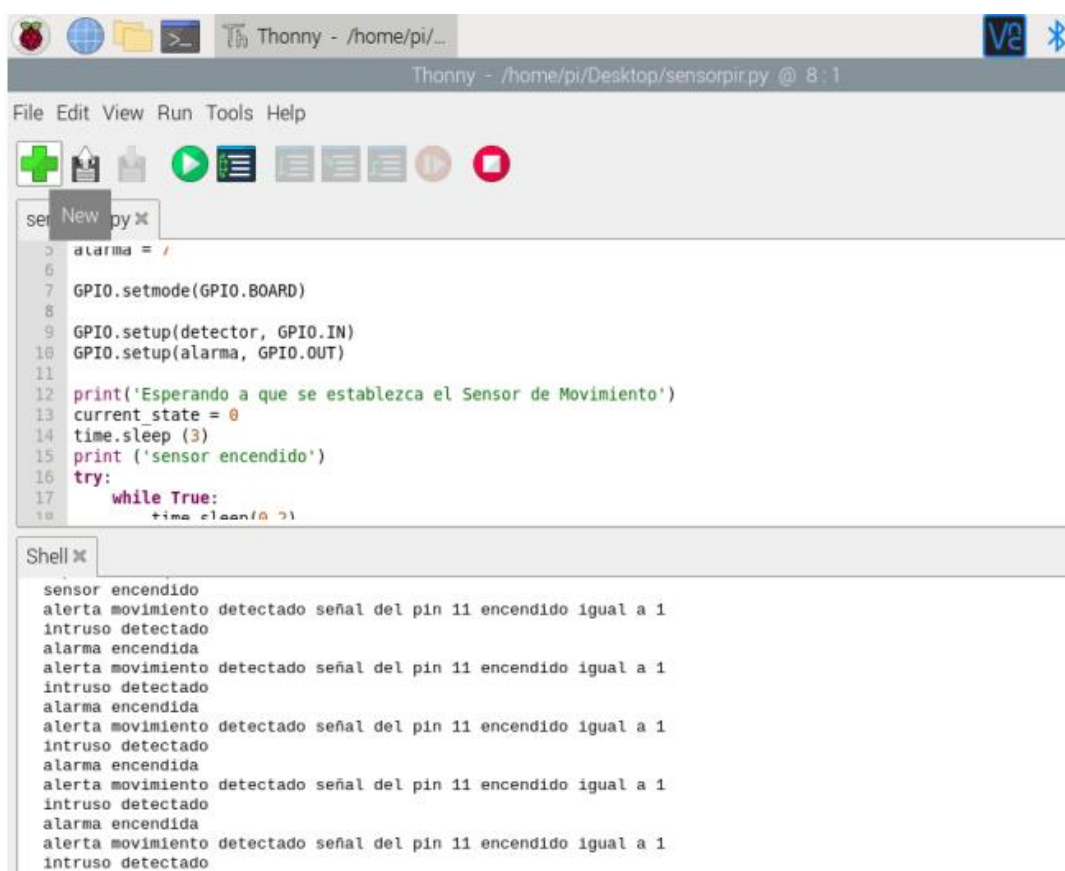
En el ambiente de programación Thonny Python solo debemos dar doble clic sobre el ícono verde de Play, inmediatamente deberá mostrarse el mensaje de Run en la ventana que se despliega en la parte inferior del monitor.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 14 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

El sistema creado se activa, cada vez que detecte movimiento o señal de infrarrojo en su rango de visión, el mismo hará que el microcontrolador active la señal física del buzzer convirtiendo el uno lógico o señal eléctrica en sonido.

El cual se activará por un tiempo estimado de 1 segundo y se apagará por 5 segundos antes de poder leer nueva señal de movimiento.



```

1 alarma = /
2
3
4
5
6
7 GPIO.setmode(GPIO.BOARD)
8
9 GPIO.setup(detector, GPIO.IN)
10 GPIO.setup(alarma, GPIO.OUT)
11
12 print('Esperando a que se establezca el Sensor de Movimiento')
13 current_state = 0
14 time.sleep(3)
15 print('sensor encendido')
16 try:
17     while True:
18         time.sleep(0.5)

```


```

Shell x
sensor encendido
alerta movimiento detectado señal del pin 11 encendido igual a 1
intruso detectado
alarma encendida
alerta movimiento detectado señal del pin 11 encendido igual a 1
intruso detectado
alarma encendida
alerta movimiento detectado señal del pin 11 encendido igual a 1
intruso detectado
alarma encendida
alerta movimiento detectado señal del pin 11 encendido igual a 1
intruso detectado
alarma encendida
alerta movimiento detectado señal del pin 11 encendido igual a 1
intruso detectado

```

Figura 10. Sistema de detección de movimiento leyendo señales de movimiento infrarrojo.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 15 de 15
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

5) RECURSOS UTILIZADOS

- Kit Raspberry Pi.
- Conector alimentación 5V 3A.
- Tarjeta MicroSD.
- Laptop.
- WI-FI.
- Protocolo VNC.
- Sensor PIR HC-SR501
- Buzzer.
- Luces Led.
- Thonny Python.

6) RESULTADOS OBTENIDOS

Mediante la implementación de esta práctica hemos comprendido el funcionamiento de un sensor de movimiento PIR y una de sus aplicaciones en el área de seguridad como lo es en alarmas físicas.

Comprendimos como comunicar el sensor a un microcontrolador y programarlo de tal forma que pueda activarse sin la necesidad de que un usuario se encuentre permanente supervisando.


7) CONCLUSIONES

La detección de movimiento o de señales infrarrojas es uno de los sistemas más utilizados en la seguridad electrónica, el poder controlar un sistema autónomo mediante un microcontrolador nos abre un abanico de posibilidades para la implementación de sistemas IOT.

1) BIBLIOGRAFÍA UTILIZADA

<https://projects.raspberrypi.org/en/projects/physical-computing/11>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 1 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Anexo 7. Práctica realizada para la configuración de detector de movimiento en CCTV mediante Raspberry Pi.

ELECTRÓNICA Y TELECOMUNICACIONES

PRÁCTICA # 7


DOCENTE: ING. LENIN CEVALLOS

TIEMPO ESTIMADO 2 HORAS

NÚMERO DE ESTUDIANTES: 20

TEMA: “CONFIGURACIÓN DE DETECTOR DE MOVIMIENTO EN CCTV MEDIANTE RASPBERRY PI”

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 2 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

1) OBJETIVO GENERAL.

Aprender a configurar un sensor de movimiento PIR HC - SR501 al Raspberry Pi añadiendo visualización mediante cámara HD como un sistema de seguridad electrónica con elementos pasivos y activos.

2) OBJETIVOS ESPECÍFICOS.

- Identificar las herramientas necesarias para los trabajos de programación de una placa de Raspberry Pi.
- Realizar la programación para configurar al raspberry Pi como microcontrolador para sistema de detección de movimiento.
- Añadir al sistema de detección de movimiento una cámara HD para capturar momento de alarma.
- Realizar la correcta conexión entre el dispositivo PIR HC - SR501, Cámara HD y el raspberry Pi mediante Conectores Jumper para establecer un sistema detección de movimiento.
- Realizar la ejecución y verificación de funcionamiento del sistema de detección de movimiento con captura de imagen.


3) MARCO TEÓRICO.

RASPBERRY PI 4 MODEL B

Es una placa del tamaño de una tarjeta de crédito considerado como un microcontrolador que soporta varios componentes necesarios en un computador de escritorio Común.

Posee puertos de entrada y salida que permite conectar el microcontrolador a medios físicos externos tales como dispositivos de teclado, ratones, pantallas, cámaras, sensores y actuadores.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Especificaciones:

- Procesador central.
- Procesador Gráfico (GPU).
- Módulo de memoria RAM.
- Conector RJ45.
- 2 buses USB 2.0.
- 2 buses USB 3.0.
- Wifi y Bluetooth integrado.
- Interfaz para cámara y pantalla táctil.
- Salida Analógica de audio/estéreo por Jack de 3,5 mm.
- Salida digital de video MicroHDMI y audio.
- Pines de entrada y salida de propósito General (GPIO).
- 2 puertos de conexión MicroHDMI.
- 1 conector de alimentación MicroUSB tipo C.
- Lector de Tarjeta SD.

SENSOR DE MOVIMIENTO.

El sensor de movimiento (PIR) “Pasive Infra Red” modelo HC-SR501 es un sensor piro eléctrico capaz de medir los niveles de radiación infrarroja emitida por cada cuerpo y objeto en un rango determinado de Máximo 5 metros.


Es un sensor pasivo ideal para la detección de dentro de su franja de barrido.

Su rango de operacional es de 4,5V – 20V.

Tiene 3 pines, que se denominan VCC, OUTPUT y GND.

Tiene 2 potenciómetro para ajustar los parámetros de sensibilidad y retardo. El cual puede configurarse entre 5 y 300 segundos (retardo), y 2 a 5 metros (sensibilidad).

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 4 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

CAMARA

La cámara deberá ser compatible con el microcontrolador Raspberry Pi como lo son las cámaras 5MPX OV5647 la cual cuenta con un buzz de dato de 15 cm para la conexión con la placa Raspberry mediante el puerto de cámara. La cámara OV5647 funciona con un voltaje bajo, su sensor de imagen CMOS de 5 megapíxeles nos brinda un rendimiento alto en resolución y nos proporciona una salida de video de 2592x1944 mediante la tecnología OmniBSITM. El núcleo del sensor OV5647 transmite datos de pixeles generales a una velocidad constante de fotogramas, indicadas por VSYNC y HREF.

PYTHON 3

Python es un lenguaje de programación con un amplio campo de aplicación con énfasis en la legibilidad y ponderación de palabras claves estándar en inglés con fácil manejo al escribir y leer. Es un lenguaje interpretativo e interactivo con el usuario está orientado a objetos y ofrece estructuras de datos de alto nivel por un tipado dinámico y puntual.

Su principal plataforma IDE de visualización es Thonny, un entorno de desarrollo de Python3 y es capaz de presentar los programas de parte natural y de fácil comprensión para principiantes.


PIP3

Es el sistema de gestión de paquetes que es el permite la administración e instalación de software escritos en Python.

VNC

Es una aplicación de procedimientos libre, su arquitectura es cliente – servidor lo cual nos permite visualizar la información del ordenador servidor remotamente a través de un dispositivo cliente.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 5 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

BUZZER

Es un dispositivo capaz de convertir las señales eléctricas en sonido, tiene dos cables de conexión que sobresalen de su gabinete de fabricación, estos cables representan la polaridad del dispositivo los cuales son GND y VCC que se conectan a cualquier microcontrolador ya sea Raspberry o Arduino

Accesorios a usar

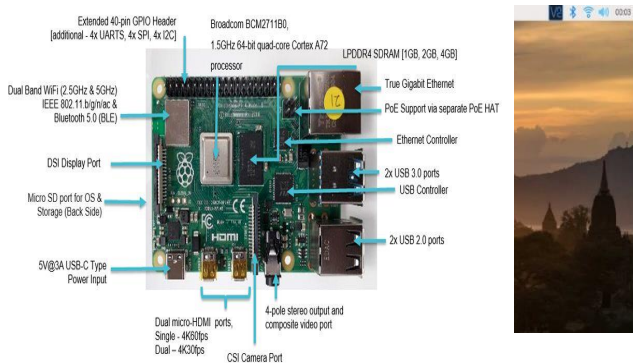


Figura 1. Raspberry Pi model 4B

Figura 2. Entorno virtual Raspberry Pi OS



Figura 3. Sensor PIR HC-SR501.

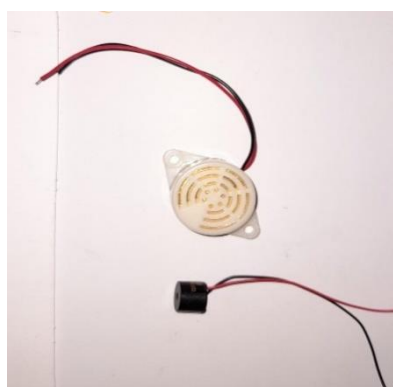



Figura 4. Buzzer.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 6 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

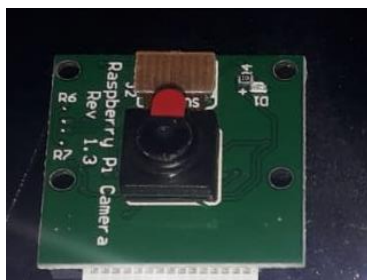


Figura 5. Cámara HD 5MPX OV564.

4) MARCO PROCEDIMENTAL


PROCEDIMIENTO

El procedimiento para realizar un sistema de detección de movimiento en una placa Raspberry Pi modelo 4B es mediante comandos programables en la pestaña de Terminal y Thonny Python, para esto se necesita un sensor compatible con el Raspberry Pi, en el proyecto será el sensor PIR HC-SR501 la cual ira capturando señales infrarrojas que estén atravesando su rango radiación en un determinado tiempo, una vez detectadas el sensor enviara un pulso positivo al microcontrolador quien ordenará a la cámara capturar el espacio que este siendo vulnerado, para que esto ocurra debe existir la codificación necesaria para que trabajen en regla de la cadena, adicional a la codificación, el sensor de movimiento y la cámara deberán estar alineadas de tal forma que su línea de vista esté direccionada hacia el mismo destino.

La conexión entre estos dispositivos de entrada y procesamiento se realiza físicamente entre sus pines, el sensor PIR cuenta con tres pines de conexión y la placa Raspberry cuenta con sus entradas y salidas de propósito general GPIO, la conexión con la cámara HD se realiza mediante la ranura para dispositivos de transmisión de imagen que tiene la placa Raspberry Pi.

Su configuración es mediante la utilización de Thonny Python o creación de un archivo ejecutable con formato (.py).

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 7 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

El código analizará continuamente el puerto de ingreso del sensor PIR en la entrada GPIO de la raspberry, cuando el sensor envíe un uno lógico a la placa, el microcontrolador procesará la orden y ejecutará el mensaje de alerta y activación de cámara y buzzer, indicando la ejecución del sistema de seguridad.

RASPBERRY PI COMO SISTEMA DE DETECCIÓN DE MOVIMIENTO E IMAGEN

Paso 1. Iniciamos Sesión en Raspberry Pi OS mediante VNC.

Para inicializar la práctica se debe energizar a la placa Raspberry Pi 4B con 5v y 3A, inmediatamente iniciamos sesión mediante nuestro protocolo VNC.

Recordandolas credenciales de acceso:

Dirección Ip del raspberry Pi: *192.168.39.167*.
 Usuario: pi
 Contraseña: raspberrypi

(NOTA IMPORTANTE): Si no recuerda la dirección Ip del raspberry conecte la placa a un monitor mediante un cable HDMI y en la pestaña comandos realice la búsqueda mediante el código ifconfig.

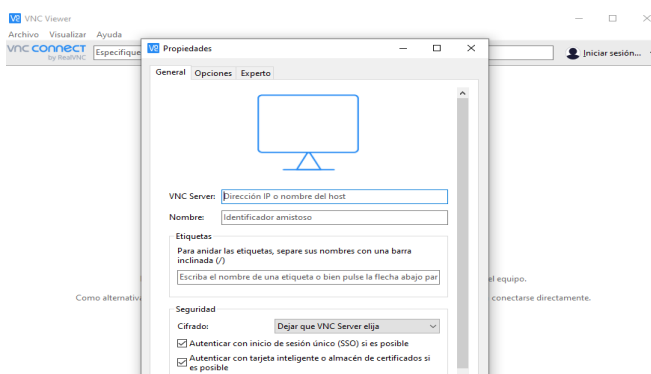



Figura 6. Conexión de acceso remoto VNC.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 8 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 2. Actualización de paquetes Raspberry Pi.

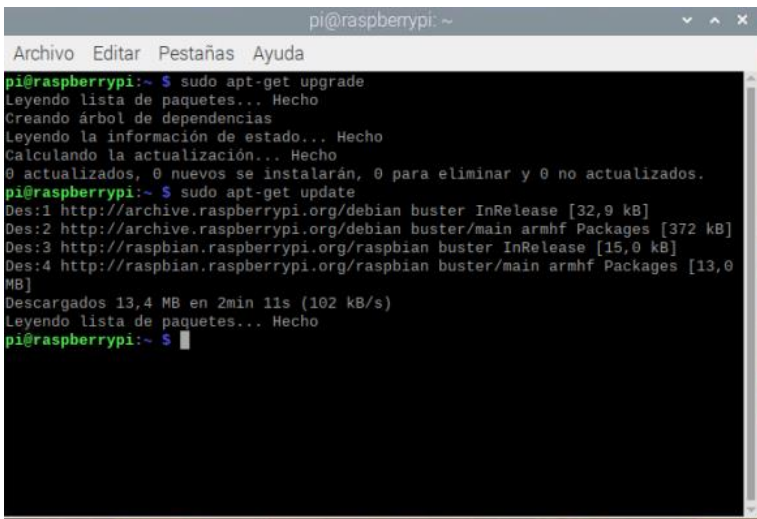
Una vez iniciado la conexión con el ambiente virtual desde el protocolo “VNC”, se procede a iniciar la pestaña de Terminal.

Se recomienda actualizar los paquetes del sistema operativo con el siguiente comando:

```
sudo apt-get update
```

Paso siguiente, se debe actualizar todos sus paquetes instalados a sus últimas versiones con el siguiente comando:

```
sudo apt-get upgrade
```



```


pi@raspberrypi:~ $ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
pi@raspberrypi:~ $ sudo apt-get update
Des:1 http://archive.raspberrypi.org/debian buster InRelease [32,9 kB]
Des:2 http://archive.raspberrypi.org/debian buster/main armhf Packages [372 kB]
Des:3 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Des:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0 MB]
Descargados 13,4 MB en 2min 11s (102 kB/s)
Leyendo lista de paquetes... Hecho
pi@raspberrypi:~ $

```

Figura 7. Actualización de paquetes.

Paso 3. Creación de código ejecutable en formato (.py) para activación de sensor PIR y cámara HD. Una vez actualizado los paquetes de microcontrolador Raspberry Pi, se debe crear un código ejecutable en un formato que una al sensor PIR a la cámara HD para Raspberry y al controlador mediante la combinación de conexiones físicas y presentación digital.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 9 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Gracias a que la cámara permite capturar imagen y grabar video se realizará dos códigos ejecutables que admitirá experimentar con estos dispositivos combinados.

Inicialmente se presenta la utilización del sensor y captura de imagen, posterior la utilización del sensor y grabado de video. Para iniciar la implementación se debe abrir la aplicación de programación virtual Thonny Python y escribir las siguientes líneas de códigos:

CÓDIGO #1:

```
from picamera import PiCamera
from time import sleep
import RPi.GPIO as GPIO
```

```
import time
```

#Importamos la librería de la cámara de raspberry pi

#Importamos librería de tiempo y de detección

#Importamos librería de GPIO para la configuración mediante código

```
GPIO.setmode(GPIO.BCM)
```

#Declaramos que serán BCM (digitales) los puertos de la GPIO

```
P=PiCamera()
P.resolution= (1024,756)
P.start_preview()
```

```
alarma = 24
```


#Declaramos P como la librería de cámara de Raspberry.

#Configuramos la resolución de la cámara.

#Declaramos alarma(buzzer) será la entrada lógica del GPIO24.

#Inicializamos la cámara.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 10 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
GPIO.setup(23, GPIO.IN)
GPIO.setup(24, GPIO.OUT)
```

#Declaramos la entrada GPIO 23 como entrada (sensor) y GPIO 24 como salida (buzzer).

```
try:
while True:
ifGPIO.input(23):
print("intruso detectado...")
GPIO.output(alarma,True)
time.sleep(1)
GPIO.output(alarma,False)
time.sleep(1)
#camera warm-up time
time.sleep(1)
P.capture(movimiento.jpg')
time.sleep(4)
```


#Creamos una condicional Try para las instrucciones de sensor y alertar

#Cuando la entrada del GPIO 23 reciba un uno lógico, el microcontrolador activara el buzzer por 1 segundo.

#La entrada GPIO 23 se activará si el sensor PIR capta un movimiento dentro de su rango de medición.

#Cuando el buzzer se active, secuencialmente el microcontrolador captura la imagen y la guardara en el escritorio.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 11 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()
```

#Declaramos un comando para detener la ejecución el cual es CTRL + C.

#Finalmente limpiamos el while true.

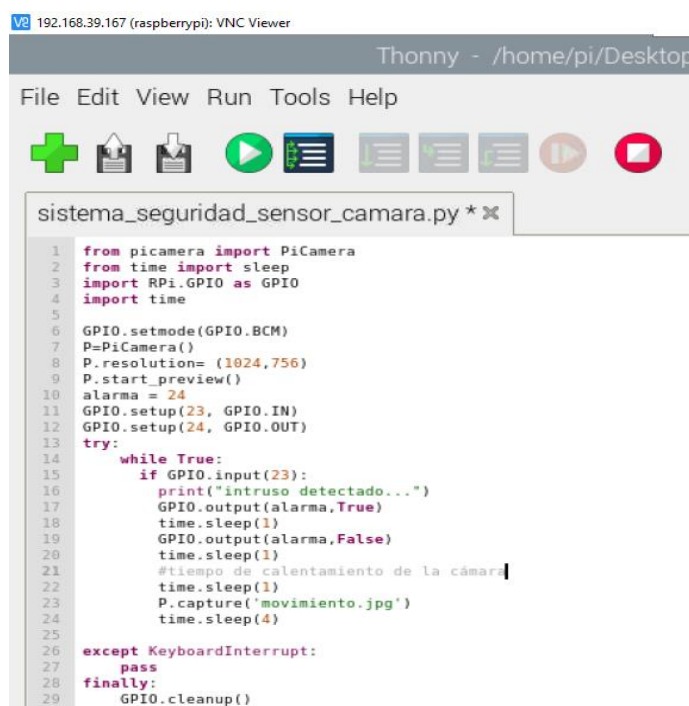



Figura 8. Código fuente de sistema de detección de movimiento y captura de imagen.

Luego de crear el código se deberá guardar en formato (xxx.py), para el código #1 se guardará con el nombre de sistema_seguridad_sensor_camara.py.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 12 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

CÓDIGO #2:

```
from picamera import PiCamera
from time import sleep
import RPi.GPIO as GPIO
import time
```

#Importamos la librería de la cámara de raspberry pi

#Importamos librería de tiempo y de detección

#Importamos librería de GPIO para la configuración mediante código

```
GPIO.setmode(GPIO.BCM)
```

#Declaramos que serán BCM (digitales) los puertos de la GPIO

```
P=PiCamera()
P.vflip = None
P.resolution = (800,600)
P.start_preview()
```

alarma = 24

#Declaramos P como la librería de cámara de Raspberry

#Configuramos para que el video salga sin giro invertido al proyectar Vflip=none


#Configuramos la resolución de la cámara

#Declaramos alarma (buzzer) será la entrada lógica del GPIO24

#Inicializamos la cámara

```
GPIO.setup(23, GPIO.IN)
GPIO.setup(24, GPIO.OUT)
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 13 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

#Declaramos la entrada GPIO 24 como entrada (sensor) y GPIO 24 como salida (buzzer).

try:

```
while True:
ifGPIO.input(23):
print("intruso detectado...")
GPIO.output(alarma,True)
time.sleep(1)
GPIO.output(alarma,False)
time.sleep(1)
    #Tiempo de calentamiento de la cámara
P.start_recording('output-video.h264')
P.wait_recording(10)
P.stop_recording()
P.stop_preview()
```

#Creamos una condicional Try para las instrucciones de sensor y alertar

#Cuando la entrada del GPIO 23 reciba un uno lógico, el microcontrolador activara el buzzer por 1 segundo.


#La entrada GPIO 23 se activará si el sensor PIR capta un movimiento dentro de su rango de medición.

#Cuando el buzzer se active, secuencialmente el microcontrolador grabará un video de 10 segundos del espacio donde se detectó el movimiento.

#Hay que recordar que la cámara de Raspberry Pi graba en formato h264.

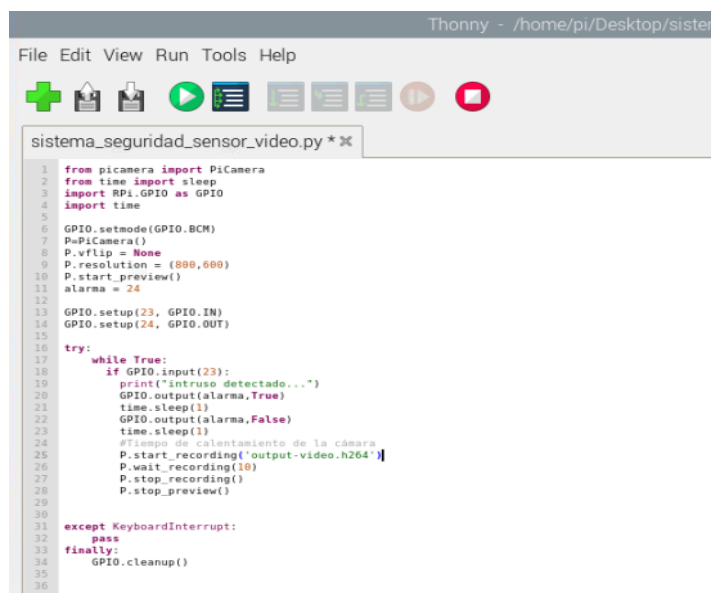
#Practica 3 recordamos como transformar de h264 a Mp4.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 14 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()
```

#Se declara un comando para detener la ejecución el cual es CTRL + C. #Finalmente limpiamos el while true.




```
Thonny - /home/pi/Desktop/sistem
File Edit View Run Tools Help
+ [Icons]
sistema_seguridad_sensor_video.py *x
1 from picamera import PiCamera
2 from time import sleep
3 import RPi.GPIO as GPIO
4 import time
5
6 GPIO.setmode(GPIO.BCM)
7 p=PiCamera()
8 P.vflip = None
9 P.resolution = (800,600)
10 P.start_preview()
11 alarma = 24
12
13 GPIO.setup(23, GPIO.IN)
14 GPIO.setup(24, GPIO.OUT)
15
16 try:
17     while True:
18         if GPIO.input(23):
19             print("intruso detectado...")
20             GPIO.output(alarma,True)
21             time.sleep(1)
22             GPIO.output(alarma,False)
23             time.sleep(1)
24             #Tiempo de calentamiento de la cámara
25             P.start_recording('output-video.h264')
26             P.wait_recording(10)
27             P.stop_recording()
28             P.stop_preview()
29
30 except KeyboardInterrupt:
31     pass
32 finally:
33     GPIO.cleanup()
34
35
36
```

Figura 9. Código fuente de sistema de detección de movimiento y grabación de video.

Luego de crear el código se deberá guardar en formato (xxx.py), para el código #1 se guardará con el nombre de sistema_seguridad_sensor_video.py.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 15 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 4. Instalación de sensor de movimiento en Raspberry Pi y cámara.

Para la conexión física del sensor de movimiento, el buzzer y la cámara HD se utilizará Jumper a la entrada GPIO de la Raspberry Py y la entrada CSI para cámaras.

Según el nuestro código se declara que las entradas que se utilizarán de las GPIO serán entradas BCM entradas digitales.

Las conexiones que se realizan del sensor PIR:

V_{cc} – GPIO 5V

GND - GPIO GROUND

SEÑAL – GPIO 23

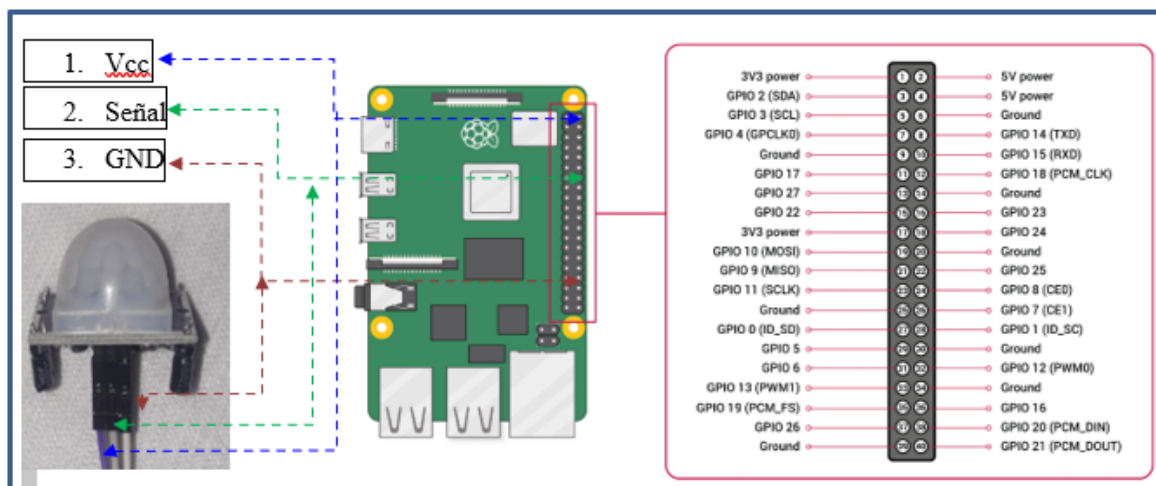



Figura 10. Conexiones físicas del sistema de detección de movimiento con cámara HD

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 16 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 5. Ejecución y visualización de resultados en pantalla.

Una vez realizadas las conexiones se debe ejecutar cada uno de los códigos con la botonera Run. Al ejecutar el código, el microcontrolador comienza a leer los pines de GPIO en búsqueda del uno lógico, cuando lo recibe muestra por pantalla el mensaje intruso detectado y ordena a la cámara de Raspberry a capturar o grabar mediante su lente y guardarlo en escritorio para su posterior revisión una vez.

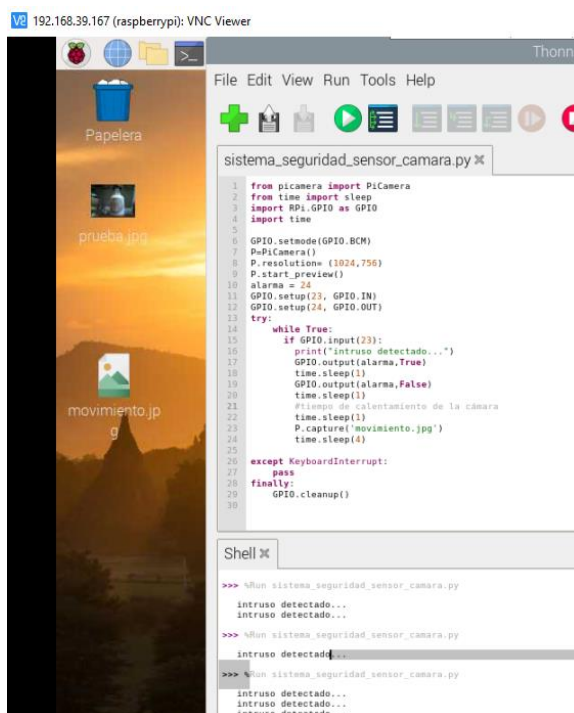



Figura 11. Sistema de detección de movimiento con captura de imagen.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 17 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

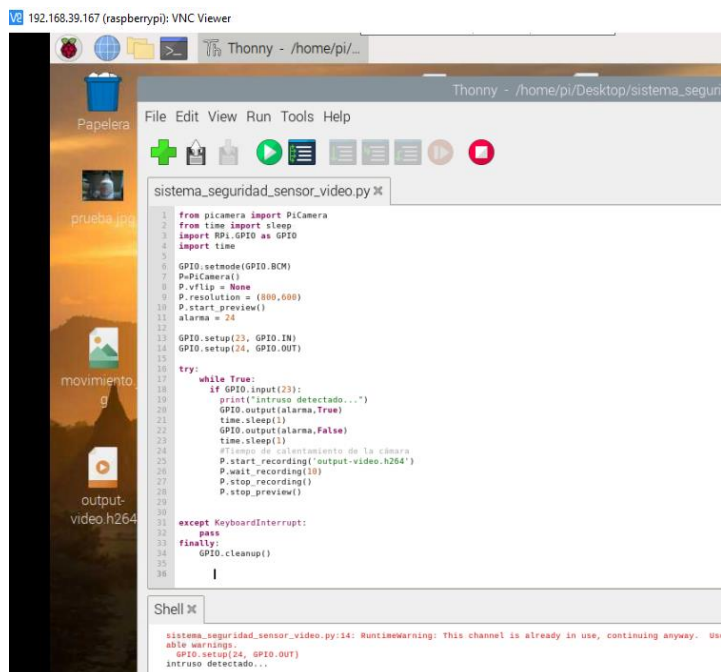



Figura 12. Sistema de detección de movimiento con grabación de video.

5) RECURSOS UTILIZADOS

- Kit Raspberry Pi.
- Conector alimentación 5V 3^a.
- Tarjeta MicroSD.
- Laptop.
- WI-FI.
- Protocolo VNC.
- Sensor PIR HC-SR501
- Buzzer.
- Thonny Python.
- Cámara HD

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 18 de 18
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

6) RESULTADOS OBTENIDOS

Mediante la implementación de esta práctica hemos comprendido el funcionamiento de un sensor de movimiento PIR vinculado a una cámara Raspberry Pi, y de esta forma crear el sistema de seguridad más completo detectando tanto señal de movimiento como imagen de vulnerabilidad. Gracias a la implementación de este sistema de seguridad comprendimos como comunicar el sensor PIR a un microcontrolador y programarlo de tal forma que pueda activarse sin la necesidad de que un usuario se encuentre permanente supervisando y poder guardar imagen o video capturado para tener una prueba física en caso de robo.

7) CONCLUSIONES

Un sistema de detección de movimiento complementado con una cámara espía nos dan la oportunidad de implementar una seguridad electrónica muy completa en el área donde deseamos obtener una protección especial tanto para registrar como para sensar.


La implementación de esta práctica es el punto de partida para la realización de IOT en la seguridad electrónica.

8) BIBLIOGRAFÍA UTILIZADA

<https://projects.raspberrypi.org/en/projects/physical-computing/11>

<https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/7>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 1 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Anexo 8. Práctica desarrollada para el control de acceso mediante uso de microcontrolador Arduino con módulo SIM800L GPRS/GSM para alertas mediante SMS.

ELECTRÓNICA Y TELECOMUNICACIONES

PRÁCTICA # 8


DOCENTE: ING. LENIN CEVALLOS

TIEMPO ESTIMADO 2 HORAS

NÚMERO DE ESTUDIANTES: 20

TEMA: “CONFIGURACIÓN Y PROGRAMACIÓN PARA FUNCIONAMIENTO DE UN MÓDULO GPRS/GSM COMO ALARMA DE TECNOLOGIA RFID, LECTOR DE HUELLA, TECLADO MATRICIAL EN ARDUINO”

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 2 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

1) OBJETIVO GENERAL.

Aprender a instalar y programar un control de acceso mediante el uso de un microcontrolador Arduino con módulo GPRS/GSM.

2) OBJETIVOS ESPECÍFICOS.

- Identificar las herramientas necesarias para las conexiones de un control de acceso
- Realizar programación mediante el uso de la plataforma de Arduino.
- Aprender sobre el uso de este sistema como seguridad de ingresos a áreas restringidas
- Aprender a programar módulo GPRS/GSM para aviso de error de ingresos.

3) MARCO TEÓRICO.


Arduino Uno

Arduino Uno es una placa basada en un microcontrolador, con código abierto orientado al microchip Atmega328p y desarrollado por arduino.cc. La placa contiene entradas/salidas digitales y analógicas, en total tiene 14 pines digitales. 6 pines analógicos y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B

Especificaciones:

- Microcontrolador: ATmega328
- Voltaje de operación: 5V
- Voltaje de entrada (recomendado): 7-12V
- Voltaje de entrada (límites): 6-20V
- Pines de E/S digitales: 14 (de los cuales 6 proporcionan salida PWM)
- Pines de entrada analógica: 6
- Corriente DC por pin de E/S: 40 mA
- Corriente DC para 3.3V Pin: 50 mA
- Memoria Flash: 32 KB de los cuales 0,5 KB utilizados por el bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Velocidad de reloj: 16 MHz

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		


Arduino Mega

Es una placa basada en un microcontrolador, tiene 54 entradas/salidas digitales, pueden ser usadas 15 como salidas PWM, 16 entradas analógicas, 4UARTs y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B.

Especificaciones:

- Microcontrolador: ATmega2560
- Voltaje de operación: 5V
- Voltaje de entrada (recomendado): 7-12V
- Voltaje de entrada (límites): 6-20V
- Pines de E/S digitales: 54 (de los cuales 15 proporcionan salida PWM)
- Pines de entrada analógica: 16
- Corriente DC por pin de E/S: 40 mA
- Corriente DC para 3.3V Pin: 50 mA
- Memoria Flash: 256 KB de los cuales 8 KB utilizados por el bootloader
- SRAM: 8 KB (ATmega2560)
- EEPROM: 4 KB (ATmega2560)
- Velocidad de reloj: 16 MHz
- Modulo Lector RC522.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 4 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

El módulo RC522 es Lector-Grabador RFID 13.56MHz, posee comunicación SPI lo que permite trabajar fácilmente con la mayoría de los microcontroladores. Utiliza un sistema de modulación y demodulación para todo tipo de dispositivos pasivos RFID de 13.56MHz. El dispositivo maneja el ISO14443-A y soporta el algoritmo de encriptación Quick CRYPTO1 y MIFARE. El rango de detección de tags RFID es de aprox. 5-7cm. Compatible con Arduino, Pic, Raspberry Pi y más. Los sistemas RFID son muy útiles para sistemas de control de acceso, seguridad electrónica, trazabilidad.

Tarjeta RFID

Es un sistema de almacenamiento y recuperación de datos remotos que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio.

Lector de huella dactilar

El Lector de huella dactilar es un sensor biométrico que me permite leer, guardar e identificar huellas de todos los dedos, este dispositivo contiene un conector para alimentación de 5V y comunicación serie.

LCD Display

Es un dispositivo con pantalla de cristal líquida, conformado por pixeles de color o monocromos, el cual permite mostrar información al usuario


Teclado Matricial

Es un dispositivo formado por una matriz de pulsadores, ordenados en filas y columnas para que se pueda reducir el número de pines a usar para la conexión.

Modulo Relé

Es un dispositivo electromecánico que permite a un procesador como Arduino controlar cargas a un nivel tensión o intensidad muy superior a las que su electrónica puede soportar

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 5 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Solenoides

Se trata básicamente de una cerradura electrónica, diseñada para un gabinete de base o de seguridad o de puerta. Normalmente, el bloqueo está activo por lo que no puede abrir la puerta porque el lingote del solenoide está en el camino (no utiliza ningún poder en este estado). Cuando se aplica 9 ~ 12 VDC, el lingote se contrae para que no sobresalga más y la puerta se pueda abrir.

Buzzer

Es dispositivo que produce sonido de un mismo tono, es usado como señalización o alarma y es usado en diversos sistemas de seguridad

Módulo GSM/GPRS sim800l

Es un dispositivo GSM/GPRS que trabaja en cuatro bandas 850, 900, 1800 y 1900 MHz , permite enviar y recibir mensajes de textos o llamadas, trabaja con un voltaje de 3.4 a 4.4 V

Accesorios para usar:

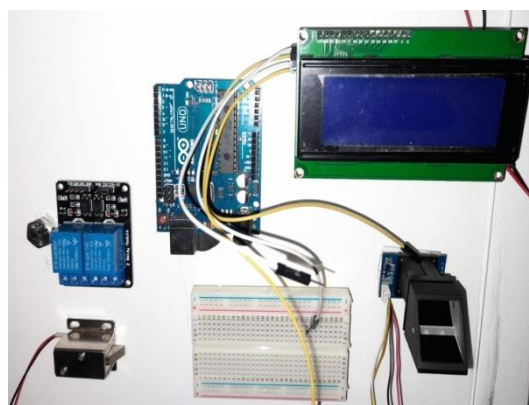



Figura 1. Kit de práctica con tarjeta RFID **Figura 2.** Kit de práctica con lector de huellas

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 6 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

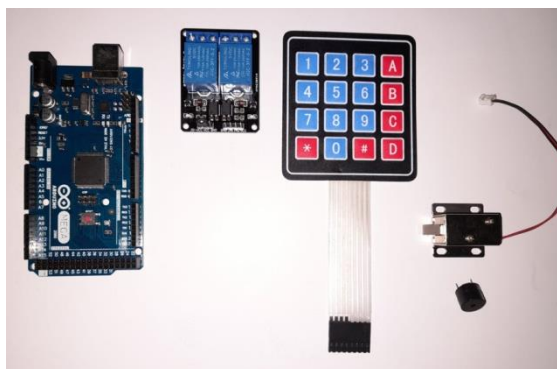


Figura 3. Kit de práctica con teclado matricial

4) MARCO PROCEDIMENTAL

4.1.CONTROL DE ACCESO CON TARJETA RFID

El procedimiento para realizar un control de acceso mediante uso de Arduino es reconocer los funcionamientos de que cada equipo, a que voltaje trabaja para no producir daños en ellos, y realizar la programación y conexiones de la manera correcta, para una mejor operatividad.


Paso 1: Verificar las entradas y salidas del Arduino.

Es fundamental conocerlas para empezar a programar Arduino. En el Arduino UNO existen entradas y salidas analógicas (6) y digitales (16). Por un lado, las analógicas son utilizadas, generalmente, para dispositivos de entrada y nos permiten leer un rango de valores. En cambio, las digitales pueden tener a la entrada o a la salida 0V o 5V, esto nos indica que ese pin está en estado bajo (Low) o alto (High).

Paso 2: Conexiones de los dispositivos.

Se debe verificar que entradas se va a utilizar, cablear con jumpers hembra-macho la tarjeta RFID hacia el Arduino, luego conectar como alarma de sonido y visión un buzzer y leds como indicadores, para seguir con el uso de un solenoide que trabaja con una fuente de 12V y un módulo relé, el cual se activa al pasar la tarjeta o llavero de identificación que

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 7 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

podrá brindar el acceso y la chapa eléctrica se abrirá, en caso de error envía una alarma mediante mensaje de texto y correo electrónico

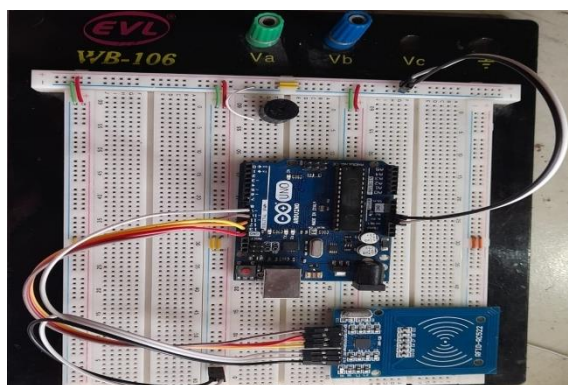


Figura 4. Conexión del módulo RFID con Arduino.

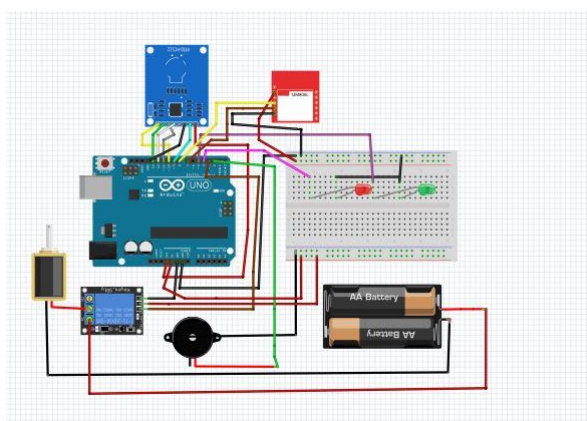



Figura 5. Conexión completa del control de acceso con RFID y Sim800L.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 8 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3: Programación

Después de las conexiones, proceder a realizar la programación mediante el uso del programa de Arduino, para comenzar se debe descargar la librería MFRC522, seguir con la codificación con uso de parámetros establecidos por parte de cliente como por ejemplos números de tarjetas a usar, tiempo de espera para que no refleje error e intruso.

Paso 3.1. Iniciamos en incluir las librerías a usar.

Comenzamos a programar, debemos definir las librerías de los dispositivos a usar, para que se pueda efectuar el proceso.

```
#include <SPI.h> // Include: permite añadir las librerías para poder hacer uso de
diferentes dispositivos

#include <MFRC522.h>

#include <SoftwareSerial.h>
```

Paso 3.2. Definir entradas a utilizar

```
#define SS_PIN 10 // número de pines a usar

#define RST_PIN 9


#define LED_G 5

#define LED_R 4

#define RELAY 3

#define BUZZER 2
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 9 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3.3. Declarar variables para la ejecución del programa

```

voidsetup() { //iniciar variables

Serial.begin(9600); // iniciar una comunicación serial

  _buffer.reserve(50);

sim.begin(9600);

SPI.begin();

  mfr522.PCD_Init();

pinMode(LED_G, OUTPUT); //pinMode permite configurar cada pin como entrada o salida

pinMode(LED_R, OUTPUT);

pinMode(RELAY, OUTPUT);

pinMode(BUZZER, OUTPUT);

noTone(BUZZER);


digitalWrite(RELAY,LOW);

Serial.println("Bienvenidos"); // Serial.println permite escribir datos en el puerto serial

Serial.println("Porfavor ingrese tarjeta");

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 10 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3.4. Ejecutamos los comandos de los resultados a obtener

```
voidloop() { //permite ejecutar los comandos en la placa
if(! mfr522.PICC_IsNewCardPresent()){
return;}
if(! mfr522.PICC_ReadCardSerial()){
return;}
Serial.print("Tag");
Stringcontent="";
byte letter;
for(byte i=0; i < mfr522.uid.size; i++){
Serial.print(mfr522.uid.uidByte[i] < 0x10 ? "0:" " ");
Serial.print(mfr522.uid.uidByte[i], HEX);
content.concat(String(mfr522.uid.uidByte[i]<0x10 ? "
0" : " "));
content.concat(String(mfr522.uid.uidByte[i], HEX));
}Serial.println();
Serial.print("Mensaje:");
content.toUpperCase();
if (content.substring(1) == "29 74 C2 98"){ //permite
ingresar una condicion
Serial.println("Acceso autorizado, Bienvenido");
Serial.println();
delay(500);
digitalWrite(RELAY, HIGH);
digitalWrite(LED_G, HIGH);
digitalWrite(RELAY, LOW);

digitalWrite(LED_G, LOW);
```

```
else {
Serial.println("Acceso denegado - Enviando
SMS....");
digitalWrite(LED_R, HIGH);
tone(BUZZER, 300);
delay(DENIED_DELAY);
digitalWrite(LED_R, LOW);
noTone(BUZZER);

sim.write("AT+CMGF=1\r\n");//Permite
utilizar comandos AT
delay(200);
sim.println("AT+CMGS=\""+numero+"\" \r")
;
delay(200);
Stringsms="Error, Acceso
Denegado...Verificar";
sim.println(sms);

delay(100);


sim.println((char)26);

delay(200);

}

}
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 11 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

4.2. CONTROL DE ACCESO CON LECTOR DE HUELLAS

El procedimiento para realizar un control de acceso mediante uso de Arduino es reconocer los funcionamientos de que cada equipo, a que voltaje trabaja para no producir daños en ellos, y realizar la programación y conexiones de la manera correcta, para una mejor operatividad.

Paso 1: Verificar las entradas y salidas del Arduino.

Es fundamental conocerlas para empezar a programar Arduino. En el Arduino UNO existen entradas y salidas analógicas (6) y digitales (16). Por un lado, las analógicas son utilizadas, generalmente, para dispositivos de entrada y nos permiten leer un rango de valores. En cambio, las digitales pueden tener a la entrada o a la salida 0V o 5V, esto nos indica que ese pin está en estado bajo (Low) o alto (High).

Paso 2: Conexiones de los dispositivos.

Se debe a verificar que entradas se va a utilizar, cablear con jumpers el lector de huellas hacia el Arduino, luego conectar como alarma de sonido un buzzer, para seguir con el uso de un solenoide que trabaja con una fuente de 12V y un módulo relé, el cual se activa al poner el dedo de identificación al lector de huellas que podrá brindar el acceso y la chapa eléctrica se abrirá, en caso de error envía una alarma mediante mensaje de texto y correo electrónico.

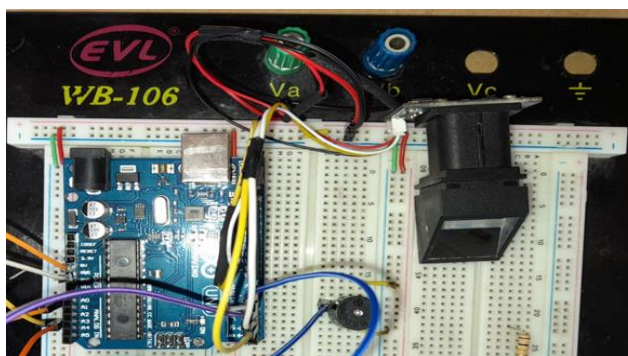



Figura 6. Conexión del lector de huella con Arduino.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 12 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

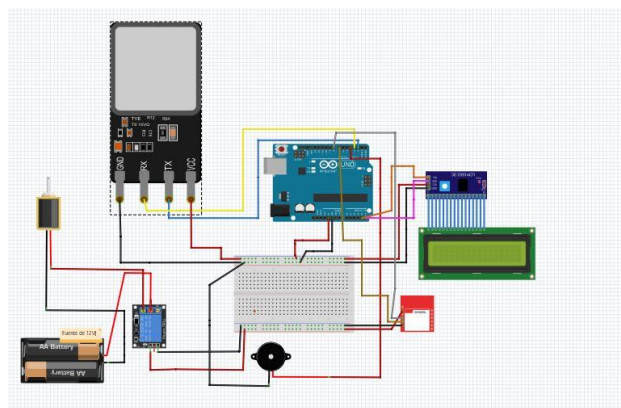


Figura 7. Conexión completa del control de acceso lector de huella.

Paso 3: Programación

Después de las conexiones, se debe proceder a realizar la programación mediante el uso del programa de Arduino, para comenzar se debe descargar la librería “adafruit-fingerprint”, seguir con la codificación con uso de parámetros establecidos por parte de cliente como por ejemplos números de huellas a usar, tiempo de espera para que no refleje error e intruso.


Paso 3.1. Iniciamos en incluir las librerías a usar.

Para comenzar a programar se debe definir las librerías de los dispositivos a usar, para que se pueda efectuar el proceso.

```
#include <Adafruit_Fingerprint.h>//Permite incluir librerías con la línea de Código
include

#include <SoftwareSerial.h>
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 13 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		


Paso 3.2. Definir entradas a utilizar

```

intgetFingerprintIDez();
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
SoftwareSerial mySerial(2, 3);
SoftwareSerial sim(7, 8);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
LiquidCrystal_I2C lcd(0x27, 20,4);
int ID_reg1 = 2;
int ID_reg2 = 8;
intgetFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) return -1;
  p = finger.image2Tz();
  if (p != FINGERPRINT_OK) return -1;
  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK) return -1;
  // found a match!
  Serial.print("Found ID #");
  Serial.print(finger.fingerID);
  Serial.print(" with confidence of ");
  Serial.println(finger.confidence);}

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 14 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3.3. Declarar variables para la ejecución del programa

```
voidsetup() {
Serial.begin(9600);
Serial.println("Prueba de seguridad");
finger.begin(57600);
lcd.init();
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Bienvenidos");
lcd.setCursor(0,1);
lcd.print("Ingrese Huella");
//lcd.clear();
//if (finger.verifyPassword()) {
//  Serial.println("");
// }else {
//  Serial.println("");
//  while (1);}
// Serial.println("Waitingforvalidfinger...");}
```


Paso 3.4. Ejecutamos los comandos de los resultados a obtener

```
voidloop()
getFingerprintIDez();

if(finger.fingerID==ID_reg1)
{ lcd.clear();
lcd.setCursor(0,0);
lcd.print("Registro Encontrado");
lcd.setCursor(0,1);
lcd.print("Welcome");
lcd.setCursor(0,2);
lcd.print("R1");
```

```
else {
Serial.println("Acceso denegado - Enviando SMS....");
sim.write("AT+CMGF=1\r\n");
delay(200);
sim.println("AT+CMGS=\""+numero+"\"\r");
delay(200);
Stringsms="Error, Acceso Denegado... Verificar";
sim.println(sms);
delay(100);
sim.println((char)26);
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 15 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

4.4. CONTROL DE ACCESO CON TECLADO MATRICIAL

El procedimiento para realizar un control de acceso mediante uso de Arduino es reconocer los funcionamientos de que cada equipo, a que voltaje trabaja para no producir daños en ellos, y realizar la programación y conexiones de la manera correcta, para una mejor operatividad.

Paso 1: Verificar las entradas y salidas del Arduino.

Es fundamental conocerlas para empezar a programar Arduino. En el Arduino Mega existen entradas y salidas analógicas (6) y digitales (16). Por un lado, las analógicas son utilizadas, generalmente, para dispositivos de entrada y nos permiten leer un rango de valores. En cambio, las digitales pueden tener a la entrada o a la salida 0V o 5V, esto nos indica que ese pin está en estado bajo (Low) o alto (High).

Paso 2: Conexiones de los dispositivos.

Para iniciar las conexiones se debe verificar que entradas se va a utilizar, cablear con jumpers el teclado matricial tipo membrana hacia el Arduino, luego conectar como alarma de sonido un buzzer, para seguir con el uso de un solenoide que trabaja con una fuente de 12V y un módulo relé, el cual se activa al poner la clave de identificación el cual podrá brindar el acceso y la chapa eléctrica se abrirá, en caso de error envía una alarma mediante mensaje de texto y correo electrónico

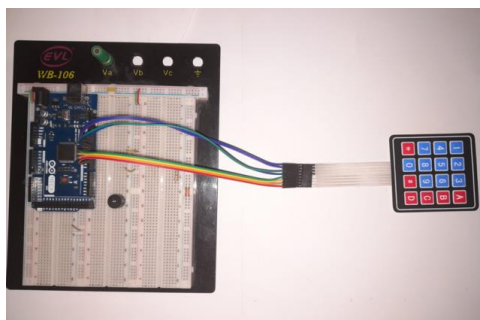



Figura 8. Conexión del teclado matricial con Arduino.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 16 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

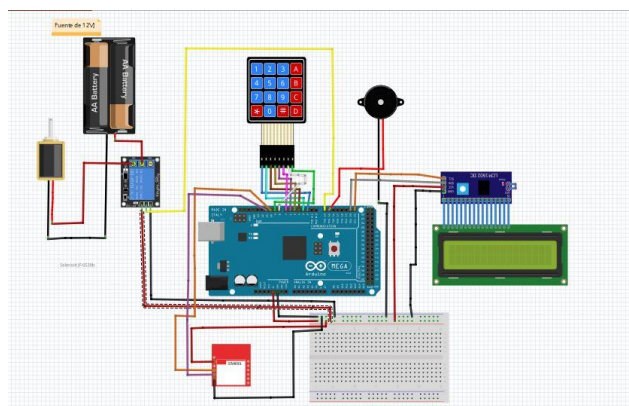


Figura 9. Conexión completa del control de acceso.

Paso 3: Programación

Después de las conexiones, se debe realizar la programación mediante el uso del programa de Arduino, para comenzar se debe descargar la librería “Keypad”, seguir con la codificación con uso de parámetros establecidos por parte de cliente como por ejemplos números de claves a usar, tiempo de espera para que no refleje error e intruso.


Paso 3.1. Iniciamos en incluir las librerías a usar.

En el inicio de la programación se define las librerías de los dispositivos a usar, para que se pueda efectuar el proceso.

```
#include <LiquidCrystal_I2C.h> //Incluir librería de dispositivos a usar

#include <Wire.h>
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 17 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3.2. Definir entradas a utilizar


```
int _timeout;
String _buffer;
String numero = "+593991338815";//Permite tener comunicación con un numero
celular
const byte ROWS = 4;
const byte COLS = 4;
charkeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}};
charpin[5];
intposicion = 0;
byte rowPins[ROWS] = { 9,8,7,6 };
byte colPins[COLS] = { 5,4,3,2, };

Keypad teclado = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS
```

Paso 3.3. Declarar variables para la ejecución del programa

```
voidsetup(){
  Serial.begin(9600);
  _buffer.reserve(50);
  sim.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Bienvenido ");
  lcd.setCursor(0,1);
  lcd.print("Introduce tu pin");
  pinMode(22,OUTPUT);
  pinMode(43,OUTPUT);
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:


		REVISION 1/1	Página 18 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3.4. Ejecutamos los comandos de los resultados a obtener

```
voidloop(){
pin[posicion] = teclado.getKey();
if (pin[posicion]){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Introduciendo");
for(int i = 0; i <= posicion; i++){
lcd.setCursor(i,1);
lcd.print("*");}
posicion++;
if(posicion == 5) posicion = 0;}
if(pin[4] == 'A'){
pin[4] = 'a';
if(pin[0] == '2' && pin[1] == '2'
&& pin[2] == '3' && pin[3] ==
'4'){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Pin correcto ");
lcd.setCursor(0,1);
lcd.print("Acceso Autorizado
");
digitalWrite(22,LOW);
delay(4000);
digitalWrite(22,HIGH);}
```

```
else {
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Pin incorrecto ");
lcd.setCursor(0,1);
lcd.print("Enviando SMS");
digitalWrite(43,HIGH);
delay(500);
digitalWrite(43,LOW);
delay(1500);
Serial.println("Enviando SMS...");
sim.write("AT+CMGF=1\r\n");
delay(200);
sim.println("AT+CMGS=\""+numero+"\"r");
delay(200);
Stringsms="Error, Acceso
Denegado... Verificar";
sim.println(sms);
delay(100);
sim.println((char)26);
delay(200);}
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Bienvenido ");
lcd.setCursor(0,1);
lcd.print("Introduce tu pin");}
delay(10);}
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 19 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

5) RECURSOS UTILIZADOS


- Cables Jumper M-M Y M-H
- Arduino Mega
- Arduino Uno
- Lector de huellas
- Tarjeta rfid
- Pantalla lcd
- Modulo Relé
- Teclado 4x4
- Protoboard
- Buzzer
- Solenoide de 12V
- Leds
- Modulo wifi ESP8266
- Sim 800L

6). RESULTADOS OBTENIDOS

6.1. CONTROL DE ACCESO CON TARJETA RFID

Mediante esta práctica se obtiene como resultado principal el uso y manejo de dispositivos como seguridad, que permiten usar diferentes parámetros de acuerdo con la necesidad del área que lo requiera, teniendo dos tipos de alarmas mediante mensaje de texto y correo electrónico.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 20 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

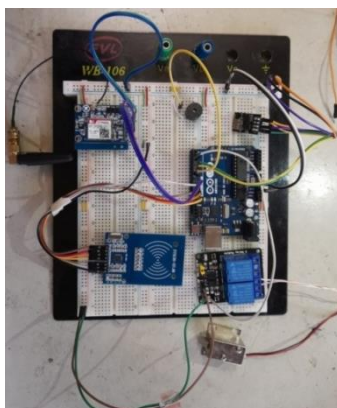


Figura 10. Conexión completa Arduino sensor RFI y Sim8001.

```

Tarjeta_RFID_Control_de_acceso
// UNIVERSIDAD POLITÉCNICA SALESIANA
// CARRERA: INGENIERIA ELECTRONICA MENCION TELECOMUNICACIONES
#include <SPI.h>
#include <MFRC522.h> // Libreria para comunicarse por el bus SPI

#define SS_PIN 10
#define RST_PIN 9
#define LED_G 3 // define green LED pin
#define LED_R 4 // define red LED
#define RELAY 3 // relay pin
#define BUZZER 2 // buzzer pin
#define ACCESS_DELAY 2000
#define DENIED_DELAY 1000
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.


void setup()
{
  Serial.begin(9600); // Iniciamos el puerto serie
  SPI.begin(); // Iniciamos el Bus SPI
  MFRC522.PCD_Init(); // Iniciamos el MFRC522. PCD es el módulo lector (proximity card)
  pinMode(LED_G, OUTPUT);
  pinMode(LED_R, OUTPUT);
  pinMode(RELAY, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  noTone(BUZZER);
  digitalWrite(RELAY, LOW);
  Serial.println("Ingrese por llavero o tarjeta");
  Serial.println();
}

El Sketch usa 7634 bytes (23%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables globales usan 314 bytes (19%) de la memoria dinámica, dejando 1664 bytes para las variables locales. El máximo es 2048 bytes.
Automa Uno en COM5

```

Figura 11. Inicio de programa.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 21 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

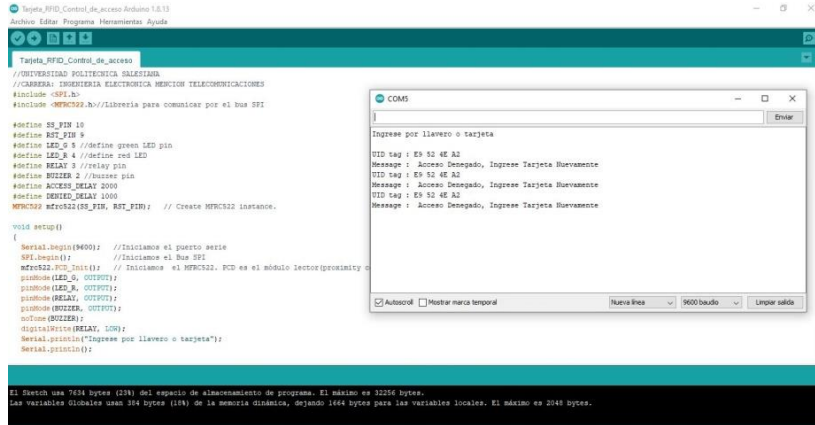


Figura 12. Error de acceso.

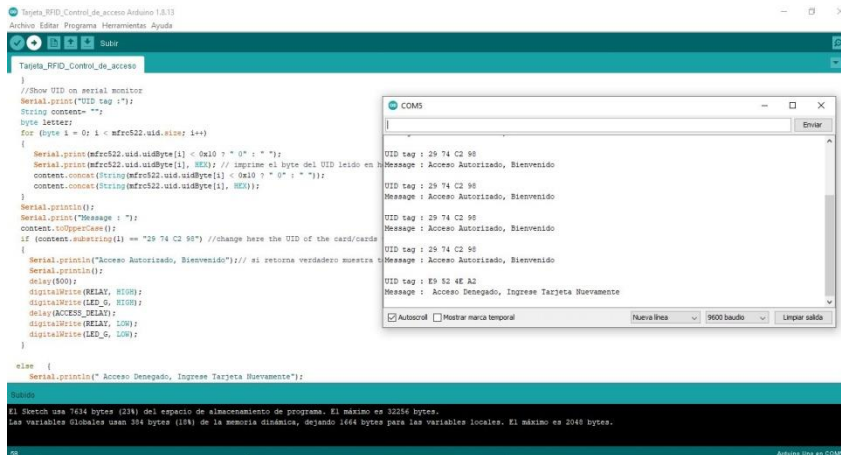



Figura 13. Acceso Autorizado

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 22 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

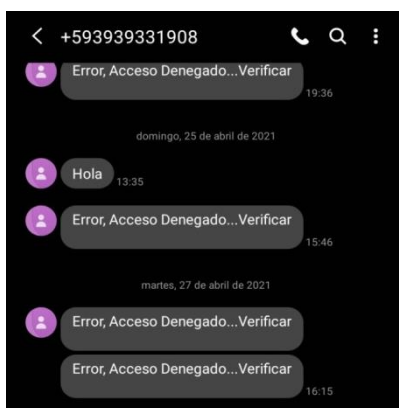


Figura 14. Mensaje de error enviado por SMS

6.2.CONTROL DE ACCESO CON LECTOR DE HUELLAS

Mediante esta práctica se obtiene como resultado principal el uso y manejo de dispositivos como seguridad, que permiten usar diferentes parámetros de acuerdo a la necesidad del área que lo requiera, teniendo dos tipos de alarmas mediante mensaje de texto y correo electrónico

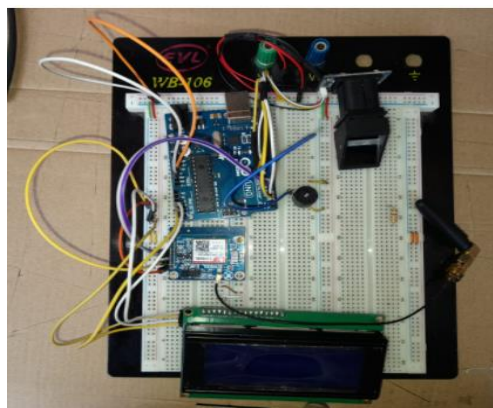

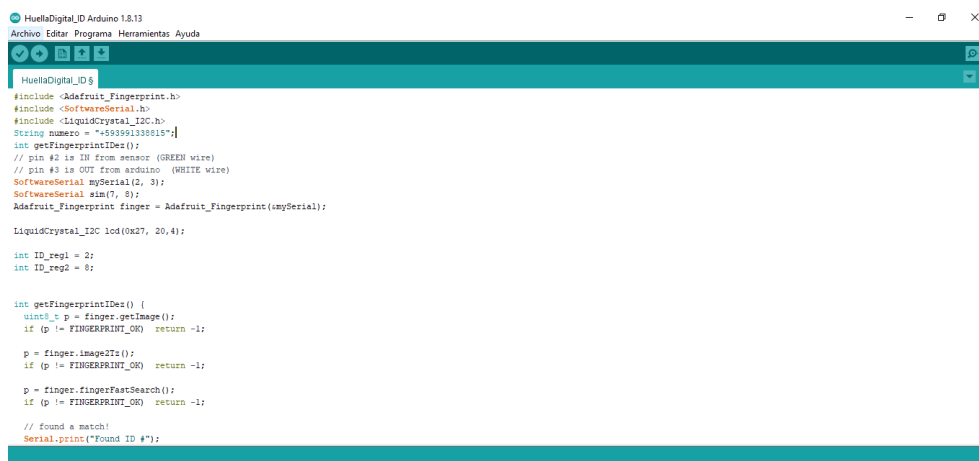


Figura 16. Conexiones.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 23 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



```

HueftaDigital_ID $
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
String numero = "+59399133815";
int getFingerprintIDex();
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
SoftwareSerial mySerial(2, 3);
SoftwareSerial sim(7, 8);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(mySerial);

LiquidCrystal_I2C lcd(0x27, 20,4);

int ID_reg1 = 2;
int ID_reg2 = 8;

int getFingerprintIDex() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) return -1;

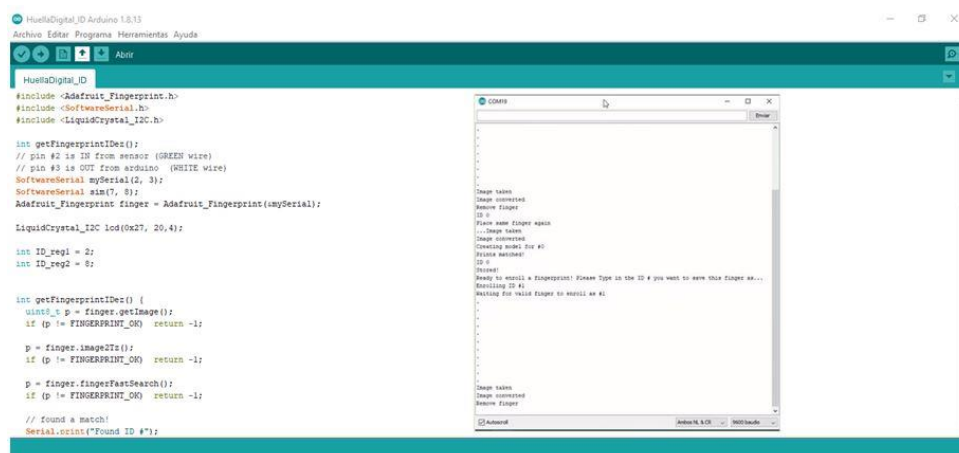
  p = finger.image2Tz();
  if (p != FINGERPRINT_OK) return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK) return -1;

  // found a match!
  Serial.print("Found ID #");

```

Figura 17. . Inicio de programa.



```

HueftaDigital_ID
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>

int getFingerprintIDex();
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
SoftwareSerial mySerial(2, 3);
SoftwareSerial sim(7, 8);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(mySerial);

LiquidCrystal_I2C lcd(0x27, 20,4);

int ID_reg1 = 2;
int ID_reg2 = 8;

int getFingerprintIDex() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK) return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK) return -1;

  // found a match!
  Serial.println("Found ID #");
}

```


```

Serial
Take take
Image captured
Remove finger
ID 0
Place new finger again
--Image taken
Image captured
Cleaning mode for 40
Frame received
ID 0
Result:
Needs to enroll a fingerprint: Place Type in the ID # you want to see this finger at...
Monitoring ID #1
Waiting for valid finger to enroll as #1
...
Take take
Image captured
Remove finger

```

Figura 18. Lectura de huella para acceso.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 24 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

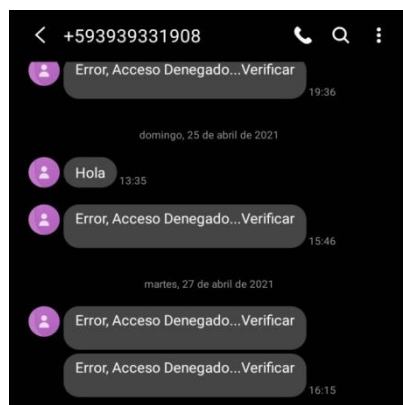


Figura 19. Mensaje de error enviado por SMS

6.3. CONTROL DE ACCESO CON TECLADO MATRICIAL

Mediante esta práctica se obtiene como resultado principal el uso y manejo de dispositivos como seguridad, que permiten usar diferentes parámetros de acuerdo con la necesidad del área que lo requiera, teniendo dos tipos de alarmas mediante mensaje de texto y correo electrónico.

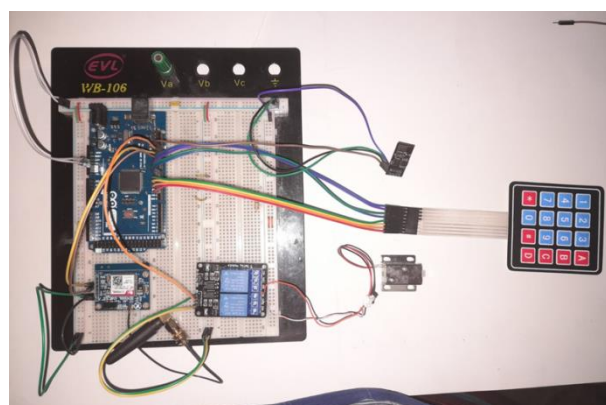



Figura 20. Conexión completa control de acceso con teclado matricial.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 25 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



```

TESIS2 Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda
TESIS2$
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <Keypad.h>
#include <SoftwareSerial.h>
SoftwareSerial sim(10, 11);
LiquidCrystal_I2C lcd(0x27, 20, 4);

int _timeout;
String _buffer;
String numero = "+593991338815";

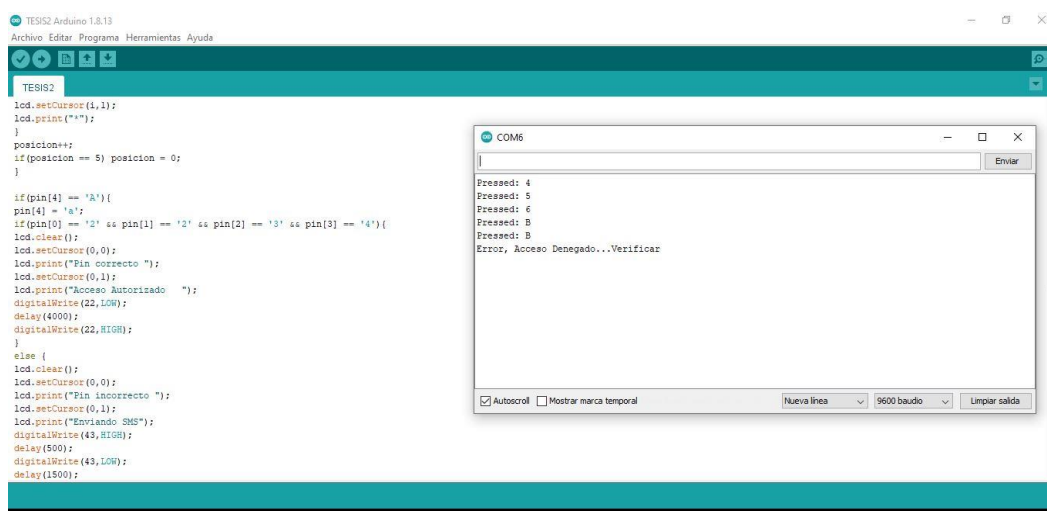
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
char pin[5];
int posicion = 0;
byte rowPins[ROWS] = { 9,8,7,6 };
byte colPins[COLS] = { 5,4,3,2, 1};

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

void setup() {
  Serial.begin(9600);
  _buffer.reserve(50);

```

Figura 21. Inicio de programa.



```

TESIS2
lcd.setCursor(1,1);
lcd.print("");
}
posicion++;
if(posicion == 5) posicion = 0;
}

if (pin[4] == 'A') {
  pin[4] = 'a';
  if (pin[0] == '2' || pin[1] == '2' || pin[2] == '3' || pin[3] == '4') {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Pin correcto ");
    lcd.setCursor(0,1);
    lcd.print("Acceso Autorizado ");
    digitalWrite(22,LOW);
    delay(4000);
    digitalWrite(22,HIGH);
  }
  else {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Pin incorrecto ");
    lcd.setCursor(0,1);
    lcd.print("Enviando SMS");
    digitalWrite(43,HIGH);
    delay(1500);
    digitalWrite(43,LOW);
    delay(1500);

```

COM6

```


Pressed: 4
Pressed: 5
Pressed: 6
Pressed: B
Pressed: B
Error, Acceso Denegado...Verificar

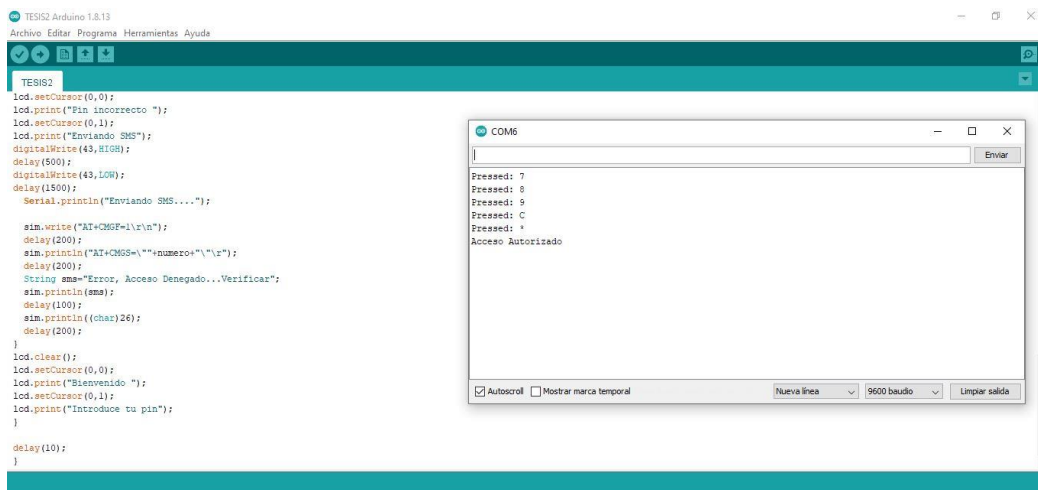
```

Autoscroll Mostrar marca temporal Nueva línea 9600 baudo Limpiar salida

Figura 22. Error de clave.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 26 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



Figura

23. Acceso Autorizado.

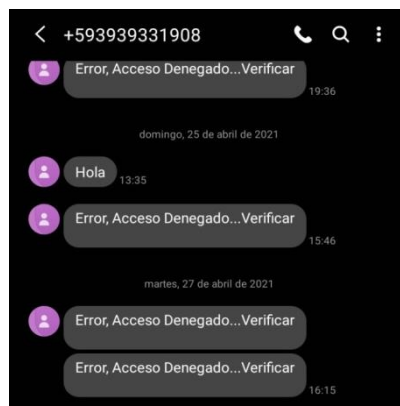




Figura 24. Mensaje de error enviado por SMS

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 27 de 27
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

7) CONCLUSIONES

De acuerdo con los objetivos planteados, en el estudio se demuestra la utilización del módulo Sim800L en un sistema de control de acceso como seguridad electrónica, se busca que este sea el principal medio de alarmas por el cual podemos notificar inalámbricamente futuros proyectos basados en seguridad electrónica a pequeños precios.

8) BIBLIOGRAFÍA UTILIZADA

<https://search.arduino.cc/search?tab=&q=sim800l>

<https://www.rinconingenieril.es/control-de-acceso-con-teclado/>


<https://programarfacil.com/blog/arduino-blog/lector-rfid-rc522-con-arduino/>

<https://www.instructables.com/Lector-De-Huella-Digital-Arduino/>

<https://www.luisllamas.es/arduino-wifi-esp8266-esp01/>

<https://www.geekfactory.mx/tutoriales/tutoriales-arduino/enviar-sms-con-arduino-y-sim800l/>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 1 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Anexo 9. Práctica realizada para la configuración de alertas mediante “Guardado de capturas en Raspberry Pi”.

ELECTRÓNICA Y TELECOMUNICACIONES

PRÁCTICA # 9


DOCENTE: ING. LENIN CEVALLOS

TIEMPO ESTIMADO 2 HORAS

NÚMERO DE ESTUDIANTES: 20

TEMA: “CONFIGURACIÓN DE ALERTAS MEDIANTE GUARDADO DE CAPTURAS EN RASPBERRY PI”

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 2 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

1) OBJETIVO GENERAL.

Configurar el sistema de seguridad de detección de movimiento con la Raspberry Pi añadiendo visualización mediante cámara HD y envío de imágenes mediante correo electrónico como un sistema de seguridad electrónica con elementos pasivos, activos y de alarma.

2) OBJETIVOS ESPECÍFICOS.


- Identificar las herramientas necesarias para los trabajos de programación de una placa de Raspberry Pi.
- Realizar la programación para configurar al raspberry Pi como microcontrolador para sistema de detección de movimiento.
- Añadir al sistema de detección de movimiento una cámara HD para capturar momento de alarma.
- Realizar la correcta conexión entre el dispositivo PIR HC - SR501, Cámara HD y el raspberry Pi mediante Conectores Jumper para establecer un sistema detección de movimiento.
- Realizar la ejecución y verificación de funcionamiento del sistema de detección de movimiento con captura de imagen.
- Realizar el envío de la alarma mediante correo electrónico.

3) MARCO TEÓRICO.

RASPBERRY PI 4 MODEL B

Es una placa del tamaño de una tarjeta de crédito considerado como un microcontrolador que soporta varios componentes necesarios en un computador de escritorio Común. Posee puertos de entrada y salida que permite conectar el microcontrolador a medios físicos externos tales como dispositivos de teclado, ratones, pantallas, cámaras, sensores y actuadores.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Especificaciones:

- Procesador central.
- Procesador Gráfico (GPU).
- Módulo de memoria RAM.
- Conector RJ45.
- 2 buses USB 2.0.
- 2 buses USB 3.0.
- Wifi y Bluetooth integrado.
- Interfaz para cámara y pantalla táctil.
- Salida Analógica de audio/estéreo por Jack de 3,5 mm.
- Salida digital de video MicroHDMI y audio.
- Pines de entrada y salida de propósito General (GPIO).
- 2 puertos de conexión MicroHDMI.
- 1 conector de alimentación MicroUSB tipo C.
- Lector de Tarjeta SD.

SENSOR DE MOVIMIENTO.

El sensor de movimiento (PIR) “Pasive Infra Red” modelo HC-SR501 es un sensor piro eléctrico capaz de medir los niveles de radiación infrarroja emitida por cada cuerpo y objeto en un rango determinado de Máximo 5 metros.


Es un sensor pasivo ideal para la detección de dentro de su franja de barrido.

Su rango de operacional es de 4,5V – 20V.

Tiene 3 pines, que se denominan VCC, OUTPUT y GND.

Tiene 2 potenciómetro para ajustar los parámetros de sensibilidad y retardo. El cual puede configurarse entre 5 y 300 segundos (retardo), y 2 a 5 metros (sensibilidad).

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 4 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

CÁMARA

La cámara deberá ser compatible con el microcontrolador Raspberry Pi como lo son las cámaras 5MPX OV5647 la cual cuenta con un buzz de dato de 15 cm para la conexión con la placa Raspberry mediante el puerto de cámara.

La cámara OV5647 funciona con un voltaje bajo, su sensor de imagen CMOS de 5 megapíxeles nos brinda un rendimiento alto en resolución y nos proporciona una salida de video de 2592x1944 mediante la tecnología OmniBSITM.

El núcleo del sensor OV5647 transmite datos de pixeles generales a una velocidad contante de fotogramas, indicadas por VSYNC y HREF.

PYTHON 3

Python es un lenguaje de programación con un amplio campo de aplicación con énfasis en la legibilidad y ponderación de palabras claves estándar en ingles con fácil manejo al escribir y leer.

Es un lenguaje interpretativo e interactivo con el usuario está orientado a objetos y ofrece estructuras de datos de alto nivel por un tipado dinámico y puntual.

Su principal plataforma IDE de visualización es Thonny, un entorno de desarrollo de Python3 y es capaz de presentar los programas de parte natural y de fácil comprensión para principiantes.


PIP3

Es el sistema de gestión de paquetes que es el permite la administración e instalación de software escritos en Python.

VNC

Es una aplicación de procedimientos libre, su arquitectura es cliente – servidor lo cual nos permite visualizar la información del ordenador servidor remotamente a través de un dispositivo cliente.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 5 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

BUZZER

Es un dispositivo capaz de convertir las señales eléctricas en sonido, tiene dos cables de conexión que sobresalen de su gabinete de fabricación, estos cables representan la polaridad del dispositivo los cuales son GND y VCC que se conectan a cualquier microcontrolador ya sea Raspberry o Arduino.

PROTOCOLO SMTP.

Protocolo SMTP (Simple Mail Transfer Protocol o protocolo para transferencia simple de correos), es un protocolo de red que permite la comunicación de mensajes vía correo electrónico entre dispositivos, computadoras, impresoras, teléfonos móviles entre otros.

El servidor SMTP es el encargado del transporte del correo electrónico, como requisito nos solicita confirmar las identidades del remitente y del destinatario, una vez registrados nos permitirá el envío de este.

Utiliza los puertos 25 o 587 para la serie de intercambios de información entre el servidor SMTP del cliente de un correo electrónico y el servidor SMTP del correo electrónico destino.

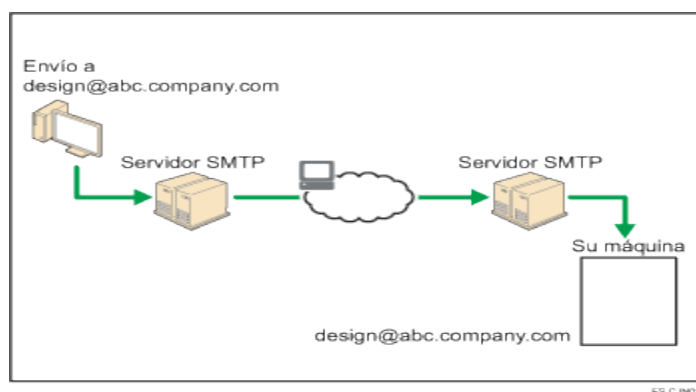



Figura1. Diagrama de Protocolo SMTP

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 6 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

SMTPLIB

El lenguaje de programación Python nos proporciona una librería de nombre Smtplib que define un objeto cliente SMTP para poder utilizar, enviar y recibir correos electrónicos a cualquier dispositivo.

MIMETEXT

Es una clase del protocolo SMTP, Sus siglas (MIME) se traduce como extensión de multipropósito de correo e internet, son una serie de reglas de escritura y estructura del correo electrónico.

Accesorios a usar:

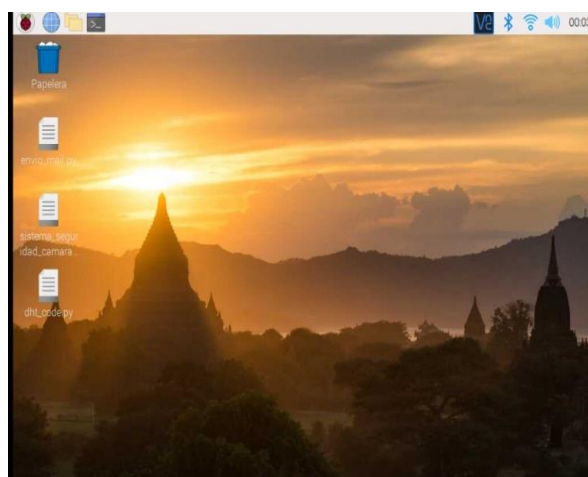
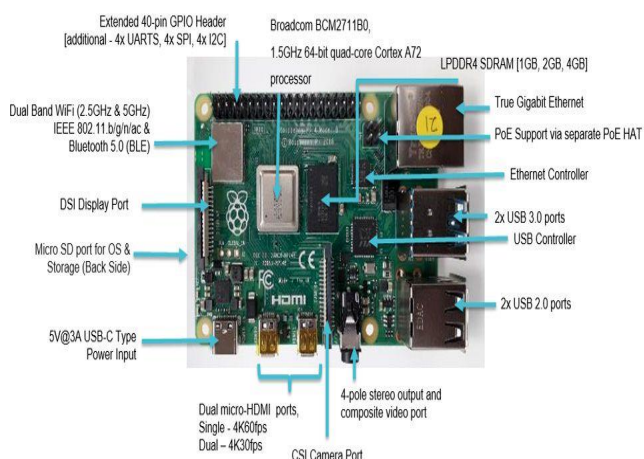


Figura 2. Raspberry Pi model 4B

Figura 3. Entorno virtual Raspberry Pi OS

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:


		REVISION 1/1	Página 7 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		



Figura 4. Sensor PIR HC-SR501.

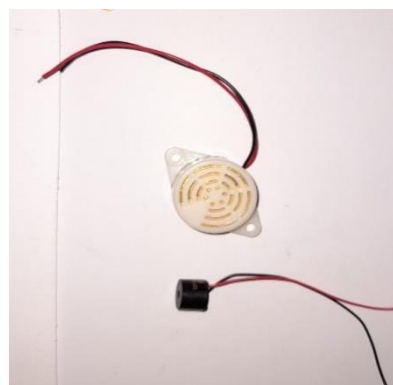


Figura 5. Buzzer.

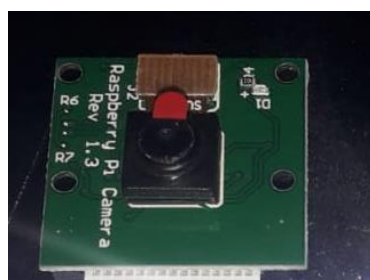



Figura 6. Cámara HD 5MPX OV564.

4) MARCO PROCEDIMENTAL

PROCEDIMIENTO

El procedimiento para realizar un sistema de detección de movimiento en una placa Raspberry Pi modelo 4B es mediante comandos programables en la pestaña de Terminal y Thonny Python, para esto se necesitará un sensor compatible con el Raspberry Pi, en el proyecto será el sensor PIR HC-SR501 la cual ira capturando señales infrarrojas que estén atravesando su rango radiación en un determinado tiempo, una vez detectadas el sensor enviara un pulso positivo o señal lógica uno al microcontrolador quien ordenará a la cámara capturar el espacio que este siendo vulnerado.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 8 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

La cámara deberá estar conectada al microcontrolador Raspberry Pi mediante el buzz de conexión al puerto CSI cámara de la placa. Para que esto ocurra debe existir la codificación necesaria para que trabajen en regla de la cadena el sensor y la cámara de Raspberry, en conexión física el sensor de movimiento y la cámara deberán estar alineadas de tal forma que su línea de vista este direccionada hacia el mismo destino.

Una vez detectado el movimiento y capturada la imagen se le debe programar de tal forma que pueda alarmar al controlador de seguridad electrónica del Data Center mediante el correo electrónico indicando las anomalías capturadas, esto se logra mediante el protocolo SMTP, que en el lenguaje de programación Python es SMTPlib el cual permitirá enviar la alarma al correo con la siguiente información:

Movimientos en zona restringida.

Captura de imagen del movimiento.


Envío de alerta con imagen por correo.

Para la implementación se necesitará dos correos electrónicos, uno será el emisor de alerta y el otro como receptor de alerta.

avisosdatacenter@gmail.com
contraseña
raspberrypi4
repciondatacenter@gmail.com
raspberrypi4

La conexión entre estos dispositivos de entrada y procesamiento se realiza físicamente entre sus pines, el sensor PIR cuenta con tres pines de conexión y la placa Raspberry cuenta con sus entradas y salidas de propósito general GPIO, la conexión con la cámara HD se realiza mediante el puerto CSI para dispositivos de transmisión de imagen que tiene

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 9 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

la placa Raspberry Pi. Su configuración es mediante la utilización de Thonny Python para la creación de un archivo ejecutable con formato (.py).

RASPBERRY PI COMO SISTEMA DE DETECCIÓN DE MOVIMIENTO CAPTURA DE IMAGEN Y ALERTA MEDIANTE ENVIO DE CAPTURA POR CORREO ELECTRÓNICO.

Paso 1. Iniciamos Sesión en Raspberry Pi OS mediante VNC.

Para inicializar la práctica se debe energizar a la placa Raspberry Pi 4B con 5v y 3A, inmediatamente iniciamos sesión mediante nuestro protocolo VNC.

Recordandolas credenciales de acceso:

Dirección Ip del raspberry Pi: <i>192.168.39.167</i> .
Usuario: pi

Contraseña: raspberrypi

(NOTA IMPORTANTE): si no recuerda la dirección Ip del raspberry conecte la placa a un monitor mediante un cable HDMI y en la pestaña comandos realice la búsqueda mediante el código ifconfig.

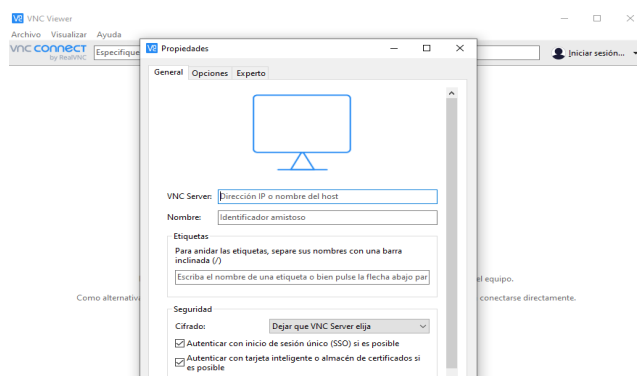



Figura 7. Conexión de acceso remoto VNC.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 10 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 2. Actualización de paquetes Raspberry Pi.

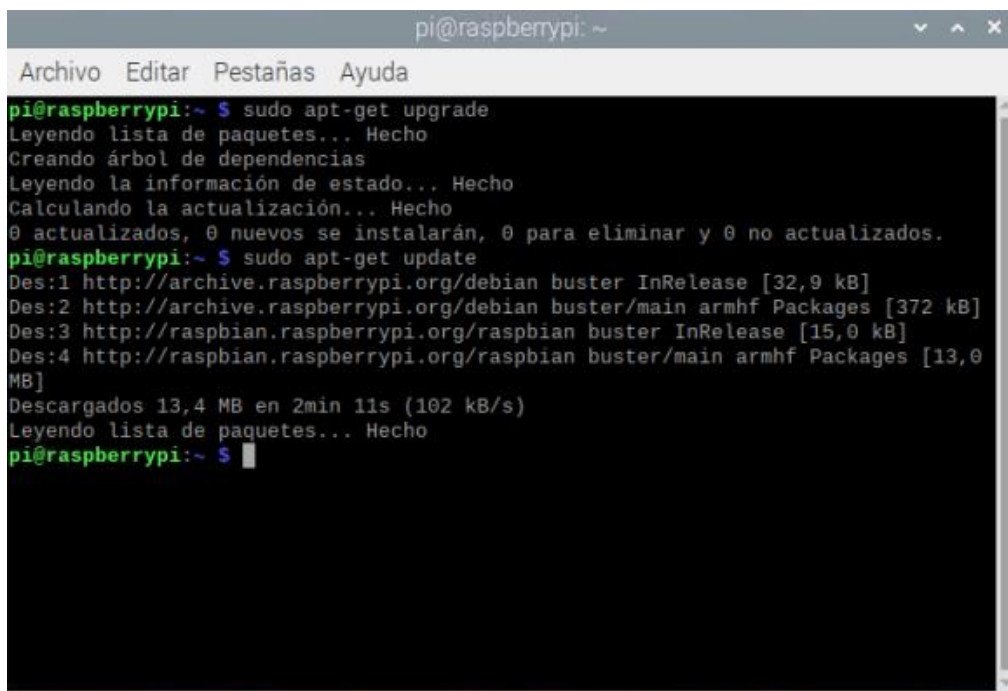
Una vez iniciado la conexión con el ambiente virtual desde el protocolo “VNC”, se procede a iniciar la pestaña de Terminal.

Se recomienda actualizar los paquetes del sistema operativo con el siguiente comando:

```
sudo apt-get update
```

Paso siguiente, se debe actualizar todos sus paquetes instalados a sus últimas versiones con el siguiente comando:

```
sudo apt-get upgrade
```




```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
pi@raspberrypi:~$ sudo apt-get update
Des:1 http://archive.raspberrypi.org/debian buster InRelease [32,9 kB]
Des:2 http://archive.raspberrypi.org/debian buster/main armhf Packages [372 kB]
Des:3 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Des:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0 MB]
Descargados 13,4 MB en 2min 11s (102 kB/s)
Leyendo lista de paquetes... Hecho
pi@raspberrypi:~$

```

Figura 8. Actualización de paquetes.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 11 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 3. Creación de código ejecutable en formato (xxx.py) para activación de sensor PIR y cámara HD y envió por protocolo SMTPlib a correo electrónico.

Una vez actualizado los paquetes de microcontrolador Raspberry Pi, se debe crear un código ejecutable en un formato que una al sensor PIR a la cámara HD para Raspberry y al controlador mediante la combinación de conexiones física, muestra de resultados en pantalla y envió de alarma con imagen adjuntada mediante correo electrónico.

La práctica #9 es la continuación de la práctica #7 y su unión con la práctica #2 por lo cual su código es parecido. Para iniciar la implementación se debe abrir la aplicación de programación virtual Thonny Python y escribir las siguientes líneas de códigos:

CÓDIGO:

```
frompicameraimportPiCamera
from time importsleep
importSmtplib
```

#Importamos librería de la cámara
#importamos la librería de tiempo y apagado


#importamos la librería de SMTPlib que es el protocolo de Python de SMTP (Simple Mail Transfer Protocol)

```
gmail_user = 'avisosdatacenter@gmail.com'
gmail_password = 'raspberrypi4'
```

#Declaramos la cuenta y usuario que va a enviar la alerta y el password

```
import time
from date time import date time
from email.mime.image import MIMEImage
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 12 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
from email.mime.multipart import MIMEMultipart
import RPi.GPIO as GPIO
import time
```

#Importamos la clase MIMEMultipart para el formato de adjuntar objeto en formato MIME

#Importamos la clase MIMEImage ya que todos los formatos del protocolo SMTP son transmitido a base de este formato MIME.

~~**#Importamos la librería GPIO para la configuración**~~

```
correo_destino = 'repciondatacenter@gmail.com'
correo_envio = 'avisosdatacenter@gmail.com'
```

Subject='Seguridad Vulnerada'

#Declaramos una variable correo_destino que va a contener el correo electrónico que nos permitirá recibir la alerta

#Declaramos una variable correo_envio que nos permitirá enviar la alerta

~~**#Declaramos una variable Subject que guardara el asunto del correo**~~

```
GPIO.setmode(GPIO.BCM)
```


#Declaramos la utilización de los Pines de GPIO de la Raspberry como entradas digitales

~~P=PiCamera()~~

```
P.resolution= (1024,756)
```

```
P.start_preview()
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 13 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

#Declaramos variables para la librería Pi cámara y ajustamos su resolución

#Inicializamos la lectura de imagen

```

alarma = 24

GPIO.setup(24, GPIO.OUT)

GPIO.setup(23, GPIO.IN)

```

#Declaramos una variable alarma con el pin lógico 24

#Declaramos la entrada GPIO 24 como entrada (sensor) y GPIO 24 como salida (buzzer).

```

try:
while True:
ifGPIO.input(23):
print("intruso detectado...")
#calentamiento de la cámara
time.sleep(1)
GPIO.output(alarma,True)
time.sleep(1)
GPIO.output(alarma,False)
time.sleep(1)
P.capture('alertaintruso.jpg')


time.sleep(4)

```

#Creamos una condicional Try para las instrucciones de sensor y alertar

#Cuando la entrada del GPIO 23 reciba un uno lógico, el microcontrolador activara el buzzer por 1 segundo

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 14 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

#La entrada GPIO 23 se activará si el sensor PIR capta un movimiento dentro de su rango de medición

#Cuando el buzzer se active, secuencialmente el microcontrolador captura la imagen y la guardara en el escritorio.

```
subject='Seguridad Vulnerada!!'
msg = MIMEMultipart()
msg['Subject'] = subject
msg['From'] = correo_envio
msg['To'] = correo_destino
```

#Declaramos los atributos del mensaje y el cuerpo de mensaje.

```
fp= open('alertaintruso.jpg','rb')
img = MIMEImage(fp.read())
fp.close()
msg.attach(img)
```

#Se genera el archivo, declaramos una variable para que contenga el formato MIMEImage para que tenga el formato MIME de imagen y podamos enviarla por el protocolo SMTP.


```
server = smtplib.SMTP('smtp.gmail.com',587)
server.starttls()
P.stop_preview()
server.login(correo_envio, contraseña)
server.sendmail(correo_envio,correo_destino,msg.as_string())
print("Alarma enviada al correo.")
server.quit()
```

#Definimos los parámetros del servidor de correo electrónico

#Declaramos método SMTP de la clase SMTPlib

#Declaramos el servidor del correo que estemos usando en este caso es el servidor de Gmail

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

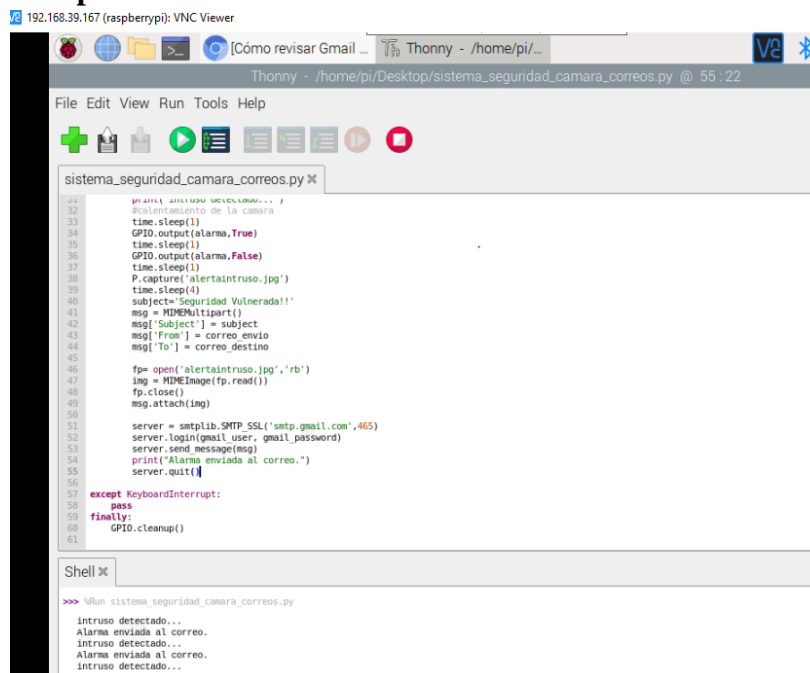
		REVISION 1/1	Página 15 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

**#Declaramos el puerto de comunicación del servidor de correo SMTP del correo Gmail.
#Cada servidor tiene un puerto de comunicación distinto.**

#Para el servidor de correo de Hotmail el puerto de comunicación es 587.

```
except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()
```


**#Declaramos un comando para detener la ejecución el cual es CTRL + C.
#Finalmente limpiamos el while true.**



```
192.168.39.167 (raspberrypi): VNC Viewer
Thonny - /home/pi/...
Thonny - /home/pi/Desktop/sistema_seguridad_camara_correos.py @ 55:22
File Edit View Run Tools Help
sistema_seguridad_camara_correos.py
24 #INICIAR ALERTAS DE MOVIMIENTO...
25 #Alineamiento de la camara
26 time.sleep(1)
27 GPIO.output(alarma,True)
28 time.sleep(1)
29 GPIO.output(alarma,False)
30 time.sleep(1)
31 P.capture('alertaintruso.jpg')
32 time.sleep(4)
33 subject='Seguridad Vulnerada!!'
34 msg = MIMEText('')
35 msg['Subject'] = subject
36 msg['From'] = correo_envia
37 msg['To'] = correo_destino
38
39 fp=open('alertaintruso.jpg','rb')
40 img = MIMEImage(fp.read())
41 fp.close()
42 msg.attach(img)
43
44 server = smtplib.SMTP_SSL('smtp.gmail.com',465)
45 server.login(gmail_user, gmail_password)
46 server.send_message(msg)
47 print("Alarma enviada al correo.")
48 server.quit()
49
50 except KeyboardInterrupt:
51     pass
52 finally:
53     GPIO.cleanup()
54
55 Shell
56 >>> %Run sistema_seguridad_camara_correos.py
57 intruso detectado...
58 Alarma enviada al correo.
59 intruso detectado...
60 Alarma enviada al correo.
61 intruso detectado...
```

Figura 9. Código fuente de sistema de detección de movimiento e imagen.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 16 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 4. Instalación de sensor de movimiento en Raspberry Pi conexión con cámara y programación por cámara.

Para la conexión del sensor PIR y la Raspberry Pi, se utilizará cables jumper para comunicar los pines GPIO de la placa y los pines del sensor.

Para la conexión física del sensor de movimiento, el buzzer y la cámara HD se utilizará Jumper a la entrada GPIO de la Raspberry Pi y la entrada CSI para cámaras.

Según nuestro código declaramos que las entradas que utilizaremos de las GPIO serán entradas BCM entradas digitales.


Las conexiones realizadas del sensor PIR:

Vcc – GPIO 5V

GND - GPIO GROUND

SEÑAL – GPIO 23

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 17 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

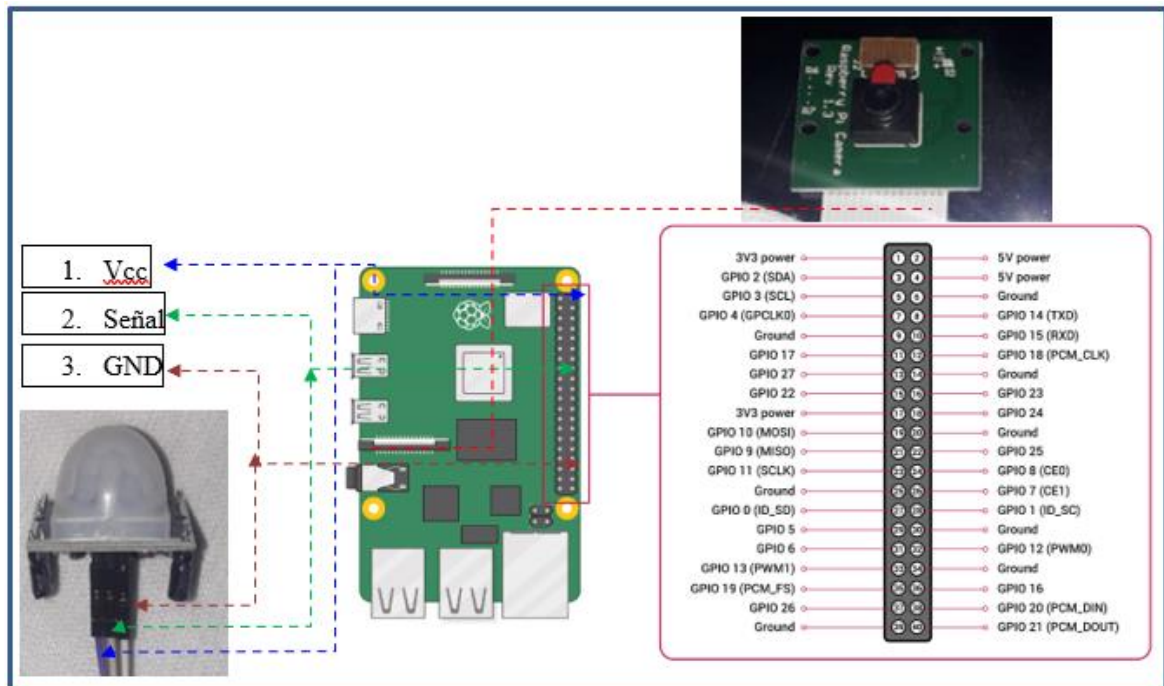



Figura 10. Conexiones físicas del sistema de detección de movimiento con cámara HD

Paso 5. envío de paquetes de alarmas mediante protocolo SMTP a correo electrónico.

Una vez realizadas las conexiones se debe ejecutar cada uno de los códigos con la botonera Run. Al ejecutar el código, el microcontrolador comienza a leer los pines de GPIO en búsqueda del uno lógico, cuando lo recibe muestra por pantalla el mensaje intruso detectado y ordena a la cámara de Raspberry a capturar o grabar mediante su lente y guardarlo en escritorio, cuando la foto es guardada mediante el comando MimeImage se lee y se convierte mediante el código MimeMultipart en un archivo adjunto el cual ya puede ser enviado mediante el protocolo SMTP en la clase SMTPlib hasta el correo repciondatacenter@gmail.com, el cual llegara con el Asunto “Seguridad Vulnerada” y la imagen de nombre aleretaintruso.jpg.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 18 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

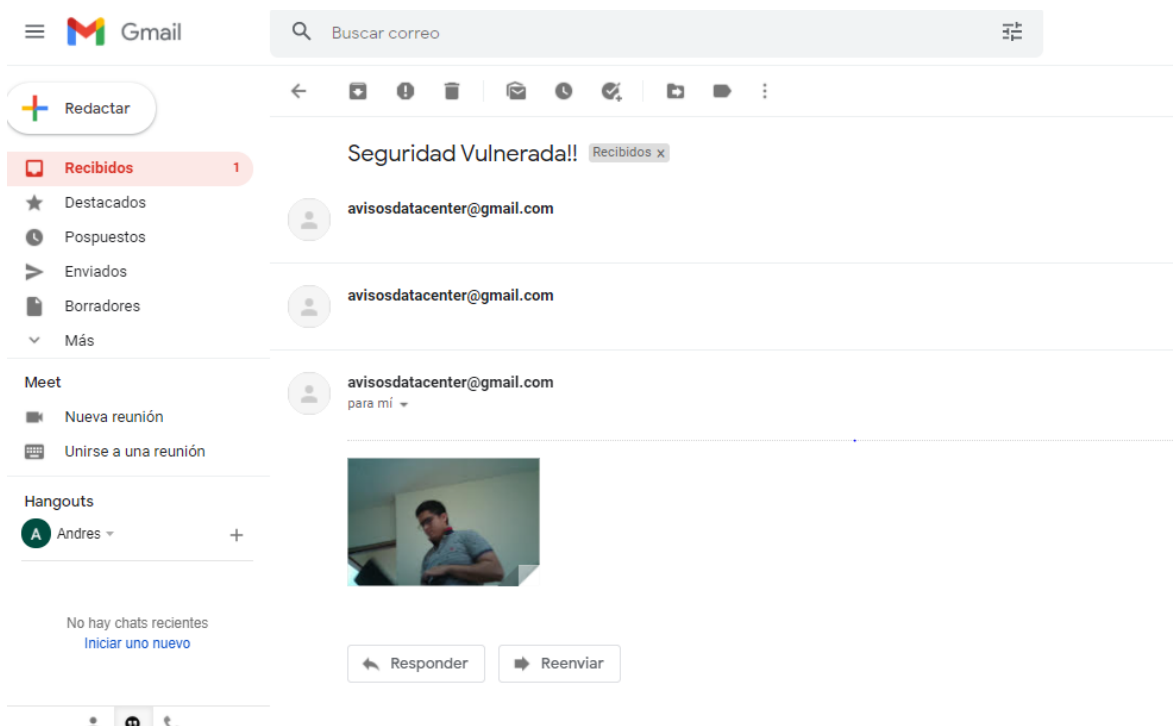



Figura 11. Recepción de correo electrónico con imagen.

5) RECURSOS UTILIZADOS

- Kit Raspberry Pi.
- Conector alimentación 5V 3^a.
- Tarjeta MicroSD.
- Laptop.
- WI-FI.
- Protocolo VNC.
- Sensor PIR HC-SR501
- Buzzer.
- Luces Led.
- Thonny Python.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 19 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

- Cámara HD.
- Protocolo SMTP.
- Jumper.
- Clase SMTPlib.

6) RESULTADOS OBTENIDOS


Mediante la implementación de esta práctica hemos efectuado el envío de alertas por correo de una detección de intruso mediante un sensor de movimiento PIR vinculado a una cámara Raspberry Pi y con un actuador de envío por Protocolo SMTP de la clase SMTPlib, y de esta forma crear el sistema de seguridad completo orientado a la tecnología IOT detectando tanto señal de movimiento como imagen de vulnerabilidad.

Gracias a la implementación de este sistema de seguridad comprendimos como comunicar el sensor PIR a un microcontrolador y programarlo de tal forma que pueda activarse sin la necesidad de que un usuario se encuentre permanente supervisando y poder guardar imagen o video capturado para tener una prueba física en caso de robo y alarmado por correo electrónico al supervisor.

7) CONCLUSIONES

Un sistema de detección de movimiento complementado con una cámara espía y un sistema de actuador de envío por correo nos dan la oportunidad de implementar una seguridad electrónica a un nivel inalámbrico lo que nos reduce el porcentaje de operación humana y automatiza un sistema de seguridad electrónica.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 20 de 20
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

8) BIBLIOGRAFÍA UTILIZADA


<https://projects.raspberrypi.org/en/projects/physical-computing/11>

<https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/7>

<https://www.raspberrypi.org/forums/viewtopic.php?t=49015>

<https://www.raspberrypi.org/forums/viewtopic.php?t=284519>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 1 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Anexo 10.Práctica realizada para la configuración de alertas mediante “envío de correos”.

ELECTRÓNICA Y TELECOMUNICACIONES

PRÁCTICA # 10


DOCENTE: ING. LENIN CEVALLOS

TIEMPO ESTIMADO 2 HORAS

NÚMERO DE ESTUDIANTES: 20

TEMA: “CONFIGURACIÓN Y VISUALIZACIÓN DE ALARMAS MEDIANTE CORREO ELECTRÓNICO CON TECNOLOGÍA RFID Y SENSOR DE MOVIMIENTO”

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 2 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

1) OBJETIVO GENERAL.

Configurar y visualizar de alarmas de sensores por correo electrónico mediante el protocolo SMTP.

2) OBJETIVOS ESPECÍFICOS.

- Identificar las herramientas necesarias para los trabajos de programación de una placa de Raspberry Pi.
- Realizar la programación en Thonny Python que vincule los sensores y el protocolo SMTP clase SMTPlib.
- Realizar el envío de la alarma mediante correo electrónico.

3) MARCO TEÓRICO.


RASPBERRY PI 4 MODEL B

Es una placa del tamaño de una tarjeta de crédito considerado como un microcontrolador que soporta varios componentes necesarios en un computador de escritorio Común. Posee puertos de entrada y salida que permite conectar el microcontrolador a medios físicos externos tales como dispositivos de teclado, ratones, pantallas, cámaras, sensores y actuadores.

Especificaciones:

- Procesador central.
- Procesador Gráfico (GPU).
- Módulo de memoria RAM.
- Conector RJ45.
- 2 buses USB 2.0.
- 2 buses USB 3.0.
- Wifi y Bluetooth integrado.
- Interfaz para cámara y pantalla táctil.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 3 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

- Salida Analógica de audio/estéreo por Jack de 3,5 mm.
- Salida digital de video MicroHDMI y audio.
- Pines de entrada y salida de propósito General (GPIO).
- 2 puertos de conexión MicroHDMI.
- 1 conector de alimentación MicroUSB tipo C.
- Lector de Tarjeta SD.

SENSOR DE MOVIMIENTO.

El sensor de movimiento (PIR) “Pasive Infra Red” modelo HC-SR501 es un sensor piro eléctrico capaz de medir los niveles de radiación infrarroja emitida por cada cuerpo y objeto en un rango determinado de Máximo 5 metros.

Es un sensor pasivo ideal para la detección de dentro de su franja de barrido.

Su rango de operacional es de 4,5V – 20V.

Tiene 3 pines, que se denominan VCC, OUTPUT y GND.

Tiene 2 potenciómetro para ajustar los parámetros de sensibilidad y retardo. El cual puede configurarse entre 5 y 300 segundos (retardo), y 2 a 5 metros (sensibilidad).


CÁMARA

La cámara deberá ser compatible con el microcontrolador Raspberry Pi como lo son las cámaras 5MPX OV5647 la cual cuenta con un buzz de dato de 15 cm para la conexión con la placa Raspberry mediante el puerto de cámara.

La cámara OV5647 funciona con un voltaje bajo, su sensor de imagen CMOS de 5 megapíxeles nos brinda un rendimiento alto en resolución y nos proporciona una salida de video de 2592x1944 mediante la tecnología OmniBSITM.

El núcleo del sensor OV5647 transmite datos de pixeles generales a una velocidad contante de fotogramas, indicadas por VSYNC y HREF.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 4 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

PYTHON 3

Python es un lenguaje de programación con un amplio campo de aplicación con énfasis en la legibilidad y ponderación de palabras claves estándar en inglés con fácil manejo al escribir y leer.

Es un lenguaje interpretativo e interactivo con el usuario está orientado a objetos y ofrece estructuras de datos de alto nivel por un tipado dinámico y puntual.

Su principal plataforma IDE de visualización es Thonny, un entorno de desarrollo de Python3 y es capaz de presentar los programas de parte natural y de fácil comprensión para principiantes.

PIP3

Es el sistema de gestión de paquetes que es el permite la administración e instalación de software escritos en Python.

VNC

Es una aplicación de procedimientos libre, su arquitectura es cliente – servidor lo cual nos permite visualizar la información del ordenador servidor remotamente a través de un dispositivo cliente.


BUZZER

Es un dispositivo capaz de convertir las señales eléctricas en sonido, tiene dos cables de conexión que sobresalen de su gabinete de fabricación, estos cables representan la polaridad del dispositivo los cuales son GND y VCC que se conectan a cualquier microcontrolador ya sea Raspberry o Arduino.

PROTOCOLO SMTP.

Protocolo SMTP (Simple Mail Transfer Protocol o protocolo para transferencia simple de correos), es un protocolo de red que permite la comunicación de mensajes vía correo electrónico entre dispositivos, computadoras, impresoras, teléfonos móviles entre otros.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 5 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

El servidor SMTP es el encargado del transporte del correo electrónico, como requisito nos solicita confirmar las identidades del remitente y del destinatario, una vez registrados nos permitirá el envío de este.

Utiliza los puertos 25 o 587 para la serie de intercambios de información entre el servidor SMTP del cliente de un correo electrónico y el servidor SMTP del correo electrónico destino.

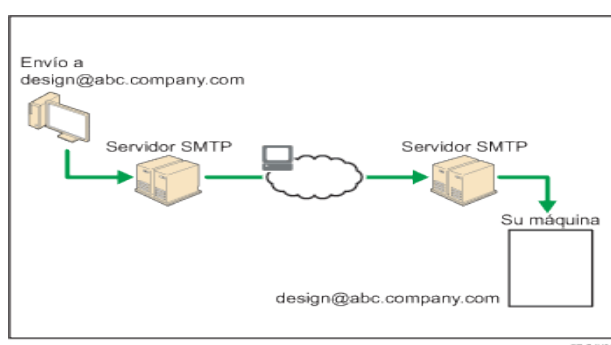


Figura 1. Diagrama de Protocolo SMTP

SMTPLIB

El lenguaje de programación Python nos proporciona una librería de nombre Smtplib que define un objeto cliente SMTP para poder utilizar, enviar y recibir correos electrónicos a cualquier dispositivo.


MIMETEXT

Es una clase del protocolo SMTP, Sus siglas (MIME) se traduce como extensión de multipropósito de correo e internet, son una serie de reglas de escritura y estructura del correo electrónico.

SENSOR DE TEMPERATURA Y HUMEDAD

El dispositivo Sensor de temperatura y humedad, con salida de señal digital, de tamaño reducido y de una precisión tan confiable que demuestra que es el indicado para un sistema

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 6 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

de seguridad electrónica, cuenta con tres pines que sobresalen de su SBC interna, los cuales son diseñados para su alimentación de voltaje +Vcc, tierra y salida de señal captada.

Este sensor digital lleva un microcontrolador para realizar el tratamiento de la señal y este compuesto de un sensor capacitivo para medir la humedad y de un termistor.


Arduino Uno

Arduino Uno es una placa basada en un microcontrolador, con código abierto orientado al microchip Atmega328p y desarrollado por arduino.cc. La placa contiene entradas/salidas digitales y analógicas, en total tiene 14 pines digitales. 6 pines analógicos y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B.

Especificaciones:

- Microcontrolador: ATmega328
- Voltaje de operación: 5V
- Voltaje de entrada (recomendado): 7-12V
- Voltaje de entrada (límites): 6-20V
- Pines de E/S digitales: 14 (de los cuales 6 proporcionan salida PWM)
- Pines de entrada analógica: 6
- Corriente DC por pin de E/S: 40 mA
- Corriente DC para 3.3V Pin: 50 mA
- Memoria Flash: 32 KB de los cuales 0,5 KB utilizados por el bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Velocidad de reloj: 16 MHz

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 7 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Arduino Mega

Es una placa basada en un microcontrolador, tiene 54 entradas/salidas digitales, pueden ser usadas 15 como salidas PWM, 16 entradas analógicas, 4UARTs y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B.


Especificaciones:

- Microcontrolador: ATmega2560
- Voltaje de operación: 5V
- Voltaje de entrada (recomendado): 7-12V
- Voltaje de entrada (límites): 6-20V
- Pines de E/S digitales: 54 (de los cuales 15 proporcionan salida PWM)
- Pines de entrada analógica: 16
- Corriente DC por pin de E/S: 40 mA
- Corriente DC para 3.3V Pin: 50 mA
- Memoria Flash: 256 KB de los cuales 8 KB utilizados por el bootloader
- SRAM: 8 KB (ATmega2560)
- EEPROM: 4 KB (ATmega2560)
- Velocidad de reloj: 16 MHz

Modulo Lector RC522

El módulo RC522 es Lector-Grabador RFID 13.56MHz, posee comunicación SPI lo que permite trabajar fácilmente con la mayoría de los microcontroladores. Utiliza un sistema de modulación y demodulación para todo tipo de dispositivos pasivos RFID de 13.56MHz. El dispositivo maneja el ISO14443-A y soporta el algoritmo de encriptación Quick CRYPTO1 y MIFARE. El rango de detección de tags RFID es de aprox. 5-7cm. Compatible con Arduino, PIC, Raspberry Pi y más.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 8 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Los sistemas RFID son muy útiles para sistemas de control de acceso, seguridad electrónica, trazabilidad.

Tarjeta RFID

Es un sistema de almacenamiento y recuperación de datos remotos que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio

Modulo Relé

Es un dispositivo electromecánico que permite a un procesador como Arduino controlar cargas a un nivel tensión o intensidad muy superior a las que su electrónica puede soportar

Solenoides

Se trata básicamente de una cerradura electrónica, diseñada para un gabinete de base o de seguridad o de puerta. Normalmente, el bloqueo está activo por lo que no puede abrir la puerta porque el lingote del solenoide está en el camino (no utiliza ningún poder en este estado). Cuando se aplica 9 ~ 12 VDC, el lingote se contrae para que no sobresalga más y la puerta se pueda abrir.

Módulo WIFI ESP8266


El módulo ESP8266 es un dispositivo que contiene un chip integrado con accesibilidad Wifi, su diseño fue pensado para ser compatible con el protocolo TCP/IP, y es capaz de dar acceso a cualquier microcontrolador a una red LAN.

El módulo ESP8266 contiene 8 puertos de GPIO (entradas y salidas de propósitos general), las cuales se identifican en dos filas de 4 puertos.

Primera fila superior de izquierda a derecha: GND – GPIO2 – GPIO0 – RXD.

Segunda fila inferior de izquierda a derecha: TXD – CH_PD – RESET – VCC.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 9 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Accesorios a usar:

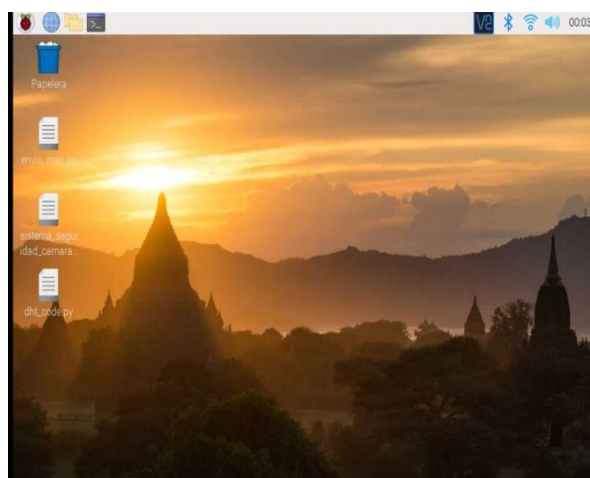
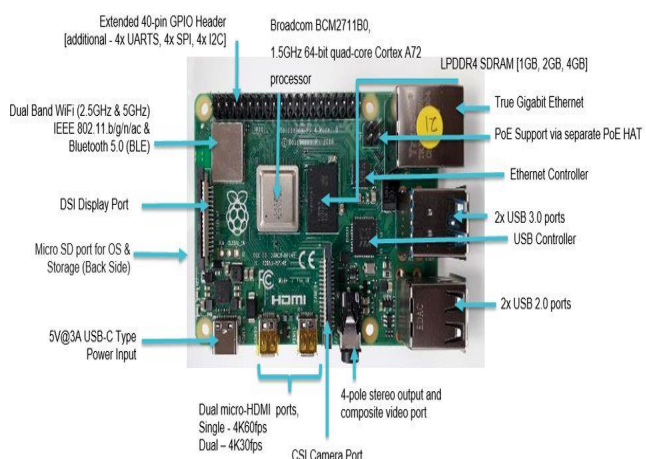


Figura2. Raspberry Pi model 4B

Figura3. Entorno virtual Raspberry Pi OS



Figura4. Sensor PIR HC-SR501.

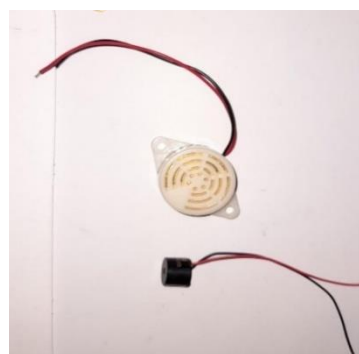



Figura5. Buzzer.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 10 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

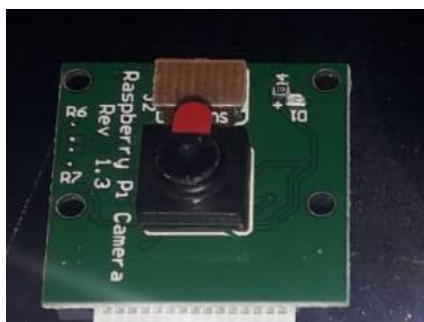


Figura 6. Cámara HD 5MPX OV564.

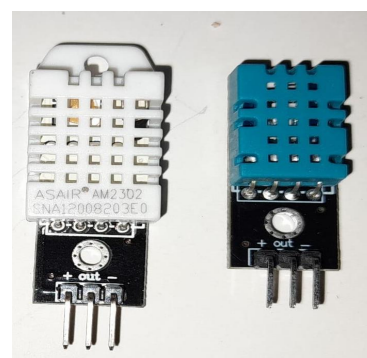


Figura7. Sensor DHT11/22.




Figura8. Kit de práctica con tarjeta RFID



Figura 9. Modulo WIFI ESP8266

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 11 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

4) MARCO PROCEDIMENTAL

PROCEDIMIENTO

El procedimiento para realizar un sistema de envío por correo basado en clase SMTPlib de la librería SMTP se basa en la recolección de datos tomados por los sensores y claves de acceso.

Llegado a este punto tomaremos como base las practicas ya realizadas. Específicamente la práctica #2, práctica #4, práctica #5, práctica #6.

Para ello, utilizaremos dos microcontroladores para la combinación de los sistemas de seguridad, Raspberry Pi trabajará con el sensor PIR (movimiento) y sensor DHT22 (temperatura y humedad), Arduino trabajará con el sensor RFID (infrarrojo).


En raspberry Pi se necesita agregar las librerías de SMTPlib y formato MIME en los códigos ejecutables que ya hemos realizado en las prácticas mencionadas.

En Arduino se necesita descargar y llamar las librerías de SMTP y conectarse mediante cables jumper a un módulo WIFI ESP8266 que nos permitirá conectarnos a la red y poder ejecutar el envío y entrega de los correos.

Para la implementación se necesitará dos correos electrónicos, uno será el emisor de alerta y el otro como receptor de alerta.

avisosdatacenter@gmail.com contraseña raspberrypi4 repciondatacenter@gmail.com raspberrypi4

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 12 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

4.1) RASPBERRY PI COMO SISTEMA DE DETECCIÓN DE MOVIMIENTO CAPTURA DE IMAGEN Y ALERTA MEDIANTE ENVIO DE CAPTURA POR CORREO ELECTRÓNICO.

Paso 1. Iniciamos Sesión en Raspberry Pi OS mediante VNC.

Para inicializar la práctica se debe energizar a la placa Raspberry Pi 4B con 5v y 3A, inmediatamente iniciamos sesión mediante nuestro protocolo VNC.

Recordandolas credenciales de acceso:

Dirección Ip del raspberry Pi: <i>172.18.63.251</i> .
Usuario: pi

Contraseña: raspberrypi

(NOTA IMPORTANTE: Si no recuerda la dirección Ip del raspberry conecte la placa a un monitor mediante un cable HDMI y en la pestaña comandos realice la búsqueda mediante el código ifconfig.

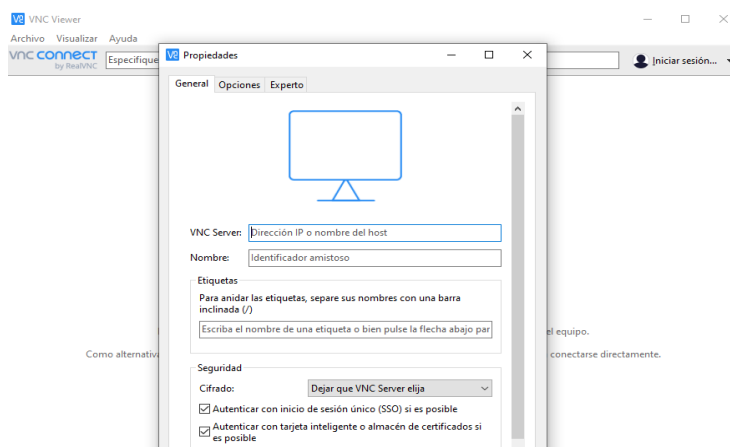



Figura 10. Conexión de acceso remoto VNC.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 13 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 2. Actualización de paquetes Raspberry Pi.

Una vez iniciado la conexión con el ambiente virtual desde el protocolo “VNC”, se procede a iniciar la pestaña de Terminal

Se recomienda actualizar los paquetes del sistema operativo con el siguiente comando:

```
sudo apt-get update
```

Paso siguiente, se debe actualizar todos sus paquetes instalados a sus últimas versiones con el siguiente comando:

```
sudo apt-get upgrade
```

Paso 3. Creación de código ejecutable en formato (xxx.py) para activación de sensor PIR y cámara HD y envió por protocolo SMTPlib a correo electrónico.

Una vez actualizado los paquetes de microcontrolador Raspberry Pi, se debe crear un código ejecutable en un formato que una al sensor PIR a la cámara HD para Raspberry y al controlador mediante la combinación de conexiones física, muestra de resultados en pantalla y envió de alarma con imagen adjuntada mediante correo electrónico.

La práctica #9 es la continuación de la práctica #7 y su unión con la práctica #2 por lo cual su código es parecido. Para iniciar la implementación se debe abrir la aplicación de programación virtual Thonny Python y escribir las siguientes líneas de códigos:

CÓDIGO:


```
From picamera import PiCamera
from time import sleep
import Smtplib
```

#Importamos librería de la cámara

#importamos la librería de tiempo y apagado

#importamos la librería de SMTPlib que es el protocolo de Python de SMTP (Simple Mail Transfer Protocol)

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 14 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
gmail_user = 'avisosdatacenter@gmail.com'
gmail_password = 'raspberrypi4'
```

#Declaramos la cuenta y usuario que va a enviar la alerta y el password

```
import time
from datetime import datetime
from email.mime.image import MIMEImage
from email.mime.multipart import MIMEMultipart
import RPi.GPIO as GPIO
import time
```

#Importamos la clase MIMEMultipart para el formato de adjuntar objeto en formato MIME

#Importamos la clase MIMEImage ya que todos los formatos del protocolo SMTP son transmitido a base de este formato MIME.

#Importamos la librería GPIO para la configuración

```
correo_destino = 'repciondatacenter@gmail.com'
correo_envio = 'avisosdatacenter@gmail.com'
Subject='Seguridad Vulnerada'
```

#Declaramos una variable correo_destino que va a contener el correo electrónico que nos permitirá recibir la alerta


#Declaramos una variable correo_envio que nos permitirá enviar la alerta

#Declaramos una variable Subject que guardara el asunto del correo

```
GPIO.setmode(GPIO.BCM)
```

#Declaramos la utilización de los Pines de GPIO de la Raspberry como entradas digitales

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 15 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
P=PiCamera()
P.resolution= (1024,756)
```

#Declaramos variables para la librería Picamara y ajustamos su resolución

#Inicializamos la lectura de imagen


```
alarma = 4
GPIO.setup(4, GPIO.OUT)
GPIO.setup(17, GPIO.IN)
```

#Declaramos una variable alarma con el pin lógico 24

#Declaramos la entrada GPIO 24 como entrada (sensor) y GPIO 24 como salida (buzzer).

```
try:
while True:
ifGPIO.input(17):
P.start_preview()
print("intruso detectado...")
#calentamiento de la camara
time.sleep(1)
GPIO.output(alarma,True)
time.sleep(1)
GPIO.output(alarma,False)
time.sleep(1)
P.capture('alertaintruso.jpg')
time.sleep(2)
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 16 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

#Creamos una condicional Try para las instrucciones de sensor y alertar

#Cuando la entrada del GPIO 17 reciba un uno lógico, el microcontrolador activara el buzzer por 1 segundo

#La entrada GPIO 17 se activará si el sensor PIR capta un movimiento dentro de su rango de medición

#Cuando el buzzer se active, secuencialmente el microcontrolador captura la imagen y la guardara en el escritorio.

```
subject='Seguridad Vulnerada!!'
msg = MIMEMultipart()
msg['Subject'] = subject
msg['From'] = correo_envio
msg['To'] = correo_destino
```


#Declaramos los atributos del mensaje y el cuerpo de mensaje.

```
fp= open('alertaintruso.jpg','rb')
img = MIMEImage(fp.read())
fp.close()
msg.attach(img)
```

#Se genera el archivo, declaramos una variable para que contenga el formato MIMEImage para que tenga el formato MIME de imagen y podamos enviarla por el protocolo SMTP.

```
server = smtplib.SMTP('smtp.gmail.com',587)
server.starttls()
P.stop_preview()
server.login(correo_envio, contraseña)
server.sendmail(correo_envio,correo_destino,msg.as_string())
print("Alarma enviada al correo.")
server.quit()
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 17 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

#Definimos los parámetros del servidor de correo electrónico

#Declaramos método SMTP de la clase SMTPlib

#Declaramos el servidor del correo que estemos usando en este caso es el servidor de Gmail

#Declaramos el puerto de comunicación del servidor de correo SMTP del correo Gmail.

#Cada servidor tiene un puerto de comunicación distinto.

~~**#Para el servidor de correo de Hotmail el puerto de comunicación es 587.**~~

```
except KeyboardInterrupt:
```

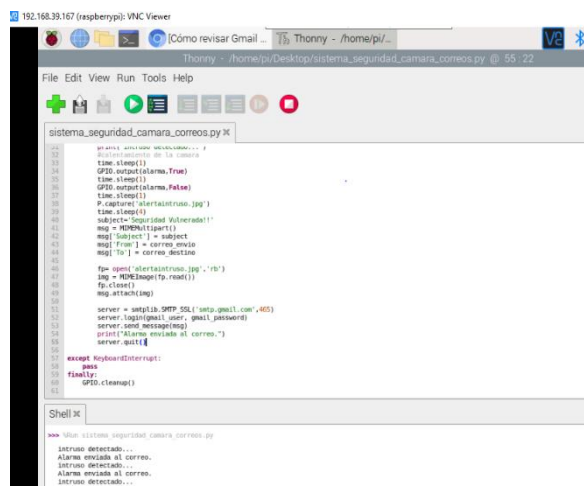
```
pass
```

```
finally:
```

```
GPIO.cleanup()
```

#Declaramos un comando para detener la ejecución el cual es CTRL + C.


#Finalmente limpiamos el while true.



```
sistema_seguridad_camara_correos.py
23  # Definimos los parámetros del servidor de correo electrónico
24  time.sleep(1)
25  GPIO.setup(alarma, True)
26  time.sleep(1)
27  GPIO.setup(alarma, False)
28  time.sleep(1)
29  # Capturamos la imagen
30  time.sleep(1)
31  subject = "Seguridad Vulnerada!!"
32  msg = MIMETextPart()
33  msg["Subject"] = subject
34  msg["From"] = correo_envio
35  msg["To"] = correo_destino
36
37  fp = open("alerta1trono.jpg", "rb")
38  img = MIMEImage(fp.read())
39  fp.close()
40  msg.attach(img)
41
42  server = smtp.SMTP_SSL('smtp.gmail.com', 465)
43  server.login(gmail_user, gmail_password)
44  server.send_message(msg)
45  print("Correo enviado al correo.")
46  server.quit()
47
48  except KeyboardInterrupt:
49      pass
50  finally:
51      GPIO.cleanup()
52
53  Shell x
54
55  *** Run sistema_seguridad_camara_correos.py
56  Intruso detectado...
57  Alarma enviada al correo.
58  Intruso detectado...
59  Alarma enviada al correo.
60  Intruso detectado...
```

Figura 11. Código fuente de sistema de detección de movimiento e imagen.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 18 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 4. Instalación de sensor de movimiento en Raspberry Pi conexión con cámara y programación por cámara.

Para la conexión del sensor PIR y la Raspberry Pi, se utilizará cables jumper para comunicar los pines GPIO de la placa y los pines del sensor.

Para la conexión física del sensor de movimiento, el buzzer y la cámara HD se utilizará Jumper a la entrada GPIO de la Raspberry Py y la entrada CSI para cámaras.

Según nuestro código declaramos que las entradas que utilizaremos de las GPIO serán entradas BCM entradas digitales.


Las conexiones realizadas del sensor PIR:

Vcc – GPIO 5V

GND - GPIO GROUND

SEÑAL – GPIO 23

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 19 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

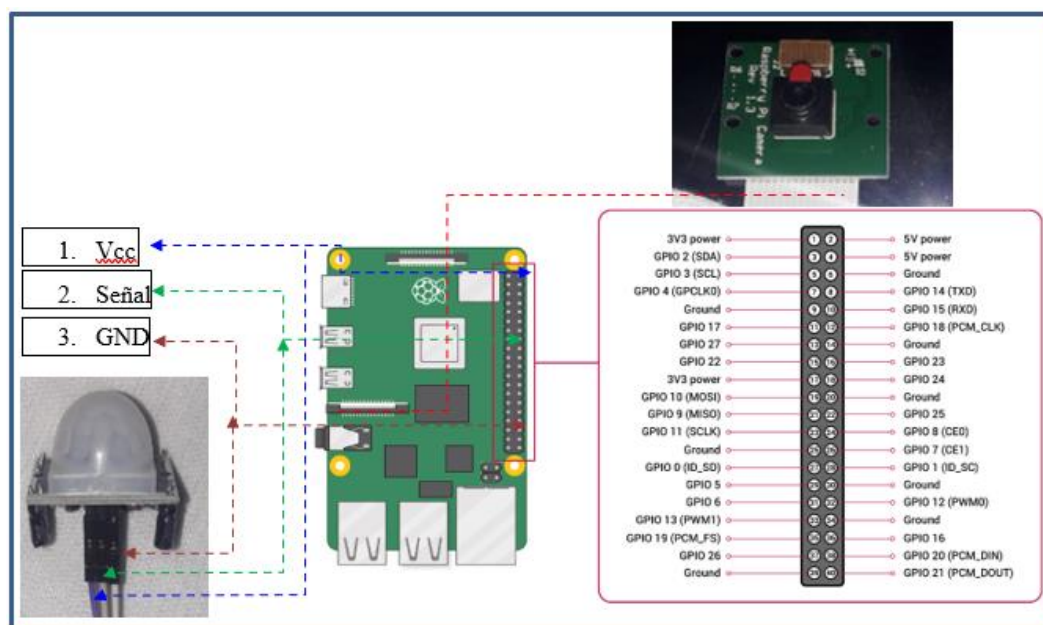



Figura 12. Conexiones físicas del sistema de detección de movimiento con cámara HD

Paso 5. envío de paquetes de alarmas mediante protocolo SMTP a correo electrónico.

Una vez realizadas las conexiones se debe ejecutar cada uno de los códigos con la botonera Run.

Al ejecutar el código, el microcontrolador comienza a leer los pines de GPIO en búsqueda de uno lógico, cuando lo recibe muestra por pantalla el mensaje intruso detectado y ordena a la cámara de Raspberry a capturar o grabar mediante su lente y guardarlo en escritorio, cuando la foto es guardada mediante el comando MimeImage se lee y se convierte mediante el código MimeMultipart en un archivo adjunto el cual ya puede ser enviado mediante el protocolo SMTP en la clase SMTPlib hasta el correo repciondatacenter@gmail.com, el cual llegara con el Asunto Seguridad Vulnerada!! y la imagen de nombre aleretaintruso.jpg.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 20 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

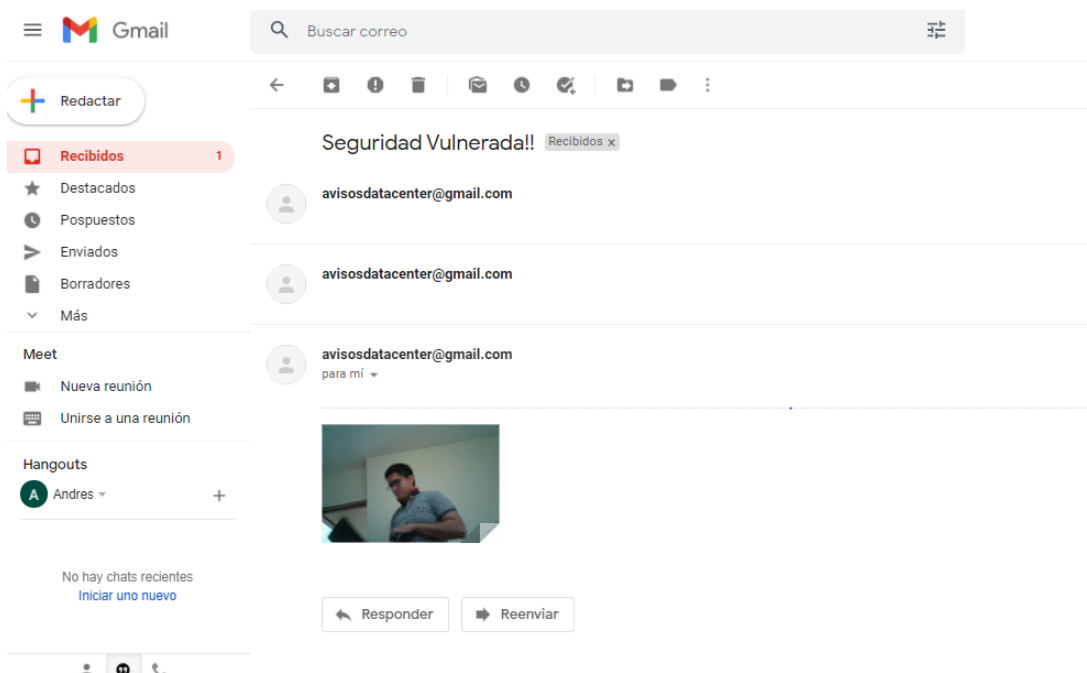


Figura 13. Recepción de correo electrónico con imagen.

4.2) RASPBERRY PI COMO SISTEMA DE MEDICIÓN DE TEMPERATURA Y HUMEDAD CON ENVIO DE ALERTA POR CORREO ELECTRÓNICO.


Paso 1. Iniciamos Sesión en Raspberry Pi OS mediante VNC.

Para inicializar la práctica se debe energizar a la placa Raspberry Pi 4B con 5v y 3A, inmediatamente iniciamos sesión mediante nuestro protocolo VNC.

Recordandolas credenciales de acceso:

Dirección Ip del raspberry Pi: 192.168.39.167. Usuario: pi Contraseña: raspberrypi
--

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 21 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 2. Actualización de paquetes Raspberry Pi.

Una vez iniciado la conexión con el ambiente virtual desde el protocolo “VNC”, se procede a iniciar la pestaña de Terminal.

Se recomienda actualizar los paquetes del sistema operativo con el siguiente comando:

```
sudo apt-get update
```

Paso siguiente, se debe actualizar todos sus paquetes instalados a sus últimas versiones con el siguiente comando:

```
sudo apt-get upgrade
```


Paso 3. Creación de código ejecutable en formato (xxx.py) para activación de sensor DHT11/22.

Una vez actualizado los paquetes de microcontrolador Raspberry Pi, se debe crear un código ejecutable en un formato que una al sensor DHT22 y al controlador mediante la combinación de conexiones físicas y presentación digital adicionando una condicional que ejecute en caso de que la lectura del sensor sea fuera de los parámetros establecidos se alerte por correo electrónico lo acontecido al supervisor Para ello se debe inicializar la aplicación de programación virtual Thonny Python y escribir las siguientes líneas de códigos:

Código:

```
import time
import board
import adafruit_dht
import time
import RPi.GPIO as GPIO
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 22 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
import smtplib
```

```
from email.mime.text import MIMEText
```

```
import time
```

#Declaramos las bibliotecas de “Adafruit_dht”, “Smtplib”, “Mimetext”, “time”

```
correo_envio = 'avisosdatacenter@gmail.com'
```

```
contraseña = 'raspberrypi4'
```

```
correo_destino = 'repciondatacenter@gmail.com, esc-98@outlook.com'
```

#Definimos variables para correos y contraseñas

```
dhtDevice = adafruit_dht.DHT22(board.D4)
```

#Definimos una variable global que contiene toda la librería de “adafruit” el modelo del sensor DHT22 y el número de ping al conectar el sensor.


```
print('Esperando a que se establezca el Sensor de temperatura')
```

```
time.sleep(2)
```

```
print('sensor encendido')
```

#una vez inicie el sistema de medición de temperatura imprimimos que se encendió el sensor esto solo para calentar el sensor DHT mientras se ejecuta las condicionales.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 23 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```

defenvio_correo():

msg = MIMEText("ALERTA. Temperatura superior al límite permitido:
{Temperatura:.2f}°C")

msg['Subject'] = 'PRACTICA 10 - ALERTA POR CORREO - SENSOR DE
TEMPERATURA'

msg['From'] = correo_envio

msg['To'] = correo_destino

server = smtplib.SMTP('smtp.gmail.com',587)

server.starttls()

server.login(correo_envio,contraseña)

server.sendmail(correo_envio,correo_destino,msg.as_string())

print("Su correo se ha procesado con éxito.")

server.quit()

```

#Declaramos la función de envío de correo electrónico mediante la función def.

#Designamos el formato de envío de correo con las variables correo de envío y correo de destino.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:



MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS

LABORATORIO	TELECOMUNICACIONES
CARRERA	INGENIERÍA ELECTRÓNICA
SEDE	GUAYAQUIL

```
try:
```

```
while True:
```

```
time.sleep(5.0)
```

```
temperature_c = dhtDevice.temperature
```

```
temperature_f = temperature_c * (9 / 5) + 32
```

```
humidity = dhtDevice.humidity
```

```
print(
```


```
    "Temp: {:.1f} F / {:.1f} C Humidity: {}% ".format(
```

```
temperature_f, temperature_c, humidity
```

```
)
```

```
)
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 25 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

#Creamos un bucle que nos permita leer la librería “adafruit” y mostrar las mediciones de temperatura y humedad

```

if temperature_c > 33.0:
    print("alerta de temperatura detectado, revisar componentes")
    print("temperatura superior a 33 grados")
    print("alarma encendida")
    envio_correo()

```

#Creamos una condicional, que si la lectura de temperatura es mayor a 33 grados imprima alerta de temperatura

#llame y ejecute la función envio_correo() que contiene la información para envío de correo electrónico.

```

else :
    print("datos de temperatura normales")
    time.sleep(3)

```


#Si la condición anterior no se cumple se debe imprimir datos de temperatura normales y repetir el bucle.

```

except RuntimeError as error:
    print(error.args[0])
    time.sleep(2.0)
except Exception as error:
    dhtDevice.exit()
    raise error
    time.sleep(3)

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 26 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

#Definimos excepciones de lectura por si la toma de información da valor erróneo vuelva a ejecutarse.


```
except KeyboardInterrupt:
pass
finally:
GPIO.cleanup()
```

#Definimos una excepción que nos permita salir del código.

```
temperatura_correo.py
1
2 import time
3 import board
4 import adafruit_dht
5 import time
6 import RPi.GPIO as GPIO
7 import smtplib
8 from email.mime.text import MIMEText
9 import time
10
11 correo_envio = 'avisosdatacenter@gmail.com'
12 contraseña = 'raspberrypi4'
13 correo_destino = 'repciondatacenter@gmail.com, esc-98@outlook.com'
14
15 dhtDevice = adafruit_dht.DHT22(board.D4)
16
17
18 print('Esperando a que se establezca el Sensor de temperatura')
19 time.sleep(2)
20 print('sensor encendido')
21
22
23 def envio_correo():
24
25     msg = MIMEText("ALERTA. Temperatura superior al limite permitido: Temp: {:.1f} F / {:.1f} C")
26     msg['Subject'] = 'PRACTICA 10 - ALERTA POR CORREO - SENSOR DE TEMPERATURA'
27     msg['From'] = correo_envio
28     msg['To'] = correo_destino
29     server = smtplib.SMTP('smtp.gmail.com',587)
30     server.starttls()
31     server.login(correo_envio,contraseña)
32     server.sendmail(correo_envio,correo_destino,msg.as_string())
33     print("Su correo se a procesado con exito.")
34     server.quit()
35
36 try:
37     while True:
38         time.sleep(5.0)
39         temperature c = dhtDevice.temperature
```

Figura 14. Inicio de código fuente de sistema de medición de temperatura y humedad con envío de correo.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 27 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		


```

temperatura_correo.py x
35
36 try:
37     while True:
38         time.sleep(5.0)
39         temperature_c = dhtDevice.temperature
40         temperature_f = temperature_c * (9 / 5) + 32
41         humidity = dhtDevice.humidity
42         print(
43             "Temp: {:.1f} F / {:.1f} C Humidity: {}% ".format(
44                 temperature_f, temperature_c, humidity
45             )
46         )
47     )
48     if temperature_c > 33.0 :
49         print("alerta de temperatura detectado, revisar componentes")
50         print("temperatura superior a 30 grados")
51         print("alarma encendida")
52         envio_correo()
53
54     else :
55         print("datos de temperatura normales")
56         time.sleep(3)
57
58
59
60 except RuntimeError as error:
61     print(error.args[0])
62     time.sleep(2.0)
63
64 except Exception as error:
65     dhtDevice.exit()
66     raise error
67     time.sleep(3)
68
69
70 except KeyboardInterrupt:
71     pass
72 finally:
73     GPIO.cleanup()

```

Figura 15. Finalización de código fuente de sistema de medición de temperatura y humedad con envío de correo.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 28 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 4. Instalación de sensor de movimiento en Raspberry Pi.

Para la conexión del sensor DHT11 y la Raspberry Pi, se debe utilizar cables jumper para comunicar los pines GPIO de la placa y los pines del sensor.

Para ello se debe identificar los pines del sensor, los cuales son tres, que sobresalen de su arquitectura de fabricación, los cuales son:

+VCC (3.3v Pin 1)

OUT

GND

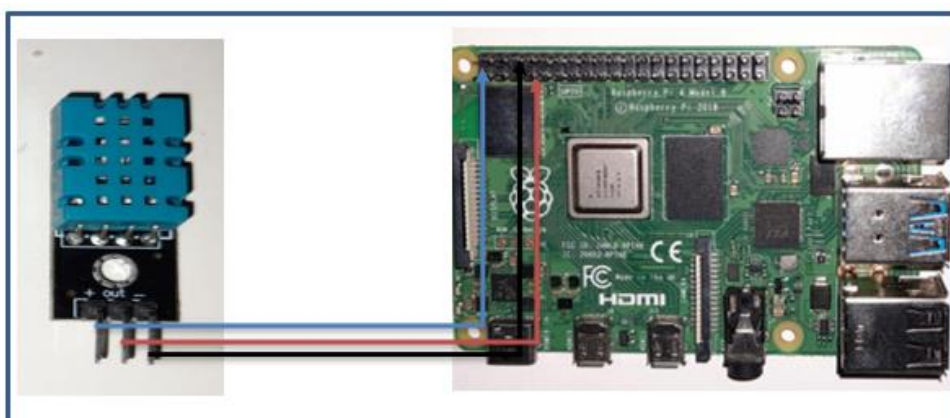



Figura 16. Pines de Sensor de temperatura y humedad DHT11.

Luego de identificar los pines del sensor DHT11 se podrá conectar a los Pines de la GPIO, para ello es necesario conocer su identificación ya que estos pueden ser Board (físicos) o BCM (digitales) según se determine en la programación.

Para ello se tendrá una tabla impresa en formato A4 para que el programador pueda acceder y realizar las conexiones sin temor a equivocarse. Según el código ejecutable del paso 3, se define que los pines GPIO de la placa de Raspberry serán utilizados como entradas digitales (BCM).

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

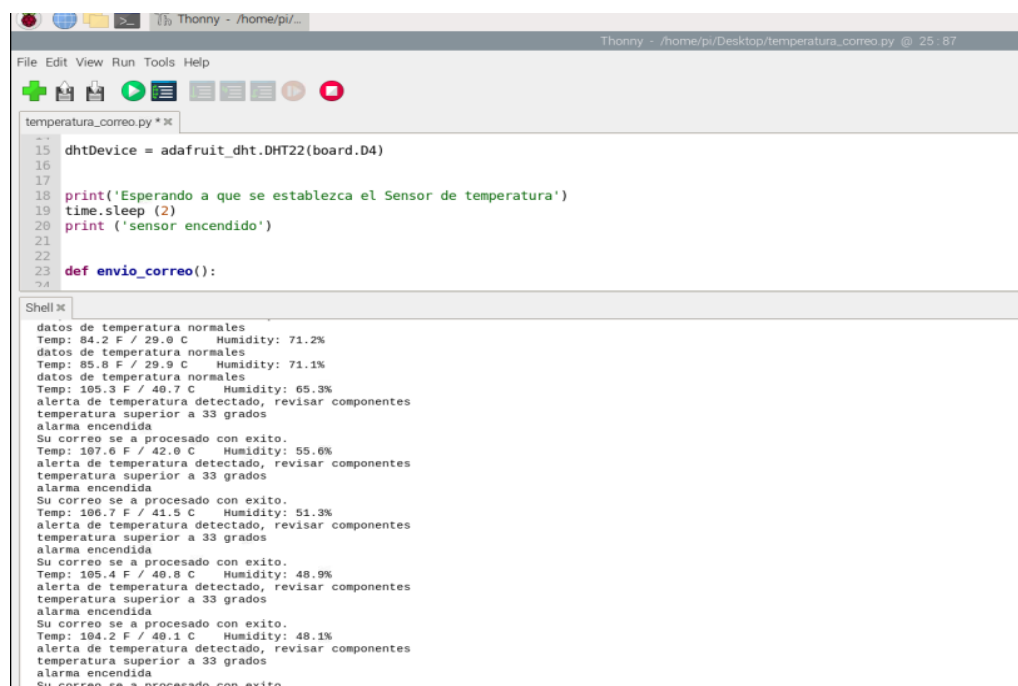
		REVISION 1/1	Página 29 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Por lo cual se debe conectar con la ayuda de un jumper la salida de la señal del sensor DHT (pin central) con la entrada digital4.

Paso 5. Ejecución y visualización de resultados en pantalla.

Luego de realizar las conexiones físicas y crear el código ejecutable en el ambiente virtual de la Raspberry Pi, se debe guardar el archivo y ejecutarlo.

Para el ejemplo se ha guardado el código de ejecución como Temperatura_correo.py el cual lo podemos ejecutar ingresando con la aplicación Thonny Python o por la línea de comandos. El código comenzara a ejecutarse y el sensor a tomar lecturas y a enviarlas al raspberry Pi.



```

15 dhtDevice = adafruit_dht.DHT22(board.D4)
16
17
18 print('Esperando a que se establezca el Sensor de temperatura')
19 time.sleep(2)
20 print('sensor encendido')
21
22
23 def envio_correo():
24

```


```

datos de temperatura normales
Temp: 84.2 F / 29.0 C Humidity: 71.2%
datos de temperatura normales
Temp: 85.8 F / 29.9 C Humidity: 71.1%
datos de temperatura normales
Temp: 105.3 F / 40.7 C Humidity: 65.3%
alerta de temperatura detectado, revisar componentes
temperatura superior a 33 grados
alarma encendida
Su correo se a procesado con exito.
Temp: 107.6 F / 42.0 C Humidity: 55.6%
alerta de temperatura detectado, revisar componentes
temperatura superior a 33 grados
alarma encendida
Su correo se a procesado con exito.
Temp: 106.7 F / 41.5 C Humidity: 51.3%
alerta de temperatura detectado, revisar componentes
temperatura superior a 33 grados
alarma encendida
Su correo se a procesado con exito.
Temp: 105.4 F / 40.8 C Humidity: 48.9%
alerta de temperatura detectado, revisar componentes
temperatura superior a 33 grados
alarma encendida
Su correo se a procesado con exito.
Temp: 104.2 F / 40.1 C Humidity: 48.1%
alerta de temperatura detectado, revisar componentes
temperatura superior a 33 grados
alarma encendida
Su correo se a procesado con exito

```

Figura 17. Resultado del sistema de medición y control de temperatura leyendo señales del ambiente.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:


		REVISION 1/1	Página 30 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Realizamos la prueba de calentar el ambiente mediante una secadora de aire y provocar que el sensor tomé lectura de un sobrecalentamiento del área para que nos envíe la alerta por correo



Figura 18. Disparo de calor al sensor DHT22.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 31 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Al sensor la temperatura superior a los parámetros definidos el sensor envía la lectura al microcontrolador, cuando lo recibe muestra por pantalla el mensaje Alarma encendida, temperatura elevada. Y mediante el protocolo SMTPlib envía un mensaje de formato MIMEtext al correo designado con el mensaje de alarma de seguridad hasta el correo repciondatacenter@gmail.com.

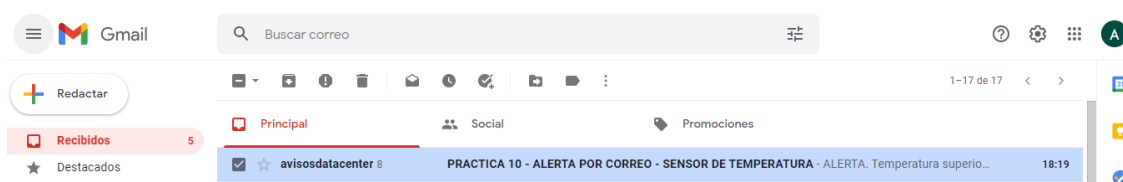



Figura 19. Recepción de correo electrónico de alarma de temperatura.

4.3. ARDUINO COMO SISTEMA DE CONTROL DE ACCESO Y ALERTA POR CORREO ELECTRÓNICO.

El procedimiento para realizar un control de acceso mediante uso de Arduino, es reconocer los funcionamientos de que cada equipo, a que voltaje trabaja para no producir daños en ellos, y realizar la programación y conexiones de la manera correcta, para una mejor operatividad.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 32 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

Paso 1: Verificar las entradas y salidas del Arduino.

Es fundamental conocerlas para empezar a programar Arduino. En el Arduino UNO existen entradas y salidas analógicas (6) y digitales (16). Por un lado, las analógicas son utilizadas, generalmente, para dispositivos de entrada y nos permiten leer un rango de valores. En cambio, las digitales pueden tener a la entrada o a la salida 0V o 5V, esto nos indica que ese pin está en estado bajo (Low) o alto (High).

Paso 2: Conexiones de los dispositivos.

Procedemos a verificar que entradas se va a utilizar, cablear con jumpers hembra-macho la tarjeta RFID hacia el Arduino, luego conectar como alarma de sonido y visión un buzzer y leds como indicadores, para seguir con el uso de un solenoide que trabaja con una fuente de 12V y un módulo relé, el cual se activa al pasar la tarjeta o llavero de identificación que podrá brindar el acceso y la chapa eléctrica se abrirá, en caso de error envía una alarma mediante mensaje de texto y correo electrónico


Paso 3: Programación

Después de las conexiones, proceder a realizar la programación mediante el uso del programa de Arduino, para comenzar se debe descargar la librería MFRC522, seguir con la codificación con uso de parámetros establecidos por parte de cliente como por ejemplos números de tarjetas a usar, tiempo de espera para que no refleje error e intruso.

Paso 3.1. Iniciamos en incluir las librerías a usar.

Comenzamos a programar, debemos definir las librerías de los dispositivos a usar, para que se pueda efectuar el proceso.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 33 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```
#include "Arduino.h"
#include <EMailSender.h>
#include <ESP8266WiFi.h>
#include <SPI.h>
#include <MFRC522.h>
#include <SoftwareSerial.h>
#define rxPin D0
#define txPin D1
SoftwareSerial sim(rxPin,txPin);
int _timeout;
String _buffer;
String numero = "+593991338815";
```

#Declaramos las librerías y número a utilizar

Paso 3.2. Definir entradas a utilizar

```
constchar* ssid = "Micke";
constchar* password = "Dennis94";


uint8_t connection_state = 0;
uint16_t reconnect_interval = 10000;

#define SS_PIN D4
#define RST_PIN D3
#define RELAY D8
#define BUZZER D2

#define ACCESS_DELAY 2000

#define DFNIFD DFLAY 1000
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 34 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		


Paso 3.3. Declarar función de conexión con modulo Wifi ESP

```

MFRC522 mfrc522(SS_PIN, RST_PIN);
EMailSenderemailSend("avisosdatacenter@gmail.com", "raspberrypi4");
uint8_t WiFiConnect(constchar* nSSID = nullptr, constchar* nPassword = nullptr)
{
static uint16_t attempt = 0;
Serial.print("Connectingto ");
if(nSSID) {
WiFi.begin(nSSID, nPassword);
Serial.println(nSSID); }
uint8_t i = 0;
while(WiFi.status() != WL_CONNECTED && i++ < 50)
{ delay(200);
Serial.print(".");
}
++attempt;
Serial.println("");
if(i == 51) {
Serial.print("Connection: TIMEOUT onattempt: ");
Serial.println(attempt);
if(attempt % 2 == 0)
Serial.println("Checkifaccesspointavailableor SSID and Password\r\n");
return false;
}
Serial.println("Connection: ESTABLISHED");
Serial.print("Got IP address: ");
Serial.println(WiFi.localIP());
return true;
}

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 35 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```

Serial.println("Connection: ESTABLISHED");

Serial.print("Got IP address: ");

Serial.println(WiFi.localIP());

return true;

```


Paso 3.4. Creamos una variable de espera elocuente para la ejecución del módulo WIFI

```

voidAwaits()
{
  uint32_t ts = millis();
  while(!connection_state)
  {
    delay(50);
    if(millis() > (ts + reconnect_interval) && !connection_state){
      connection_state = WiFiConnect();
      ts = millis();
    }
  }
}

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 36 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		


Paso 3.5. Declarar variables para la ejecución del programa

```
voidsetup()
{
  Serial.begin(115200);
  _buffer.reserve(50);
  sim.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  pinMode(RELAY, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  noTone(BUZZER);
  digitalWrite(RELAY,LOW);
  connection_state = WiFiConnect(ssid, password);
  Serial.println("Ingrese tarjeta");
  Serial.println();
}
```

Paso 3.6. Creamos las líneas de comando de ejecución

```
voidloop()
{if(! mfrc522.PICC_IsNewCardPresent()){
  return;}
if(! mfrc522.PICC_ReadCardSerial()){
  return;
}
}
```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:


		REVISION 1/1	Página 37 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```

Serial.print("Tag");
Stringcontent="";
byte letter;
for(byte i=0; i < mfrc522.uid.size; i++){
Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0": " ");
Serial.print(mfrc522.uid.uidByte[i], HEX);
content.concat(String(mfrc522.uid.uidByte[i]<0x10 ? " 0" : " "));
content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
Serial.println();
Serial.print("Mensaje:");
content.toUpperCase();
if (content.substring(1) == "29 74 C2 98"){
Serial.println("Acceso autorizado, Bienvenido");
Serial.println();
delay(500);
digitalWrite(RELAY, HIGH);
delay(ACCESS_DELAY);
digitalWrite(RELAY, LOW);
} else {
Serial.println("Acceso Denegado");
Serial.println("Enviando SMS....");
Serial.println("Enviando E-MAIL....");
sim.write("AT+CMGF=1\r\n");
delay(200);
sim.println("AT+CMGS=\""+numero+"\"");
delay(200);
Stringsms="Error, Acceso Denegado...Verificar";
sim.println(sms);
delay(100);
sim.println((char)26);
delay(200);
EnviarMensajeGMAIL();
//delay(200);

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 38 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```

sim.write("AT+CMGF=1\r\n");
delay(200);
sim.println("AT+CMGS=\"" + numero + "\"\r");
delay(200);
Stringsms="Error, Acceso Denegado...Verificar";
sim.println(sms);
delay(100);
sim.println((char)26);
delay(200);
EnviarMensajeGMAIL();

//delay(200);

```

Paso 3.7. Ejecutamos función para envío del mail.


```

voidEnviarMensajeGMAIL (void)
{
if(!connection_state) // ifnotconnectedto WIFI
Awaits(); // constantlytryingtoconnect
EMailSender::EMailMessagemessage;
message.subject = "ESTADO DE ALARMA";

message.message = "ALERTA INTRUSO<br>Usuario no registrado<br>se requiere de

```

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 39 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

```

EMailSender::Response
resp = emailSend.send("repciondatacenter@gmail.com", message);

Serial.println("Sendingstatus: ");

Serial.println(resp.status);
Serial.println(resp.code);
Serial.println(resp.desc);
Serial.println("MENSAJE ENVIADO EXITOSAMENTE");

```

Cargamos el archivo y hacemos pruebas con las tarjetas y llaveros del módulo RFID.

Cuando pasemos la tarjeta registrada nos permitirá el acceso y activará el solenoide para la apertura de la puerta, cuando pasemos la llave no registrada nos enviará el mensaje de alerta el correo.




Figura 20. Recepción de correo electrónico de alarma de temperatura.

5) RECURSOS UTILIZADOS

- Kit Raspberry Pi.
- Conector alimentación 5V 3^a.
- Tarjeta MicroSD.
- Laptop.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 40 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

- WI-FI.
- Protocolo VNC.
- Sensor PIR HC-SR501
- Buzzer.
- Luces Led.
- Thonny Python.
- Cámara HD.
- Protocolo SMTP.
- Jumper.
- Clase SMTPlib.
- Arduino Uno
- Modulo ESP8266WiFi


6) RESULTADOS OBTENIDOS

Mediante las prácticas en Raspberry se ha podido demostrar la funcionalidad de los sistemas de alarmas mediante el protocolo SMTP en la clase SMTPlib para Python.

El cual es de mucha utilidad en caso de querer monitorear un área de manera remota

Mediante esta práctica se obtiene como resultado principal el uso y manejo de dispositivos como seguridad, que permiten usar diferentes parámetros de acuerdo con la necesidad del área que lo requiera, teniendo dos tipos de alarmas mediante mensaje de texto y correo electrónico.

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1	Página 41 de 41
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	TELECOMUNICACIONES		
CARRERA	INGENIERÍA ELECTRÓNICA		
SEDE	GUAYAQUIL		

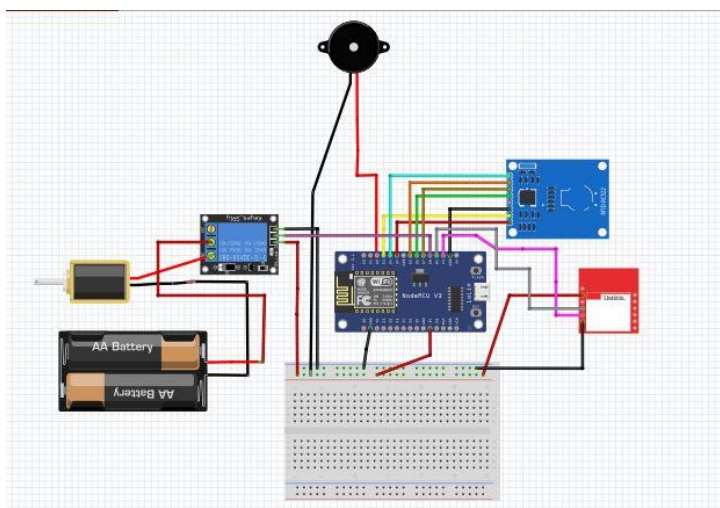


Figura 21. Conexión de sensor RFID con modulo WIFI ESP8266WiFi

7) CONCLUSIONES

Con la finalidad de implementar sensores que permitan alarmar remotamente en caso de anomalías y falsos accesos se puede encontrar al protocolo SMTPlib como la principal herramienta para notificaciones inalámbricas instaladas en el sistema de seguridad creado. Mediante su implementación se asegura que el sistema de seguridad tendrá un respaldo remoto en caso de que no exista personal de vigilancia en los puestos de monitoreo.

8) BIBLIOGRAFÍA UTILIZADA

<https://projects.raspberrypi.org/en/projects/physical-computing/11>

<https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/7>

<https://www.raspberrypi.org/forums/viewtopic.php?t=49015>

<https://www.raspberrypi.org/forums/viewtopic.php?t=284519>

<https://projects.raspberrypi.org/en/projects/physical-computing/11>

Elaborado por: Andrés Vera Cando Henry Zamora Mendoza	Revisado por: PhD Lenín Cevallos	Aprobado por: Ing. Orlando barcia
Fecha de Elaboración 21/07/21	Fecha de Revisión	Número de Resolución Consejo de Carrera: