



**UNIVERSIDAD ESTATAL  
PENÍNSULA DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

**TECNOLOGÍAS DE LA INFORMACIÓN**

**EXAMEN COMPLEXIVO**

Componente Práctico, previo a la obtención del Título de:  
**INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

**Implementación de sistema de seguridad para el control de acceso a  
aulas del Bloque 1 de la Facultad de Sistemas y Telecomunicaciones  
mediante el uso de lectores QR y arduino.**

**AUTOR**

**GARY ADONIS GONZÁLEZ CASTILLO**

**LA LIBERTAD – ECUADOR**

**2021**

## APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de componente práctico del examen de carácter complejo: **“Implementación de sistema de seguridad para el control de acceso a aulas del Bloque 1 de la Facultad de Sistemas y Telecomunicaciones mediante el uso de lectores QR y arduino.”**, elaborado por el Sr. González Castillo Gary Adonis, de la carrera de Tecnología de la Información de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes.

La Libertad, 20 de septiembre de 2021.



.....  
Lsi. Daniel Quirumbay Yagual

## DECLARACIÓN

El contenido del presente componente práctico del examen de carácter complejo es de mi responsabilidad; el patrimonio intelectual del mismo pertenece a la Universidad Estatal Península de Santa Elena.

A handwritten signature in blue ink, appearing to be 'Gary Adonis González Castillo', written in a cursive style.

.....  
Gary Adonis González Castillo

## **AGRADECIMIENTO**

Le agradezco a Dios por haberme permitido llegar hasta este punto de mi vida, por el amor transmitido mediante las personas que me rodearon en el camino, por sus señales claras en el momento preciso para brindarme ese aliento necesario y nunca darme por vencido.

Le doy gracias a mis padres, Sinforiano y Juana, quienes han sido el motor principal de mi motivación en cada proyecto emprendido, sus consejos y enseñanzas que nunca se hicieron esperar, pero sobre todo por ese cálido apoyo incondicional.

A mis hermanos, que de una u otra forma estuvieron presentes en todo este proceso, sus bromas y sus palabras de aliento las llevo conmigo cada día.

A mi abuelo, quien, pese a no estar presente ya en este mundo, nunca dejó de transmitirme su fortaleza, cada recuerdo de su forma de ser, fue inspiración para no decaer.

A mis tíos, Fernando y Mercedes quienes me brindaron un espacio en su hogar para descansar y un plato de comida cuando se hacía complicado viajar a casa. Pese al pronto fallecimiento de mi tío Fernando, quiero plasmar mi agradecimiento especial hacia el por compartir conmigo una enseñanza de vida que hasta el día de hoy pongo en práctica.

A mis amigos, que no dudaron en brindarme su apoyo cuando lo necesité e hicieron del proceso de aprendizaje, uno más divertido y ameno. De manera especial agradezco a mi amiga, Angie, quien pese a ser parte de este proyecto, su apoyo desinteresado hizo posible que se llegase a la meta.

Finalmente, agradezco a todos aquellos docentes con el que compartí un aula, por haber puesto todos sus conocimientos, paciencia, apoyo y dedicación en cada hora de clase.

***Gary González Castillo.***

## **DEDICATORIA**

Éste trabajo se lo dedico a todas las personas que me apoyaron, pero en especial se lo dedico a mi madre, quién ha sido, es y será la mujer más noble de mi vida, el motivo de mi inspiración, la fuente de amor por la cual hoy lucho y me mueve ser mejor persona cada día. Aquella mujer que me llena de orgullo y que con una bendición al salir de casa me arma de valor para enfrentar al mundo.

***Gary González Castillo.***

**TRIBUNAL DE GRADO**



Ing. Washington Torres Guin, Mgti.

**DIRECTOR(E) DE LA CARRERA DE  
TECNOLOGÍAS DE LA INFORMACIÓN**



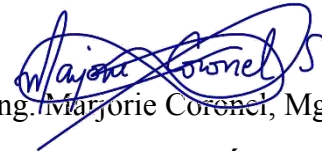
Ing. Iván Coronel Suárez, MSIA.

**DOCENTE ESPECIALISTA**



Lsi. Daniel Quirumbay.

**DOCENTE TUTOR**



Ing. Marjorie Coronel, Mgti.

**DOCENTE GUÍA UIC**

## RESUMEN

El presente trabajo se desarrolla para el Bloque 1 de la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, el cual está estructurado por dos plantas divididas en nueve secciones respectivamente, cinco en la parte inferior, y cuatro en la parte superior, de las cuales tres son aulas, empleadas para impartir clases a estudiantes o realizar diversas actividades con fines educativos.

Dichas salas de clases, tienen la necesidad de integrar nuevas tecnologías para la seguridad, ya que cuentan con una cerradura común y corriente que se apertura con el uso de llaves físicas, que, en ciertos casos, puede haber pérdida o clonación de las mismas. Por consiguiente, el encargado debe mantener las llaves bajo custodia, de tal manera que los usuarios que empleen las aulas deban esperar a que den apertura para luego emplearla.

En base a la problemática mencionada anteriormente, se propone realizar un control en las aulas de clases del Bloque 1 de la Facultad de Sistemas y Telecomunicaciones, que comprende el diseño y la implementación de un sistema de seguridad basado en Arduino, adicionando una cerradura eléctrica, que se apertura mediante el uso de códigos QR. La comunidad universitaria se beneficia con este sistema, ya que se brinda la seguridad pertinente y un ágil acceso a las aulas.

La metodología de investigación que se utiliza en el presente informe, es de tipo exploratoria y diagnóstica, las cuales fueron útiles para la recolección de información, a fin de identificar los procedimientos que se llevan a cabo en cuanto al control y la seguridad de las aulas del Bloque 1, a través de la técnica de observación y entrevista al técnico docente. Por otro lado, se empleó la metodología de tipo incremental en el desarrollo del hardware y software, que logra una rápida adaptación a los cambios y permite fusionar los flujos de trabajo.

Como resultado, se realizaron las pruebas de funcionalidad del sistema de seguridad, demostrando que los usuarios que lo emplean acceden a las aulas de clases de una manera más ágil y segura.

## TABLA DE CONTENIDO

1. FUNDAMENTACIÓN	16
1.1. ANTECEDENTES	16
1.2. DESCRIPCIÓN DEL PROYECTO	19
1.3 OBJETIVOS DEL PROYECTO	22
1.3.1 OBJETIVO GENERAL	22
1.3.2 OBJETIVOS ESPECÍFICOS	22
1.4 JUSTIFICACIÓN	23
1.5 ALCANCE DEL PROYECTO	25
2. MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO	26
2.1 MARCO TEÓRICO	26
2.1.1. AHORRO ECONÓMICO PARA LA TECNOLOGÍA CIENTÍFICA DE CÓDIGO ABIERTO Y LIBRE: UNA REVISIÓN.	26
2.1.2. CONTROL DE ACCESO FÍSICO BASADO EN EL CÓDIGO QR.	27
2.2 MARCO CONCEPTUAL	28
2.2.1. ARDUINO UNO	28
2.2.2. IDE ARDUINO	29
2.2.3. LECTOR 2D QR - DP8405	29
2.2.4. RELÉ 5V 1 CANAL	31
2.2.5. SHIELD ETHERNET W5100	31
2.2.6. VISUALIZADOR LCD	32
2.2.7. C++	33
2.2.8. PHP	33
2.2.9. FUSION 360	34
2.2.10. PROTOCOLO HTTP	34
2.2.11. SPI	35
2.2.12. ESTÁNDAR I2c	35
2.2.13. MICROCONTROLADOR	36
2.3 METODOLOGÍA DEL PROYECTO	38
2.3.1 METODOLOGÍA DE INVESTIGACIÓN	38
2.3.2 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN	38
2.3.3 METODOLOGÍA DE DESARROLLO DEL PROYECTO	40
3. PROPUESTA	43



3.1. REQUERIMIENTOS	43
3.1.1. REQUERIMIENTOS FUNCIONALES	43
3.1.2. REQUERIMIENTOS NO FUNCIONALES	44
3.2. COMPONENTES DE LA PROPUESTA	45
3.2.1. ARQUITECTURA DEL SISTEMA	45
3.2.2. DIAGRAMA DE BLOQUES DE COMPONENTES	45
3.2.3. DIAGRAMA ESQUEMÁTICO DE COMPONENTES	54
3.2.4. DIAGRAMA ESQUEMÁTICO DE RED	55
3.2.5. DIAGRAMA ESQUEMÁTICO DE CONEXION ELÉCTRICA	56
3.2.6. DIAGRAMA DE ACTIVIDAD	57
3.2.8. DIAGRAMA DE BLOQUES GENERAL – PROGRAMACIÓN DEL SISTEMA	60
3.2.9. PROGRAMACION DEL MICROCONTROLADOR	61
3.2.10. MODELADO DE DATOS DEL SISTEMA EN GENERAL	72
3.2.11. MODELADO DE DATOS – DISPOSITIVO DE SEGURIDAD	73
3.7. DISEÑO Y CONSTRUCCIÓN DEL CONTENEDOR DEL DISPOSITIVO	85
3.4. PRUEBAS	90
CONCLUSIONES	94
RECOMENDACIONES	95
BIBLIOGRAFÍA	96
ANEXOS	99

## ÍNDICE DE FIGURA

Figura 1. Diagrama de pines del Arduino Uno	29
Figura 2. Vista posterior del lector QR DP8405	30
Figura 3. Vista frontal del lector QR DP8405	30
Figura 4. Módulo relay 5DC 1CH	31
Figura 5. Módulo de ethernet W5100	32
Figura 6. LCD 2x16	33
Figura 7. Esquema de funcionamiento Php	33
Figura 8. SPI Bus: Un maestro 2 esclavos	35
Figura 9. Ejemplo de una configuración del bus I2c	36
Figura 10. Modelo incremental del Sistema - Hardware y Software.	42
Figura 11. Arquitectura del sistema	45
Figura 12. Diagrama de bloques de los elementos.	46
Figura 13. Diagrama de conexiones de DP8405 con Arduino.	47
Figura 14. Diagrama de conexiones de Ethernet Shield W5100 con Arduino.	49
Figura 15. Diagrama de conexión I2C LCD	50
Figura 16. Diagrama de conexión cerradura eléctrica mediante Relé.	51
Figura 17. Diagrama final de conexiones del sistema	53
Figura 18. Diagrama esquemático de componentes.	54
Figura 19. Diagrama esquemático básico de red.	55
Figura 20. Diagrama conexión switch – dispositivo de seguridad.	55
Figura 21. Diagrama básico - conexión eléctrica.	56
Figura 22. Diagrama esquemático - instalación de cableado eléctrico.	56
Figura 23. Diagrama de actividad del sistema de seguridad.	57
Figura 24. Diagrama de proceso validación de código QR	58
Figura 25. Diagrama de proceso para adquirir dirección IP por DHCP	59
Figura 26. Diagrama de bloques general – Programación del sistema	60
Figura 27. Librerías Módulo de comunicación	61
Figura 28. Inicialización del módulo Ethernet	62
Figura 29. Adquisición de dirección IP por DHCP	62
Figura 30. Declaración de variables para el uso del lector QR	63
Figura 31. Librerías utilizadas para el módulo de alertas	63
Figura 32. Declaración de variable LiquidCrystal_I2c	63

Figura 34. Inicialización de la pantalla LCD	64
Figura 35. Variables globales para el uso de millis ()	64
Figura 36. Estructuración general del loop ()	65
Figura 37. Código de ejecución si se detecta código QR	66
Figura 38. Función de control para el lector QR	67
Figura 39. Función petición servidor	67
Figura 40. Función respuesta servidor	68
Figura 41. Función para el accionar de la cerradura eléctrica	69
Figura 42. Código de ejecución petición constante	70
Figura 43. Función estados pantalla Lcd	71
Figura 44. Modelado de datos del sistema en General	72
Figura 45. Modelado de datos – Dispositivo de seguridad	73
Figura 46. Vista frontal, parte superior del diseño en 3D	85
Figura 47. Vista lateral, parte superior del diseño en 3D	85
Figura 48. Vista posterior, parte superior del diseño en 3D	86
Figura 49. Medidas de la vista posterior, parte superior del diseño en 3D	86
Figura 50. Medidas de la vista superior, parte superior del diseño en 3D	87
Figura 51. Medidas de la vista lateral, parte inferior del diseño en 3D	87
Figura 52. Medidas de la vista frontal, parte superior del diseño en 3D	88
Figura 53. Medidas de la vista lateral, parte superior del diseño en 3D	88
Figura 54. Medidas de la vista superior, parte superior del diseño en 3D	88
Figura 55. Impresión en 3D de la parte inferior del diseño	89
Figura 56. Impresión en 3D de la parte superior del diseño	89
Figura 57. Ensamblaje de componentes electrónicos y estructura en 3D.	113
Figura 58. Instalación del dispositivo de seguridad a un costado de la puerta de acceso.	113
Figura 60. Dispositivo final en funcionamiento	113
Figura 59. Conexión e instalación de cerradura eléctrica	113

## ÍNDICE DE TABLA

Tabla 1. Especificaciones técnicas de la placa Arduino Uno. [14]	28
Tabla 2. Especificaciones técnicas del lector 2D QR CODE. [17]	30
Tabla 3. Especificaciones técnicas Relé 5V 1 canal [19]	31
Tabla 4. Especificaciones técnicas del módulo de ethernet W5100 [20]	32
Tabla 5. Especificaciones técnicas LCD 2x16 [21]	32
Tabla 6. Formato URL [25]	34
Tabla 7. Requerimientos funcionales	44
Tabla 8. Requerimientos no funcionales	44
Tabla 10. Comparación de módulos Ethernet	48
Tabla 11. Comparativa de módulos LCD	49
Tabla 12. Comparativa de cerraduras eléctricas.	51
Tabla 13. Comparativa de microcontroladores Arduino	52
Tabla 14. Diccionario de datos tabla accesos	74
Tabla 15. Diccionario de datos tabla asignación_dm	74
Tabla 16. Diccionario de datos tabla aulas	74
Tabla 17. Diccionario de datos tabla avisos	75
Tabla 18. Diccionario de datos tabla bloques	75
Tabla 19. Diccionario de datos tabla carreras	75
Tabla 20. Diccionario de datos tabla códigos	76
Tabla 21. Diccionario de datos tabla copia_asignacion	76
Tabla 22. Diccionario de datos tabla copia_sesion	76
Tabla 23. Diccionario de datos tabla cs	77
Tabla 24. Diccionario de datos tabla detalles_dh_dm	77
Tabla 25. Diccionario de datos tabla detalle_dh	78
Tabla 26. Diccionario de datos tabla detalle_sesion	78
Tabla 27. Diccionario de datos tabla dias	78
Tabla 28. Diccionario de datos tabla docentes	78
Tabla 29. Diccionario de datos tabla equipos	79
Tabla 30. Diccionario de datos tabla facultades	79
Tabla 31. Diccionario de datos tabla horarios	79
Tabla 32. Diccionario de datos tabla horas	80
Tabla 33. Diccionario de datos tabla lecturas	80

Tabla 34. Diccionario de datos tabla materias	80
Tabla 35. Diccionario de datos tabla menús	81
Tabla 36. Diccionario de datos tabla motivos	81
Tabla 37. Diccionario de datos tabla paralelos	81
Tabla 38. Diccionario de datos tabla periodos	81
Tabla 39. Diccionario de datos tabla personas	82
Tabla 40. Diccionario de datos tabla peticiones	82
Tabla 41. Diccionario de datos tabla presidentes	82
Tabla 42. Diccionario de datos tabla roles	83
Tabla 43. Diccionario de datos tabla semestres	83
Tabla 44. Diccionario de datos tabla sesiones	83
Tabla 45. Diccionario de datos tabla tablero_dh	84
Tabla 46. Diccionario de datos tabla usuarios	84
Tabla 47. Prueba 1 de funcionamiento para validación de código QR	90
Tabla 48. Prueba 2 de funcionamiento para validación de código QR	91
Tabla 49. Prueba 3 de funcionamiento para validación de código QR	91
Tabla 50. Prueba 4 de funcionamiento para validación de código QR	92
Tabla 51. Prueba 5 de funcionamiento para validación de código QR	93
Tabla 52. Prueba 6 de funcionamiento llenar tabla lectura	93

## **LISTA DE ANEXOS**

Anexo 1. Entrevista al técnico docente	100
Anexo 2. Registro de la técnica de observación aplicada en el Bloque 1 de la Facultad de Sistemas y Telecomunicaciones	102
Anexo 3: Código del proyecto (microcontrolador)	103
Anexo 4: Código del proyecto (Web service - Php)	110
Anexo 5: Implementación	113

## INTRODUCCIÓN

Hoy en día las instituciones educativas, enfatizan la importancia de emplear nuevas tecnologías de seguridad en los lugares del establecimiento, protegiendo los activos físicos que poseen. Estos sistemas de seguridad y vigilancia, se realizan mediante equipos que facilitan el resguardo de los mismos, los cuales contribuyen a la agilización de los procesos de control de acceso, brindando un mayor grado de comodidad, tranquilidad y agilidad.

La Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, presenta la necesidad de incluir nuevas tecnologías aplicadas a la seguridad en las aulas del Bloque 1, las cuales se encuentran ubicadas en la segunda planta. Teniendo en cuenta la problemática que surge al controlar el acceso a las aulas, esperar al encargado de las llaves, dar apertura a las puertas y riesgo de pérdida de equipos.

Por esta razón, se propone el diseño e implementación de un sistema de seguridad electrónica, destinado a la automatización de los procesos que conllevan a la accesibilidad, brindando seguridad a las aulas de clases del lugar antes mencionado en la Universidad.

El sistema está dividido en los siguientes módulos: En el primer módulo se especifican de manera puntual las entradas del sistema, así como sus conexiones y modos de funcionamiento, como por ejemplo: el lector QR o el Ethernet Shield, de la misma manera se procede para el segundo módulo, con la diferencia de que se dirige al tratamiento de las salidas del sistema, el módulo tres, en cambio, se centra en el procesamiento de las señales antes mencionadas, aquí interviene toda la programación, como último módulo está la implementación y pruebas en tiempo real, en dónde se llevará a cabo la instalación del dispositivo en las respectivas aulas, cabe mencionar que en este punto, el cableado eléctrico y de red respectivamente son de suma importancia para el correcto desenvolvimiento del dispositivo.

La metodología de investigación empleada para la realización del presente informe es de tipo exploratoria y diagnóstica, las cuales fueron esenciales en la recolección de información y elaboración los requerimientos del sistema. Para el desarrollo, se emplea el modelo incremental, tomando en cuenta su adaptabilidad entorno al proyecto.

Se utilizó la herramienta Arduino para el desarrollo del proyecto y lenguaje de programación C. Así mismo, para administrar la base de datos, se usa MySQL como sistema gestor de base de datos y para diagramar los esquemáticos se empleó el programa Fritzing.

El componente de arquitectura se basa en el modelo Cliente – Servidor, realizando peticiones a través de HTTP Request, interactuando con Arduino y PHP.

Con la aplicación de las pruebas funcionales del sistema de seguridad, se demostró que se mejoraron considerablemente los procesos de control de acceso, minimizando riesgos de pérdidas de equipos dentro de las aulas, y brindando seguridad a los beneficiarios.

El capítulo I, abarca los antecedentes de la institución, los inconvenientes existentes, la descripción, objetivos, justificación y alcance del proyecto.

El capítulo II, encierra el marco teórico, marco conceptual y la metodología de investigación, aplicada para la elaboración del proyecto.

El capítulo III expone la propuesta en la que se detallan los requerimientos funcionales y no funcionales, sus componentes, la arquitectura del sistema, los módulos empleados en el desarrollo, realización de esquemáticos, estructuración del código fuente, diagramas de procesos, conclusiones y recomendaciones

## **1. FUNDAMENTACIÓN**

### **1.1. ANTECEDENTES**

Las instalaciones de las instituciones educativas son especiales por el propósito al que están destinadas, sin embargo, a diferencia de la mayoría de construcciones habitacionales y de servicios, estas albergan durante gran parte del día y por lapsos prolongados a decenas e incluso centenas de personas, siendo el caso de estudiantes y personal académico [1]. En su mayoría las entidades públicas y privadas tienden a ser controladas por un sistema de acceso de algún área determinada, con el fin de llevar un control tanto de empleados como recursos de una empresa. Determinar el control de accesos, es importante, ya que así se logra proteger de alguna manera los recursos físicos, evitando daños, pérdidas, negligencia o cualquier otro aspecto que pudiera afectar directamente al sitio. [2]

La Universidad Estatal Península de Santa Elena, se encuentra ubicada en la Av. Principal, del cantón La Libertad, perteneciente a la provincia de Santa Elena. Actualmente tiene 23 años laborando, en conjunto con un equipo de profesionales con experiencia en temas educacionales [3]. Conformada por las siguientes autoridades:

**Rector:** Ing. Néstor Acosta Lozano, Ph.D.

**Vicerrectora:** Lcda. Maritza Gisella Paula Chica, Ph.D.

Dicha institución tiene como misión, formar profesionales competentes, comprometidos con la sociedad y el ambiente, sobre la base de una elevada calidad académica, la investigación, la adopción y generación de conocimientos científicos y tecnológicos. [4]

Mediante la entrevista realizada al técnico docente (**Anexo 1**) se logra rescatar que para poder verificar que un aula de clases se encuentra desocupada y respectivamente asegurada, es necesario trasladarse puerta a puerta hacia los respectivos salones tanto del piso inferior como superior, lo cual sin duda requiere de un esfuerzo físico y pérdida de tiempo. Una forma de controlar que aula debe estar disponible o no para su uso es el horario de clases, un sustento poco eficaz, ya que las aulas que se encuentran desocupadas suelen ser invadidas por estudiantes o docentes, sin previo aviso.



Pese a que es de responsabilidad de todo el personal educativo, asegurar y cuidar los bienes materiales que comprenden un aula de clases, a la vez es una desventaja en caso de que existan pérdidas materiales no poder contar con un responsable, debido a que el sistema de seguridad actual es rudimentario y no existe un control de quienes ingresan o salen de las aulas.

A través del método de observación (**Anexo 2**), se determinó, que, en varias ocasiones los estudiantes se atribuyen el uso de las aulas por cuenta propia, y que en ciertos extremos usan sus propios medios para abrir las puertas, las cuales sin negación alguna y por su seguridad obsoleta son abiertas con éxito fácilmente.

A nivel mundial existen diversos dispositivos que permiten el aseguramiento de domicilios e instituciones en cuanto a puertas, siendo el caso de cerraduras con acceso a través de código o pin, sensores biométricos, tarjeta electrónica, y también el uso de código QR, que es ya un proyecto en marcha por parte de Lockart quién describe su producto como una cerradura inteligente que permite abrir el hogar sin necesidad de llaves. La principal característica es la seguridad que además se enfoca en mejorar el día a día y la calidad de vida de las personas. La cerradura inteligente genera un código de acceso para abrir la puerta, dicho código será de un solo uso, es decir, una vez utilizado, el mismo no se puede usar de nuevo para abrir la puerta. [5]

Por otro lado, Altiria TIC es una empresa española dedicada a brindar servicios de comunicaciones a través de SMS, además ofrece un sistema de control de acceso con códigos QR y aplicación móvil que permiten gestionar la entrada a cualquier recinto. Se emplea para ello un código QR único que identifica a cada persona. Los sistemas ofrecidos por Altiria resultan idóneos para eventos, registro de jornada laboral, congresos, ferias, convenciones, acceso a condominios, entrada a aulas, ingreso de estudiantes a academias, reuniones y cualquier acto que tenga lugar en un recinto cerrado. Se pueden registrar tanto entradas como salidas. [6]

De forma local, en la Universidad Estatal Península de Santa Elena, se implementó un proyecto como propuesta tecnológica, que fue destinado a los laboratorios 1, 2, 3 de Informática y laboratorios de Cisco 1 y 2. Principalmente cubre el acceso y brinda reportes sobre las visitas a dichos laboratorios mediante dispositivos biométricos, los cuales son administrados por el sistema ZKAccess. [7].

Los trabajos antes citados, fueron tomados como base de este proyecto, pese a su similitud en cuanto a la seguridad de una habitación en específico. Sin embargo, son dispositivos de seguridad prefabricados y de elevados costos.

Para concluir, una vez revisado los proyectos anteriores, se logra determinar que, por efecto de presupuesto (**Anexo 4**), y la originalidad que requiere el sistema, este proyecto está orientado al desarrollo de un sistema propio con características personalizadas, para el aseguramiento de aulas, mediante una cerradura que permitirá el acceso a través de código QR.

## 1.2. DESCRIPCIÓN DEL PROYECTO

Mediante este proyecto se propone dar soluciones que comprendan una mejoría notable en los procesos que se llevan a cabo para comprobar la disponibilidad de aula, y que aporten al control de acceso en las instalaciones del bloque 1, perteneciente a la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena (UPSE). Para lograr dichas soluciones, es necesario la cobertura de tres secciones distintas:

- **Primera sección:** Aplicativo web generando códigos QR en el ingreso de información de docentes y estudiantes en la UPSE – Facultad de Sistemas y Telecomunicaciones.
- **Segunda sección:** Aplicación móvil para solicitar aulas disponibles con respuesta QR en el Bloque 1 de la Facultad de Sistemas y Telecomunicaciones en la UPSE.
- **Tercera sección:** Sistema eléctrico para el control de acceso a aulas del Bloque 1 de la Facultad de Sistemas y Telecomunicaciones mediante el uso de Arduino y lectores de códigos QR.

Durante este trabajo se detallará la tercera sección, la cual comprende desde el diseño a la implementación de un sistema de seguridad electrónica personalizada, que permitirá el ingreso a las aulas de clases a todos aquellos quienes conforman el sistema educativo y que además cuenten con el respectivo código de acceso QR.

El funcionamiento del sistema inicia desde la lectura del código QR, dicho código se comprueba mediante una consulta hacia el servidor de base de datos donde se encuentran almacenados los códigos de acceso QR, si el código leído es el correcto, el sistema emitirá la señal respectiva hacia la pantalla Lcd y a la cerradura electrónica respectivamente para que se accione, de tal forma que permita el acceso al aula de clase, en caso de que el código no sea validado, se emitirá un mensaje por pantalla indicando que el código es incorrecto y por lo posterior no se dará el acceso al aula de clase.

El desarrollo del sistema se llevará a cabo considerando de forma general 4 módulos, los cuales comprenden: Módulo de entradas del sistema (lector 2D QR, Ethernet), Módulo

de salidas del sistema (Relé, Pantalla LCD), Módulo procesamiento (Microcontrolador) y el Módulo de implementación (instalaciones eléctricas y de red).

La solución planteada, contribuirá de forma significativa a la seguridad de las aulas de clases y su respectivo control de acceso, además de la agilización de los procesos físicos al momento de solicitar las llaves al encargado, contribuyendo a la movilidad y comodidad del usuario.

Para el diseño, la creación e implementación del sistema, se utilizarán las siguientes herramientas tecnológicas:

## **Hardware**

### **Componentes y accesorios.**

- Arduino Uno.
- Lector 2D QR - DP8405.
- Relé 5v 1 Canal.
- Cerradura Eléctrica 12V.
- Shield Ethernet W5100
- Pantalla Lcd 1602-I2c
- Cargador 12V.

## **Software**

- IDE de Arduino
- Fritzing
- Fusion 360

### **Lenguaje de programación.**

- C++
- Php

Este proyecto contribuye a la línea de investigación, Internet de las cosas a través de las redes de comunicación, sensores eléctricos y sistemas informáticos. Ya que, abarca la creación de un sistema eléctrico con lectores QR, permitiendo el control de acceso a las aulas, y consumiendo información desde la base de datos, alojada en un servidor web. [8]

## **1.3 OBJETIVOS DEL PROYECTO**

### **1.3.1 OBJETIVO GENERAL**

Implementar un sistema de cerradura eléctrica utilizando lectores de código QR y dispositivos adecuados para permitir y controlar el acceso a aulas del Bloque 1 de la Facultad de Sistemas y Telecomunicaciones en la UPSE.

### **1.3.2 OBJETIVOS ESPECÍFICOS**

- Determinar los dispositivos electrónicos y mecánicos necesarios para llevar a cabo el diseño del sistema de cerradura eléctrica.
- Programar la placa controladora Arduino Uno, para que realice las respectivas validaciones de las entradas y salidas del sistema.
- Esquematizar las conexiones internas y externas del dispositivo, para la correcta instalación del sistema de cerradura eléctrica.

## 1.4 JUSTIFICACIÓN

El continuo progreso de la tecnología en cuanto a sistemas de seguridad y de vigilancia ha llevado a que la gran mayoría de hogares, negocios e instituciones públicas y privadas tengan la necesidad de poseer equipos que le faciliten el resguardo de sus establecimientos y por lo consiguiente de sus activos físicos. [9]

Gracias a los avances tecnológicos y a la domótica, hoy existen sistemas de control y vigilancia para nuestros hogares, los cuales utilizan una unidad central, que es la que recibe las instrucciones, procesa y envía las órdenes para los distintos dispositivos a controlar que han sido conectados a dicha unidad. Aunque funcionales, la mayoría de estos sistemas tienen un costo elevado y en nuestro país no es muy fácil conseguirlos. [10]

Debido a la necesidad de integrar nuevas tecnologías para la seguridad de las aulas del bloque 1 de la Facultad de Sistemas y Telecomunicaciones, a través de este proyecto, el cual comprende desde el diseño a la implementación, se busca mejorar aspectos que van netamente relacionados a la seguridad de dichas aulas, considerando los principales requerimientos que una cerradura común y corriente debería cumplir, adicionando el factor eléctrico y su respectiva manipulación sin llaves, a fin de garantizar la no falsificación de llaves físicas, se incursiona al uso de llaves virtuales mediante códigos QR.

La implementación del sistema de seguridad eléctrico, beneficiará a quienes conforman la comunidad universitaria en general, de tal manera que los usuarios puedan sentirse seguros, además de obtener un cómodo y ágil acceso a las aulas.

El principal beneficiario es el técnico docente, quien es el encargado de controlar el acceso a las aulas, puesto que, la cerradura eléctrica automatiza el proceso de aperturar las aulas de forma manual, con llaves físicas.

Por otro lado, y no menos importante, es que también es de gran aporte a los usuarios tales como docentes, presidentes de curso y estudiantes en general, brindando un mayor grado de tranquilidad, comodidad y agilidad en el proceso de obtener el acceso al aula de

clase, debido a que las llaves virtuales o los ya mencionados códigos QR se pueden llevar consigo mismo en cualquier dispositivo móvil o celular.

Dicho sistema de seguridad electrónica, además de estar dirigido al bloque 1 de la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, sin duda alguna puede escalar notablemente en los demás bloques de la Facultad, y como no en los diferentes sectores que comprenden la universidad en su totalidad. Además, la propuesta puede ser aplicada a cualquier otra institución educativa, dado que aporta significativamente a la seguridad de una habitación específica, y al control de acceso a la misma.

El presente proyecto, está alineado a los objetivos del **Plan Nacional de Desarrollo**, que se detallan a continuación:

## **Eje 2: Economía al Servicio de la Sociedad**

**Objetivo 5:** Impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria. [11]

**Política 5.6:** Promover la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica, la invención y el emprendimiento, la protección de la propiedad intelectual, para impulsar el cambio de la matriz productiva mediante la vinculación entre el sector público, productivo y las universidades [11].



## 1.5 ALCANCE DEL PROYECTO

El sistema de seguridad electrónica, está destinado a la automatización de los procesos que conllevan a la accesibilidad de las aulas de clases, en el Bloque 1 de la Facultad de Sistemas y Telecomunicaciones en la Universidad Estatal Península de Santa Elena (UPSE).

El sistema se llevará a cabo mediante los siguientes módulos:

- **Módulo de Entradas:** Comprende la conexión del lector 2D QR, parte primordial del sistema quien se encargará de procesar los códigos QR escaneados, el Shield de Ethernet mediante el cual se tendrá acceso a la red y por ende conexión con el servidor. Se considera además la respectiva programación, para que dichos componentes actúen correctamente.
- **Módulo de Salidas:** La conexión de la cerradura eléctrica que actuará permitiendo el paso hacia el aula, un relé 5V para enviar la señal a la cerradura, y una pantalla LCD a través de la cual el sistema emitirá mensajes de alerta entorno al funcionamiento y manipulación del mismo, siendo el caso más relevante, cuándo se escanee el código QR y éste sea validado o no.
- **Módulo de Procesamiento:** Pudiéndose llamar también como el módulo de programación, comprende la codificación de todos los componentes electrónicos, así como también el tratamiento de las señales de entrada para el accionamiento respectivo de las señales de salida.
- **Módulo de implementación:** Encierra el diseño mecánico del sistema poniendo en consideración aspectos importantes como: la infraestructura física del dispositivo, el cableado eléctrico y el cableado de red.

El sistema de seguridad electrónica no abarcará una aplicación móvil o web, para su gestión, ni tampoco, presentará reportes, es netamente la fabricación de hardware y programación como tal para el correcto desempeño, éste se complementará con las dos primeras partes del sistema en general, para la recepción y envío de información siendo el caso del código QR.

## **2. MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO**

### **2.1 MARCO TEÓRICO**

#### **2.1.1. AHORRO ECONÓMICO PARA LA TECNOLOGÍA CIENTÍFICA DE CÓDIGO ABIERTO Y LIBRE: UNA REVISIÓN.**

El uso de plataformas de hardware libre abre paso al desarrollo de un sin número de proyectos académicos o personales, los cuales comprenden un diseño de sistema electrónico que gira en torno a un microprocesador, el cual sus creadores difunden a consciencia con total libertad, cabe mencionar que estas plataformas han obtenido una notable acogida en los últimos años por parte de la educación en tecnología, dando la potestad de utilizarse sin tener que cancelar algún tipo de licencia, entre ellos destaca Arduino. [12]

Joshua M. Pearce, expone que “la tecnología de código abierto es una herramienta muy poderosa dentro de la ciencia, porque un investigador puede crear y publicar un diseño”. [12] La fabricación digital distribuida de hardware libre y de código abierto o también conocida como: FOSH, por sus siglas en inglés, ha sido base para que científicos puedan desarrollar herramientas científicas personalizadas, además de proveer el código del hardware, también pone a disposición la lista de materiales, los esquemas, las instrucciones de instalación de componentes, su respectiva programación, y otra información que se requiera para crear un artefacto físico. [12]

Las ventajas que poseen las tecnologías libres y de código abierto son muchas, sin embargo, se destacan cuatro de ellas que se fijan sobre el diseño tradicional de productos: 1) los diseños y su garantía de idoneidad para cumplir una determinada tarea de aspecto científico, esta ventaja se cumple debido a que los diseños no sólo son libres, si no que han sido desarrollados conscientemente por los usuarios; 2) la construcción del equipo experimental, esta ventaja permite entender todos los conocimientos planteados en el diseño además de que se esclarecen los límites, ambas ventajas promueven una mejor ciencia; 3) la fabricación es local e inmediata; 4) el movimiento de código abierto globalizado es un fenómeno que permite el reclutamiento de programadores y constructores con talento fuera del entorno tradicional científico . [12]

### **2.1.2. CONTROL DE ACCESO FÍSICO BASADO EN EL CÓDIGO QR.**

En la actualidad, se puede apreciar la existencia de distintos métodos para la autenticación del control de acceso físico, entre ellos: llaves USB y las tarjetas OTP (One Time Password-código de un solo uso), sin embargo, al momento de implementarse dichos métodos se requiere de hardware adicional, en caso de que estos componentes físicos se extravíen, no será fácil percatarse de forma inmediata debido a la baja tasa de uso. [13] El teléfono móvil por otra parte, es uno de los dispositivos que más uso se le da en el diario vivir, va con nosotros a casi todos los sitios, siendo así, que la pérdida de dicho dispositivo produciría un estímulo de respuesta inmediata por nuestra parte. [13]

Y. Kao, G. Luo, H. Lin, Y. Huang y S. Yuan, plantean un método de autenticación para el control de acceso físico utilizando el equipo común, el teléfono móvil y la técnica del código QR, por otro lado, se hace uso de la contraseña de un solo uso (OTP), así como también el método de cifrado asimétrico RSA, el cuál comprende la creación de una llave pública y una privada, mediante las cuales se envuelve el paquete enviado por la red y se asegura de que éste no sufra ataques o alteraciones. [13]

La arquitectura del sistema comprende 3 partes en las que se describe: un servidor principal como ORS (OTP RSA SERVER) el cuál se utiliza para registrar la información de los usuarios con acceso permitido además del servicio relacionado, una aplicación móvil denominada OQG(OTP QRcode Generator) es el que manejará la recepción de mensaje multimedia o mms, así como también la generación de la imagen QR de autenticación; un pc cliente llamado ORC (OTP RSA Client), poseerá una cámara para la lectura respectiva del código. [13]

Es posible la autenticación a través del código QR, partiendo del envío de la imagen de autenticación hacia el dispositivo móvil mediante mensajería multimedia, lo que libera al usuario llevar consigo otro tipo de hardware específico que representen un costo adicional. Además, gracias al uso de OTP la seguridad en la autenticación se refuerza debido a que la contraseña sólo será válida una vez, a esto se le suma el uso de RSA, que produce que la imagen QR como tal, sea difícil de acceder, modificar o copiar, por lo que se concluye de que dicho método se puede aplicar a muchos servicios que requieran autenticación. [13]

## 2.2 MARCO CONCEPTUAL

### 2.2.1. ARDUINO UNO

Arduino Uno es un hardware de uso libre, basado en el chip microcontrolador ATmega328P que fue creado por Atmel. Posee entre sus características generales 14 pines de entrada/salida digital, de los cuáles 6 de ellos pueden ser utilizados como salidas PWM (Pulse Width Modulation), este tipo de salidas son útiles para controlar la cantidad de energía que se maneja sobre una carga; otro de los componentes que lo caracteriza es un resonador cerámico de 16MHz (CSTCE16M0V53-R0), el cuál detalla la velocidad de reloj del sistema; cuenta también con conexión USB, un puerto para conexión de alimentación, un encabezado ICSP() y un botón de reinicio. [14]

<i>Microcontrolador</i>	<a href="#">ATmega328P</a>
<i>Voltaje de funcionamiento</i>	5V
<i>V. Entrada recomendada</i>	7-12 V
<i>V. Entrada límite</i>	6-20 V
<i>Pines de E / S digitales</i>	14 (de los cuales 6 proporcionan salida PWM)
<i>Pines PWM</i>	6
<i>Pines Entrada analógica</i>	6
<i>Memoria flash</i>	32 KB en total, de los cuales 0,5 KB son consumidos por el gestor de arranque
<i>SRAM</i>	2 KB
<i>EEPROM</i>	1 KB
<i>Velocidad de reloj</i>	16 MHz
<i>LED_BUILTIN</i>	13
<i>Largo</i>	68,6 milímetros
<i>Ancho</i>	53,4 milímetros
<i>Peso</i>	25 g

*Tabla 1. Especificaciones técnicas de la placa Arduino Uno. [14]*

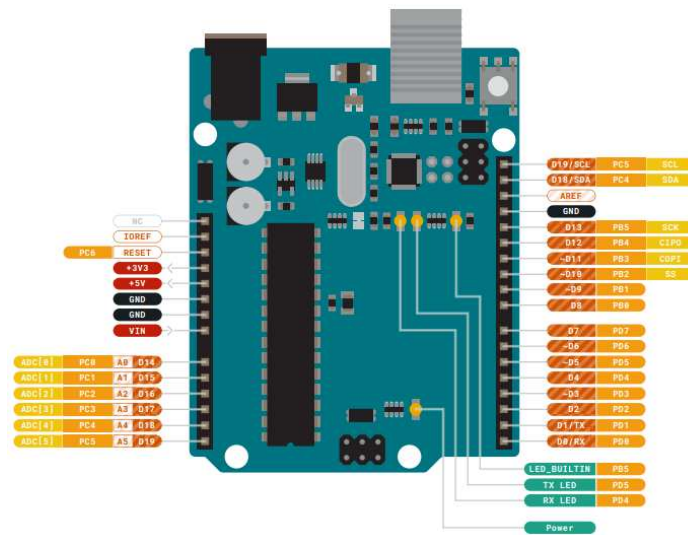


Figura 1. Diagrama de pines del Arduino Uno [15]

### 2.2.2. IDE ARDUINO

El IDE de Arduino es software open source que facilita la escritura de códigos o programas para luego poder ser transmitidos o cargados en la placa, sirve además con todo tipo de modelo de placas producidos por arduino. Su descarga se realiza de forma segura desde la página web oficial “arduino.cc”, existen algunas opciones de descarga, cada una de ellas disponible para los diferentes sistemas operativos existentes: Windows, Linux, Mac OS x 10.10 o más reciente. [16]

Debido a que es un software de código abierto, en la página oficial se expone el enlace al repositorio GitHub dónde se encuentra el código fuente del IDE de Arduino. [16]

### 2.2.3. LECTOR 2D QR - DP8405

Es un lector que logra interpretar códigos en 2D, es decir en 2 direcciones, vertical y horizontal, de manera que pueden contener muchos más datos a diferencia de los códigos de barras, o códigos 1D. [17]

Éste poderoso lector 2D QR, es producido por la empresa DYscan de china, especializada en la fabricación de escáneres de códigos de barras, quién actualmente cuenta con personal cualificado con más de 15 años de experiencia y todas sus patentes cuentan con certificaciones tales como: CE, ROHS, FCC, ISO9001, ISO14001. [17]

<i>Fuente de luz</i>	56000K LED (iluminación)
<i>CPU</i>	32 bits
<i>Campo visual</i>	80° (H) x 60° (V)
<i>Resolución</i>	640 x 480
<i>Velocidad el descifrar</i>	50 CM/S
<i>Profundidad del campo</i>	0mm-210mm
<i>Modo de exploración</i>	Sentido automático
<i>Luz ambiente</i>	Ambiente oscuro, luz natural interior
<i>Símbolos</i>	1D: Código 39, código completo 39 del ASCII, código 128, código 93, código 11, interpolación 2 de 5, NEC 2of 5, derecho 2 de 5 industriales, derecho 2 de 5 IATA, GS1-128.  2D: PDF417, QR CODE, QR MICRO, DATAMATRIX, CÓDIGO AZTECA, HanXin, Maxicode, PDF417 micro, etc.
<i>Modo de comunicación</i>	USB, TTL
<i>Fuente de alimentación</i>	DC 5V
<i>Temperatura de funcionamiento</i>	-20°C a 60°C

Tabla 2. Especificaciones técnicas del lector 2D QR CODE. [17]



Figura 3. Vista frontal del lector QR DP8405 [31]



Figura 2. Vista posterior del lector QR DP8405 [31]

#### 2.2.4. RELÉ 5V 1 CANAL

Interruptor accionado por un electroimán, el cual permite abrir o cerrar un circuito en el que se disipa una potencia mayor a la que se utilizó para estimular dicho circuito., principalmente se compone por 1 relé o un canal normalmente fabricado para la usabilidad con Arduino, su conexión es asequible, así como también su programación, además depende solamente de un voltaje de entrada (TTL-5V) producido por uno de los pines de salida de la placa arduino. [18]

<i>Voltaje de funcionamiento</i>	5V DC
<i>Señal de Control</i>	3.3V - 5V Interfaz TTL
<i>N.º de Relays</i>	<b>1 CH</b>
<i>Capacidad máx.</i>	10A/250VAC, 10A/30VDC
<i>Corriente máx.</i>	10A (NO), 5A (NC)
<i>Tiempo de acción</i>	10 ms / 5 ms
<i>Para activar salida NO</i>	0 voltios

*Tabla 3. Especificaciones técnicas Relé 5V 1 canal [19]*

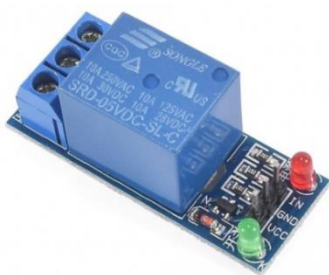


Figura 4. Módulo relay 5DC 1CH [19]

#### 2.2.5. SHIELD ETHERNET W5100

El ethernet shield es una herramienta indispensable a la hora de querer conectar arduino a internet mediante conexión LAN, este módulo es compatible con algunas placas de arduino tales como: Uno, Leonardo, y Mega. La versión del chip integrado en el módulo ethernet es la W5100, el cuál ha sido diseñado para facilitar implementación de conectividad a Internet sin la necesidad de un SO. [20]

La interfaz de conexión que maneja el W5100 es la SPI (Serial Protocol Interface), este protocolo viene integrado en las placas de arduino en sus respectivos pines dedicados, para el caso de arduino uno, los pines predeterminados para el uso de SPI son: 13 (CKL), 12(MISO),11(MOSI) y 10(SS). [20]

<i>Chip</i>	Wiznet W5100
<i>Conector</i>	Rj45
<i>Tamaño del búfer interno</i>	16k
<i>Velocidad de conexión</i>	10/100Mb
<i>Interfaz</i>	SPI
<i>Conector</i>	RJ45

*Tabla 4. Especificaciones técnicas del módulo de ethernet W5100 [20]*



*Figura 5. Módulo de ethernet W5100*

## 2.2.6. VISUALIZADOR LCD

Las pantallas de cristal líquido LCD o display LCD para mensajes (Liquid Cristal Display) tienen la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier equipo electrónico de una forma fácil y económica.

La pantalla consta de una matriz de caracteres (normalmente de 5x7 o 5x8 puntos) distribuidos en una, dos, tres o cuatro líneas de 16 hasta 40 caracteres cada línea. El proceso de visualización es gobernado por un microcontrolador incorporado a la pantalla.

<i>Alimentación</i>	5V
<i>Pines</i>	16
<i>Número de caracteres</i>	32 a la vez
<i>Caracteres</i>	ASCII, Kanji, griego, símbolos matemáticos

*Tabla 5. Especificaciones técnicas LCD 2x16 [21]*



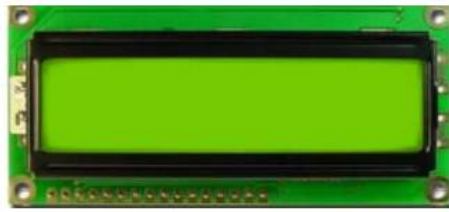


Figura 6. LCD 2x16 [21]

### 2.2.7. C++

C++, es un lenguaje de programación diseñado en la época de los setenta por Dennis Ritchie, de los laboratorios Bell, se creó con la intención de que se convirtiera en el lenguaje del sistema operativo UNIX, sin embargo, fue para los años ochenta en que nace su autor B. Stroustrup y se empieza a desarrollar la prolongación del lenguaje C, éste nuevo lenguaje sería el encargado de introducir el concepto de objetos que posteriormente empezó a ser utilizado por el año 1983. En la actualidad es un lenguaje versátil, potente y general; mantiene las ventajas del C, en cuanto al conjunto de operadores lógicos y expresiones, además de la flexibilidad y eficiencia. [22]

### 2.2.8. PHP

Un lenguaje de código abierto muy popular para el desarrollo de aplicaciones y páginas web, su fácil uso y constante perfeccionamiento es una alternativa segura para aquellos que quieren incursionar a la creación de proyectos calificados pero que no presenten complicaciones. Diseñado para la programación de scripts del lado del servidor, como, por ejemplo: la recopilación de datos en formularios, la generación de páginas dinámicas, así como también el envío y recepción de cookies. [23]

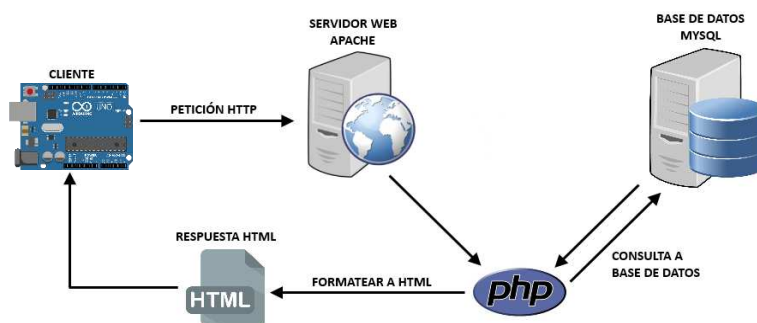


Figura 7. Esquema de funcionamiento Php.

### 2.2.9. FUSION 360

Es una herramienta indispensable a la hora de realizar un modelado en 3D, su arquitectura de funcionamiento actualmente es en la nube, permite de forma gratuita la realización de 10 diseños, luego de esto el pago de su licencia es requerida. Para la carcasa o el contenedor de los componentes electrónicos y la placa controladora principal se hizo uso de esta herramienta, logrando un diseño acorde a los gustos y necesidades. [24]

### 2.2.10. PROTOCOLO HTTP

El envío y la recepción de datos entre el cliente y el servidor es posible mediante HTTP, también conocido como protocolo de transferencia de hipertexto, es un protocolo muy utilizado por los elementos de software para comunicarse entre sí, debido a que hace posible desde visualizar páginas web hasta el intercambio de datos entre aplicaciones. [25]

El comando HTTP Request hace posible enviar todo tipo de petición HTTP a un URL específico y por lo consiguiente el procesamiento de la respuesta del servidor HTTP. [25]

La forma común en que los usuarios acceden a una página web desde el navegador es a través de una dirección que sigue un formato denominado URL (Universal Resource Locator). [25]

<i>Esquema</i>	Aquí se define el mecanismo de acceso al recurso. Por lo general corresponde con el protocolo de comunicaciones, por ejemplo: http, ftp o https.
<i>Usuario</i>	Esto es opcional, se define el usuario que accede al recurso sea este propietario o no.
<i>Contraseña</i>	Esto es opcional, se define la contraseña del usuario.
<i>Host</i>	Se detalla el nombre completo o la dirección IP del servidor que alberga el recurso al que se quiere acceder.
<i>Path (sendero):</i>	Esto es opcional, se define la dirección del recurso dentro del servidor.
<i>QueryString</i>	Opcional. Lista de parámetros usados para acceder al recurso, estos se componen de pares “nombre=valor” separados por “&”
<i>Fragment</i>	Opcional. Se referencia a un fragmento dentro de un recurso. Por ejemplo: la sección de una página web.
<i>Ejemplo</i>	Esquema://usuario:contraseña@host:puerto/path?querystring#fragment

Tabla 6. Formato URL [25]

### 2.2.11. SPI

Conocido como Interfaz Periférica de Serie, es un protocolo de datos en serie síncrono muy utilizado para comunicarse con uno o más dispositivos de una manera eficaz en distancias cortas. [26] SPI es un bus de tres líneas, que maneja transmisiones de paquetes de 8 bits, una de las características del SPI es que cada microcontrolador o dispositivo puede actuar como receptor y transmisor al mismo tiempo, llamándose comunicación serial full duplex. [26]

Los dispositivos conectados al bus reciben los nombres de maestros y esclavos, teniendo en cuenta que un maestro es aquel que inicia la transferencia de información sobre el bus y es el encargado de generar las señales de reloj y control, por otro lado, un esclavo es un dispositivo controlado por el maestro, a través de una línea selectora denominada Chip Select o Select Slave, por lo que, un esclavo es llamado solo cuando esta línea es seleccionada. [26]

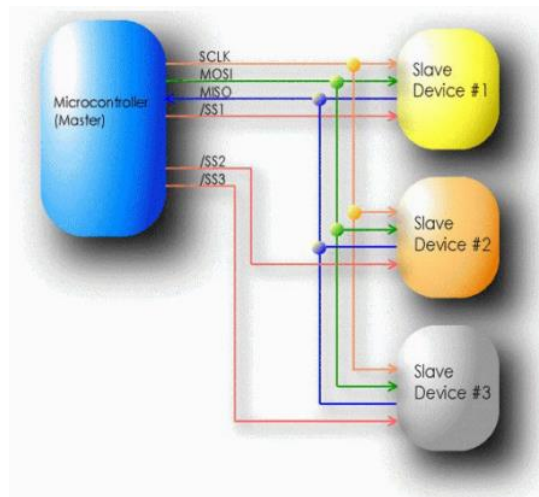


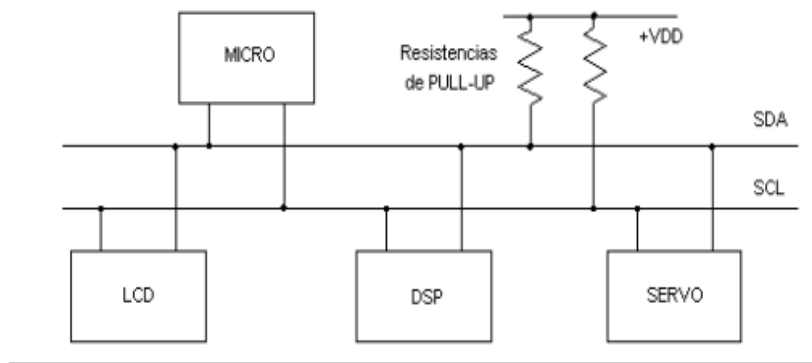
Figura 8. SPI Bus: Un maestro 2 esclavos [26]

### 2.2.12. ESTÁNDAR I2c

A inicios de los ochenta, una de las empresas reconocidas a nivel de fabricantes de electrónica (Phillips), puso en marcha una norma de comunicación digital que serviría para establecer contacto entre los diferentes componentes de un sistema electrónico a lo que se llamó Inter Integrated Circuits Bus o I2c, convirtiéndose de manera repentina en un estándar importante en la industria [27]. La idea básica de su funcionamiento es el siguiente:

- Protocolo de 2 hilos: SDA (Transmisión de datos) y SCL (Reloj asíncrono que indica cuándo leer datos), sumándose GND(Tierra) y 5V (En ciertos casos).
- Cada dispositivo conectado al bus I2c posee una dirección exclusiva de 7bits, en teoría se puede tener hasta 128 dispositivos.

Es importante tener en cuenta que los pines que funcionan como SDA Y SCL en Arduino Uno son A4 y A5 respectivamente. [27]



*Figura 9. Ejemplo de una configuración del bus I2c [27]*

### **2.2.13. MICROCONTROLADOR**

Un microcontrolador es una computadora pequeña que contiene en su interior de manera básica, un procesador, soporte (reloj y reset), memoria y puertos de entra y salida, cada uno de estos componentes se encuentran dentro de un pequeño chip al que se puede acceder para su programación con total flexibilidad y relativa facilidad. [28]

Variables como: el tamaño, consumo de corriente, economía, se hacen presente a la hora de hablar de un microcontrolador, hoy en día existen varios fabricantes en el mundo entre los que destacan Intel, phillips, microchip, Atmel, de los cuales los 2 últimos se encuentran incluidos en una gran variedad de placas para uso práctico del estudiante. [28]

## **Microcontroladores Atmel AVR**

Atmel es una familia de microcontroladores AVR (Arquitectura Harvard) con buses y procesamiento separados tanto para los datos como para el programa, usa además la arquitectura RISC como núcleo, esto quiere decir que poseen 8 bits con bajo consumo, que comprenden 32 registros, o también 256 posiciones accesibles, dichos microcontroladores llegan a procesar 1MIPS (millones de instrucciones por segundo) por 1 MHz de frecuencia por reloj. Según el modelo, incorporan diferentes tamaños de memoria flash, RAM y EEPROM. [28]

- **Memoria Flash:** Es la encargada de permitir programar el microcontrolador.
- **Memoria EEPROM:** Es la que permite almacenar datos aún luego de que se apague el dispositivo

## **2.3 METODOLOGÍA DEL PROYECTO**

### **2.3.1 METODOLOGÍA DE INVESTIGACIÓN**

Debido a la poca información adquirida referente al desarrollo de una cerradura eléctrica con acceso mediante código QR, permita mejorar la seguridad de una habitación, aula de clase y similares, se opta por la utilización de la metodología de investigación de tipo exploratorio [29]. Luego de haber realizado una búsqueda acerca de proyectos semejantes, se constataron cuáles son sus similitudes y diferencias en comparación del presente trabajo.

Existen proyectos que han sido desarrollados a nivel mundial, y otros que, a nivel interno de países como España, laboran como empresas ofreciendo sus productos como servicios, siendo así los mencionados en los antecedentes, Lockart [5] y Altiria TIC [6], con base a la orientación que poseen en cuanto a sus características generales como sistema de seguridad mediante código QR, así como también la forma en que otorgan el acceso a un hogar o establecimiento.

De la misma manera, se toma como ejemplo el proyecto ejecutado en los laboratorios de la UPSE, el cual consiste en la implementación de un sistema de acceso de biométricos, ZKAccess [7], permitiendo un control y administración minucioso. Sin embargo, este tipo de sistema es de costos elevados y no se adaptan a la idea de la utilización de llaves virtuales como lo son los códigos QR.

### **2.3.2 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN**

A fin de identificar los procedimientos que se llevan a cabo en cuanto al control y seguridad de las aulas de clases del Bloque 1 de la Facultad de Sistemas y Telecomunicaciones en la UPSE, se utilizaron técnicas de recolección de información las cuales son: método de observación (**Ver anexo 1**) y la entrevista (**Ver anexo 2**), dichas técnicas, permitieron identificar problemas existentes en la seguridad de los cursos, así como también el control inexistente en cuanto a los responsables que emplean los mismos.

Para lograr el objetivo propuesto, se hizo uso de la metodología de investigación de tipo diagnóstico [29], logrando determinar a los principales beneficiarios del proyecto, pudiendo rescatar de forma directa al Técnico Docente de la Facultad, quien es el encargado de las llaves de todos los cursos, por otro lado, los beneficiarios indirectos, serán aquellos quienes conforman la comunidad estudiantil que hagan uso de las aulas del Bloque 1.

Mediante el presente proyecto, se obtendrán mejores resultados, en cuanto al tiempo de espera para la apertura de un aula de clase. Para lo cual, los usuarios a evaluar serán los docentes y estudiantes (presidentes de curso) que hagan uso del sistema.

#### **2.3.2.1. ANÁLISIS DE RESULTADOS DE MÉTODO DE OBSERVACIÓN**

El método de observación (Ver anexo 2) aplicado en el Bloque 1 de la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, tuvo como resultado lo siguiente:

- Las cerraduras físicas que poseen las aulas de clases se encuentran discontinuadas.
- Docentes y estudiantes acceden a cualquier aula de clase que se encuentra desocupada sin previo aviso.
- Se debe buscar al técnico docente como encargado de las llaves, cuando se requiere ingresar a un aula de clase
- La mayor parte del tiempo los cursos permanecen abiertos, esto se debe a que docentes y estudiantes olvidan cerrar las puertas, por lo tanto, no existe responsable alguno en caso de pérdida de bienes materiales.

Con base a lo anterior mencionado, se llega a la conclusión de que dichos inconvenientes se producen a diario, debido a que no existe algún dispositivo tecnológico que apoye al control de acceso a las aulas de clases.

### 2.3.2.2. ANÁLISIS DE RESULTADOS DE ENTREVISTA

Los resultados obtenidos luego de la entrevista al técnico docente de la Facultad de Sistemas y Telecomunicaciones, respecto a las labores que él desempeña, fueron los siguientes:

- ✓ Presenta inconvenientes a la hora de trasladarse a aulas de clases alejadas de su oficina.
- ✓ El técnico docente afirma que le hace falta seguridad a las aulas de clases
- ✓ Han existido ocasiones en que se ha querido usar un aula y ésta se encuentra ocupada.

Una de las respuestas durante la entrevista por parte del técnico docente, fue de que sin duda alguna la automatización de procesos es fundamental para poder llevar con más rapidez dichos procesos. Con esto se concluye que existe la necesidad de automatizar ciertos procesos para el control de acceso y que además brinden mayor seguridad en las aulas de clases del Bloque 1.

### 2.3.3 METODOLOGÍA DE DESARROLLO DEL PROYECTO

Dado que el proyecto utiliza el microcontrolador de la placa Arduino Uno, las funcionalidades que uno programa sobre dicho componente se denomina sistema embebido, que es aquel diseñado para realizar funciones específicas dedicadas, siendo el caso de este proyecto: el control de acceso a aulas de clases mediante el uso de lector QR. A diferencia de los proyectos de desarrollo de software, los sistemas embebidos no siguen el mismo bosquejo, una de las razones es que su programación o también denominada firmware va dirigida al control de circuitos electrónicos, es decir que se programa la funcionalidad del hardware en forma general.

De esta forma se establece que la metodología a utilizar será la de **desarrollo iterativo e incremental** [30], la cual posee un desarrollo modular, con entregas parciales del producto denominados "incrementos" del sistema, gracias a esto, en cada incremento se podrá comprobar el estado del proyecto permitiendo una rápida adaptación a los cambios.



Cabe mencionar que dicha metodología será adaptada a las necesidades del sistema, considerando que no es simplemente el desarrollo de un software, si no el diseño, la creación, y la implementación de un sistema embebido.

El modelo cuenta con cuatro fases: análisis, diseño, codificación, pruebas e implementación. Sin embargo, teniendo presente de que el sistema comprende tanto hardware como software, se considera por separado la aplicabilidad de las fases, de tal manera que para la parte de hardware no se tendrá en cuenta la codificación, debido a que esta fase va dirigida netamente al software.

Fases del modelo iterativo e incremental:

✓ **Hardware:**

- **Análisis:** Consiste en especificar las necesidades del Hardware, así como también la selección de los componentes electrónicos.
- **Diseño:** Se procederá a elaborar el diseño de las conexiones de cada componente eléctrico.
  - Para esto se utilizará la herramienta Fritzing, además se considerarán las dimensiones de los componentes eléctricos para el diseño estructural del dispositivo final.
- **Pruebas e implementación:** Se implementará por separado las conexiones de los siguientes módulos: Sistemas de entradas (Shield Ethernet, lector QR), Sistema de salidas (Pantalla LCD, Cerradura eléctrica), Procesamiento (Microcontrolador). Con base al método de prueba y error se realizarán las pruebas correspondientes hasta llegar al producto final

✓ **Software**

- **Análisis:** Estudio y definición de los requisitos del sistema relacionado a toda la lógica de control del hardware.
- **Diseño:** Estructuración de los algoritmos que controlaran los componentes eléctricos.

- **Codificación:** Teniendo en cuenta el diseño del programa y todo lo que concierne a la lógica de programación, se interpretarán los algoritmos al lenguaje C de Arduino.
- **Pruebas e implementación:** en esta fase existe la cohesión entre los algoritmos de control y el hardware físico, mediante la cual se realizará la comprobación de la correcta funcionalidad del sistema.

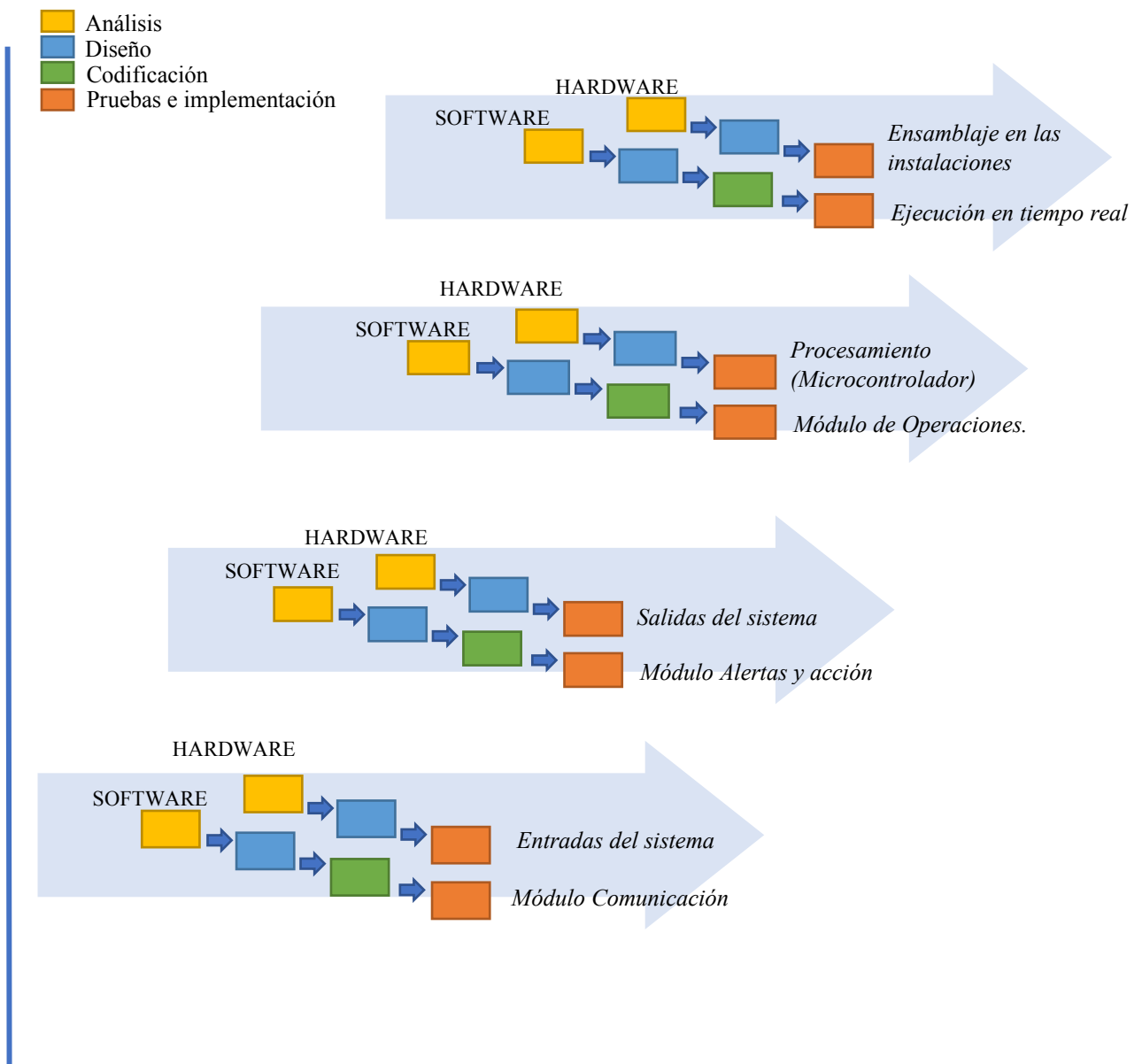


Figura 10. Modelo incremental del Sistema - Hardware y Software.

### 3. PROPUESTA

#### 3.1. REQUERIMIENTOS

##### 3.1.1. REQUERIMIENTOS FUNCIONALES

<i>Cód.</i>	<b>REQUERIMIENTOS FUNCIONALES</b>
<i>RF-1</i>	El sistema será instalado físicamente a un costado de las puertas correspondientes a las aulas de clase del bloque 1 de la Facultad de Sistemas y Telecomunicaciones.
<i>RF-2</i>	El sistema deberá contar con un suministro de energía en el aula dónde se va a instalar.
<i>RF-3</i>	El sitio dónde se instalará el sistema deberá poseer acceso a la red.
<i>RF-4</i>	El sistema contará con un webservice almacenado en un servidor web, el cuál contendrá los archivos necesarios para poder realizar las consultas a la base de datos.
<i>RF-5</i>	El sistema validará la conexión con el servidor antes de que se ejecute cualquier tipo de consulta.
<i>RF-6</i>	El sistema permitirá el acceso a los docentes y estudiantes que requieran el uso del aula y que dispongan del código QR para su acceso respectivamente.
<i>RF-7</i>	El sistema validará el acceso del usuario dependiendo del código QR escaneado.
<i>RF-8</i>	El sistema validará el código QR, considerando que el código escaneado se encuentre relacionado con el aula al que se quiera acceder.
<i>RF-9</i>	Una vez validado la relación del código con el aula, se procede a comprobar el estado del código QR, si éste es “A”, la comprobación es exitosa, caso contrario no ejecutará ninguna acción.
<i>RF-10</i>	El sistema validará el código QR respecto al horario de clases y las peticiones de las aulas en horarios desocupados.
<i>RF-11</i>	La base de datos contará con la tabla: códigos, en dónde se registrarán todos los códigos QR (llaves digitales) disponibles.
<i>RF-12</i>	La base de datos contendrá la tabla: lecturas, en donde se almacenarán los registros de los códigos QR validados.
<i>RF-13</i>	El sistema enviará los parámetros: código y aula, mediante petición GET hacia el webservice alojado en el servidor.
<i>RF-14</i>	El sistema recibirá respuesta, una vez ejecutada la consulta mediante petición GET, y dicha respuesta será almacenada en una variable de tipo string para su respectivo procesamiento.
<i>RF-15</i>	La forma en que el sistema manipulará los aparatos eléctricos será a través de señales digitales.
<i>RF-16</i>	El sistema enviará una señal digital al módulo relé, el mismo que accionará la cerradura eléctrica una vez validado el código QR escaneado, permitiendo el acceso al curso
<i>RF-17</i>	La cerradura eléctrica deberá poseer en su mecanismo de acción, un botón accionador el cuál permitirá la apertura desde el interior del aula.
<i>RF-18</i>	La conexión del dispositivo a la red será por cable mediante ethernet.
<i>RF-19</i>	El sistema deberá adquirir una dirección IP mediante DHCP

RF-20	El sistema debe tener asignado una dirección MAC que no se encuentre disponible en la red.
RF-21	El sistema emitirá un mensaje por pantalla mientras se encuentre en estado de reposo.
RF-22	El sistema emitirá un mensaje por pantalla cuándo un código QR es aceptado.
RF-23	El sistema emitirá un mensaje por pantalla cuándo un código QR es denegado
RF-24	El sistema será programado a través del IDE de Arduino.
RF-25	Los archivos alojados en el webservice serán programados en lenguaje PHP.

*Tabla 7. Requerimientos funcionales*

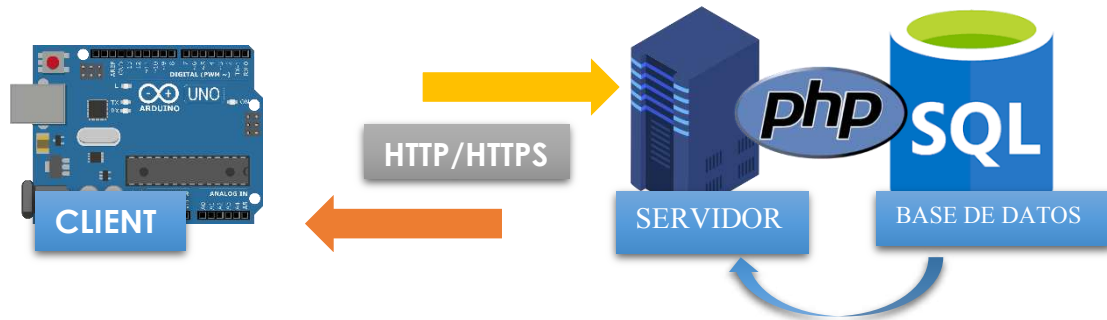
### 3.1.2. REQUERIMIENTOS NO FUNCIONALES

<b>Cód.</b>	<b>REQUERIMIENTOS NO FUNCIONALES</b>
RNF-1	El sistema será controlado por la placa Arduino Uno.
RNF-2	El sistema tendrá conectado una pantalla 16x02 I2c.
RNF-3	El sistema tendrá conexión Ethernet a través del shield ethernet W5100
RNF-4	El sistema tendrá conectado una cerradura eléctrica que trabaja a 12v.
RNF-5	El sistema tendrá conectado un módulo Relé para accionar la cerradura eléctrica.
RNF-6	El sistema leerá el código QR a través del lector QR modelo: DP8405
RNF-7	El sistema será suministrado por un voltaje de 5V
RNF-8	El sistema se conectará como cliente hacia un servidor web en el que estará alojada una de base de datos Mysql.

*Tabla 8. Requerimientos no funcionales*

## 3.2. COMPONENTES DE LA PROPUESTA

### 3.2.1. ARQUITECTURA DEL SISTEMA



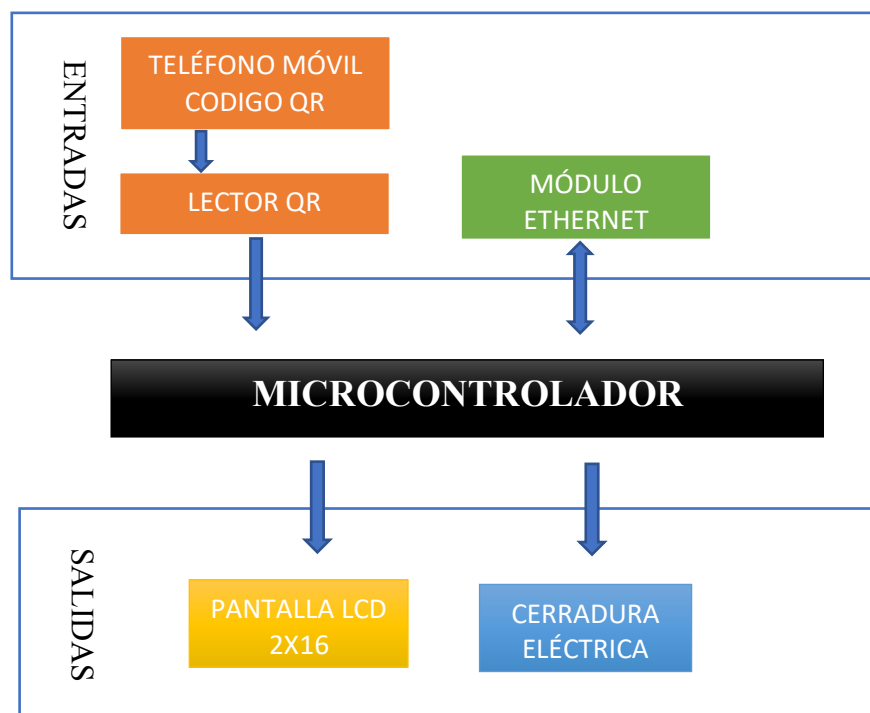
*Figura 11. Arquitectura del sistema*

La arquitectura general del sistema comprende la del tipo cliente – servidor, gracias a la capacidad que tiene arduino para programarse como cliente, es idóneo para realizar peticiones frecuentes hacia un servidor. El proceso inicia desde el cliente, en el momento en que arduino detecta un código QR establece conexión con el servidor, si la comunicación es exitosa, arduino envía la petición mediante URL a través del protocolo Http indicando la ruta específica en el servidor dónde se encuentra alojado un Webservice que contiene los archivos Php respectivos los cuales poseen los parámetros adecuados para establecer conexión con la base de datos y por lo posterior realizar las consultas programadas, si el proceso es exitoso el archivo Php emite una respuesta que es recibida por el cliente (arduino), el mismo que procesa la respuesta y acciona o no la cerradura eléctrica.

### 3.2.2. DIAGRAMA DE BLOQUES DE COMPONENTES

En el siguiente diagrama se pueden apreciar los bloques que comprenden el sistema para el cumplimiento de los requisitos propuestos anteriormente.

- Sistema de entrada
- Sistema de procesamiento (microcontrolador)
- Sistema de salida



*Figura 12. Diagrama de bloques de los elementos.*

### 3.2.2.1. SISTEMA DE ENTRADA

Este sistema es el encargado de adquirir los valores de las variables externas y la información enviada por el servidor para su posterior gestión a cargo del sistema de procesamiento.

#### - Selección del lector QR.

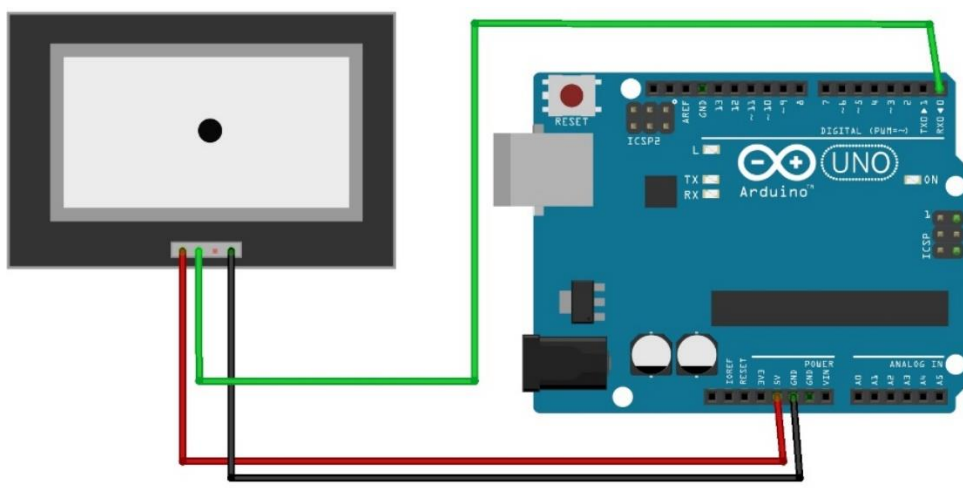
La utilización del código QR tiene un gran impacto a nivel mundial en diversas áreas de implementación que van desde la fabricación, el almacenamiento, inventariado de productos hasta la utilización como método de pago móvil, envío express, cadena de venta al por menor, entre otros. Ante tales demandas nacen los lectores QR, de igual forma de distintas dimensiones, velocidades y alcance.

<b>CARACTERÍSTICA</b>	<b>DP8402</b>	<b>DP8405</b>	<b>DS6100</b>
<i>Interfaz</i>	USB, TTL	USB, TTL	USB, RS232
<i>Fuente de alimentación</i>	5V 120mA	5V 120mA	<b>5V 130mA</b>
<i>Resolución</i>	640 x 480	640 x 480	640 x 480
<i>Profundidad del campo</i>	10cm a 20cm	0 a 21 cm	1cm a 50cm
<i>Velocidad de descifrar</i>	25 cm/s	50 cm/s	25cm/s
<i>Dimensiones</i>	L62 cm, W58 cm, H23cm	L62 cm, W58 cm, H23cm	L15cm, W7cm, H9,5cm
<i>Precio</i>	\$15	\$19	\$30
<i>Imagen.</i>			

*Tabla 9. Selección de lector QR.*

Relativamente los tres modelos de lectores QR poseen características similares en cuanto a la resolución, la profundidad del campo, y la interfaz, sin embargo, de las principales razones que se tomaron en cuenta para la selección del lector QR, fueron: La velocidad de descifrar, las dimensiones y el precio, por lo cual se determinó como modelo elegido el lector DP8405.

- **Conexión de lector QR-DP8405 con Arduino**






*Figura 13. Diagrama de conexiones de DP8405 con Arduino.*

El lector QR-DP8405 es alimentado con 5V directamente desde la placa Arduino, a su vez se conecta el pin TX (cable color verde) al pin RX de la placa Arduino declarado. Cabe recalcar que como todo dispositivo que se comunica por conexión serial, trae consigo el pin RX, sin embargo, este pin no es utilizado en el proyecto por motivo de que sólo se necesita que el lector QR transmita hacia la placa Arduino Uno, más no que reciba datos.

- **Selección del módulo Ethernet**

El módulo de ethernet shield brinda la capacidad de conectar una placa Arduino a una red por medio de cableado y un conector estándar Rj45, además de que éste se utilice como cliente o servidor.

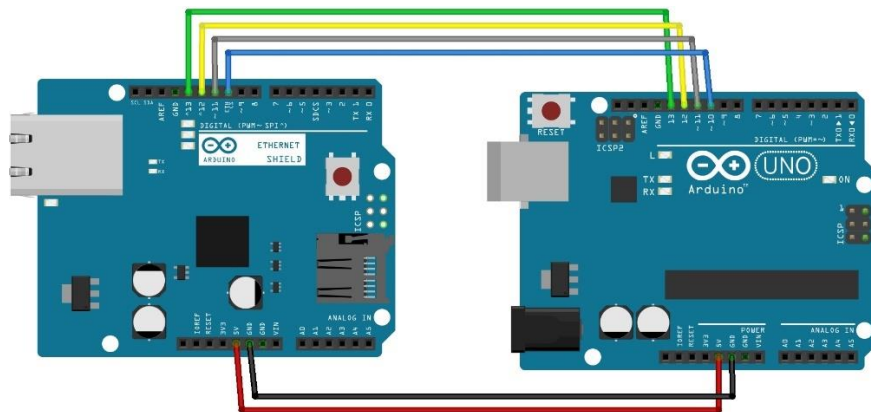
<b>CARACTERÍSTICA</b>	<b>W5100</b>	<b>W5500</b>	<b>ENC28J60</b>
<i>Interfaz</i>	ISP	ISP	ISP
<i>Fuente de alimentación</i>	5V DC.	3,3v a 5v	3,3v a 5v
<i>Compatibilidad</i>	Arduino Uno, Mega y Leonardo	Arduino Uno, Mega y Leonardo	Arduino Uno, Mega y Leonardo
<i>Velocidad</i>	10/100 Mbps	10/100 Mbps	10/100 Mbps
<i>Precio</i>	\$10	\$11	\$5
<i>Imagen</i>			

*Tabla 10. Comparación de módulos Ethernet*

El componente elegido para la conexión de arduino a internet, fue el módulo de Ethernet Shield W5100, quién pese a no variar mucho en cuánto a las funcionalidades respecto a los demás, es necesario destacar que el consumo de la memoria dinámica en el Arduino uno es muy bajo en comparación a los demás módulos.



- **Conexión del módulo Ethernet Shield W5100 con Arduino**



*Figura 14. Diagrama de conexiones de Ethernet Shield W5100 con Arduino.*

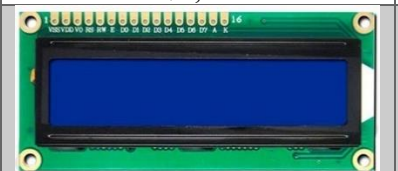
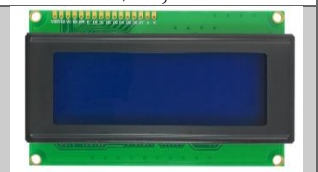
La conexión del Ethernet Shield W5100 no requiere de gran ciencia, se basa en el hecho de simplemente de conectar cada pin del ethernet shield a los pines del arduino uno, sin embargo, hay que tener presente que los pines del 10 al 13 vienen por defecto configurados para SPI (protocolo que permite la comunicación entre dos o más componentes) esto quiere decir que dichos pines no podrán ser utilizados.

**3.2.2.2. SISTEMA DE SALIDA**

En este sistema comprende los dispositivos actuadores que reaccionan de acuerdo a las señales u ordenes obtenidas en la parte del sistema de entrada, en esta etapa se encuentra principalmente la cerradura eléctrica para realizar la operación final del proceso.

- **Selección de visualizador**

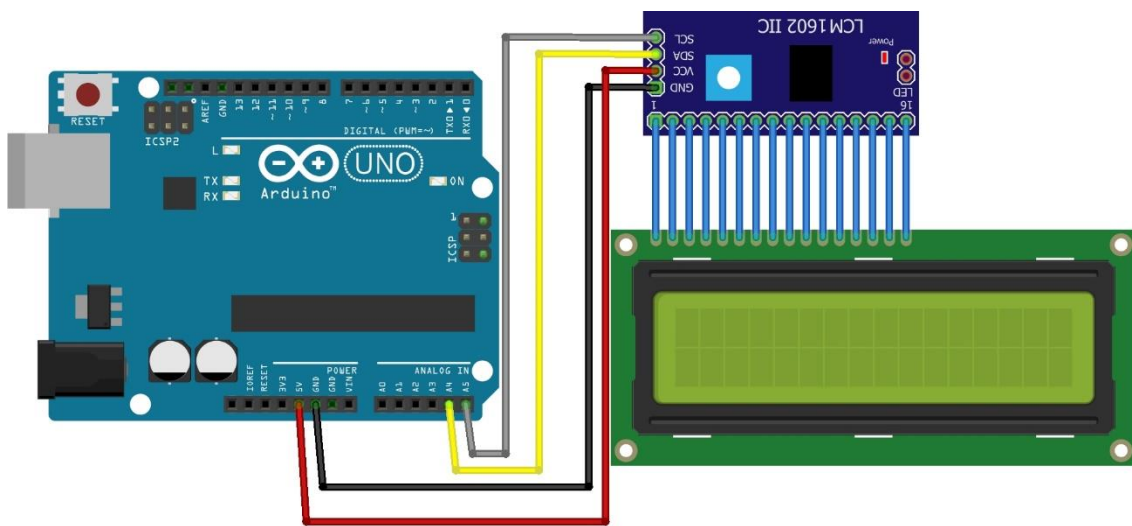
La utilización de un módulo LCD es necesario para mejorar la interacción con el usuario final, en él se aprecian los mensajes de estado o alertas que emite el dispositivo, en resumen, permite que el usuario sepa lo que está ocurriendo en el instante.

<b>CARACTERÍSTICA</b>	<b>16x2</b>	<b>20x4</b>
<b>Alimentación</b>	5v	5v
<b>Pines</b>	16	16
<b>Número de caracteres</b>	32 a la vez	80 a la vez
<b>Costo</b>	\$4,50	\$13,50
<b>Imagen</b>		

*Tabla 11. Comparativa de módulos LCD*

Las características de ambos módulos en principio son muy similares, lo que los diferencia es el hecho de que la capacidad de caracteres mostrados por pantalla es mayor en el caso del LCD 20x4 con respecto a la de 16x2, así como también se aprecia la diferencia de costos. En base a esta comparativa se determinó el uso del módulo 16x2 debido al costo y al hecho de que los mensajes a mostrar por el sistema no ocupan más de 32 caracteres.

- **Conexión del display LCD por comunicación I2C con Arduino**



*Figura 15. Diagrama de conexión I2C LCD*

Como se puede apreciar en el diagrama el número de pines que utiliza la pantalla LCD es de 16, por lo cual se hace uso de un módulo I2C permitiendo disminuir 14 pines de los puertos digitales, siendo así que sólo se necesitó 2 pines o líneas conectados a los puertos A5 y A4 destinados para la señal de reloj SCL y la señal de datos SDA respectivamente, sin dejar de tener en cuenta que el módulo ocupa 2 pines para la alimentación mediante 5V y su respectiva línea de tierra.

- **Selección de la cerradura eléctrica.**

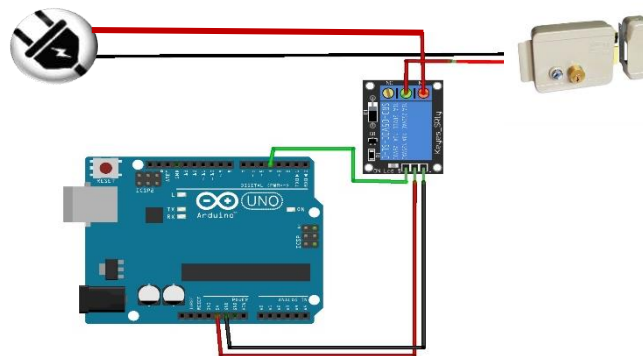
Una cerradura eléctrica es un componente electro-mecánico con fines de brindar seguridad a un determinado espacio físico que posea una puerta de acceso. En el mercado ecuatoriano existen un sin número de estos accesorios de los cuales se consideraron 3 de ellos y que se detallan sus características a continuación:

<b>CARACTERÍSTICA</b>	<b>CHAPA</b>	<b>MAGNÉTICA</b>	<b>SOLENOIDE</b>
<b>Alimentación</b>	12v	12v	12v
<b>Componente</b>	Fierro	Aluminio	Fierro
<b>Peso</b>	2165 gr.	130 gr.	145 gr.
<b>Costo</b>	\$ 33,50	\$25	\$10
<b>Imagen</b>			

*Tabla 12. Comparativa de cerraduras eléctricas.*

Cualquiera de los 3 tipos de cerraduras eléctricas se maneja a 12v, sin embargo, una de las principales características es el modo de funcionamiento, tanto la cerradura magnética como el solenoide se accionan siempre y cuándo exista conexión eléctrica, éste sería un problema en caso de que se requiera acceder al aula y no haya energía eléctrica, por tal motivo pese a su elevado costo se optó por el modelo de chapa, que a más de accionarse mediante una señal eléctrica se puede accionar por el mecanismo tradicional de llaves físicas.

- **Conexión cerradura eléctrica**



*Figura 16. Diagrama de conexión cerradura eléctrica mediante Relé.*

La conexión de la cerradura eléctrica se realiza mediante un módulo relay de 1 canal, útil para el manejo de cargas de componentes de alto voltaje como: bombillas de luz, motores AC, electroválvulas y para el caso del proyecto, solenoides. El módulo relay acciona la salida NO (Normalmente abierta) cuándo recibe una señal digital “0” o 0 voltios y se desactiva con “1” o 5v.

El relé es alimentado desde la placa arduino, además se conecta al pin 4 del Arduino para recibir la señal que este le envía. En cuanto a la salida del relé se conecta directamente el cable de corriente positiva de la cerradura eléctrica al puerto COM, y el puerto NO va

conectado a la fuente de alimentación 12V en este caso. La cerradura eléctrica conectará la línea negativa a la fuente de alimentación, de esta forma el circuito para manejar la cerradura eléctrica queda concluido.

### 3.2.2.3. SISTEMA DE PROCESAMIENTO (Microcontrolador)

Este sistema cumple dos funciones primordiales, la primera es el procesamiento y el análisis de las señales emitidas por el sistema de entrada a través de la programación del microcontrolador, mientras que su segunda función es efectuar el envío de señales en forma de órdenes a fin de poner en funcionamiento el sistema de salida.

#### - Selección del microcontrolador





<b>Características de Arduino</b>	<b>UNO</b>	<b>Mega 2560</b>	<b>Leonardo</b>	<b>NANO</b>
<i>Microcontrolador</i>	Atmega 328	Atmega2560	Atmega 32U4	Atmega 168
<i>Velocidad de reloj</i>	16 MHz	16 MHz	16 MHz	16 MHz
<i>Pines E/S digitales</i>	14	54	20	14
<i>Pines E. analógicas</i>	6	16	12	8
<i>Memoria FLASH de programa</i>	32 kb	256 kb	32 kb	32 kb
<i>Memoria de datos (SRAM)</i>	2 kb	8 kb	2.5 Kb	2 kb
<i>Memoria auxiliar (EEPROM)</i>	1 kb	4 kb	1 kb	1 kb
				

Tabla 13. Comparativa de microcontroladores Arduino [28].

La familia de microcontroladores elaborados por el fabricante estadounidense Atmel, ofrecen varios diseños con dimensiones y características que se distinguen entre sí, en la tabla comparativa mostrada anteriormente se rescataron 4 modelos de los cuales el microcontrolador escogido fue el de la placa Uno, Atmega 328 ofrece las capacidades adecuadas para el efecto del sistema, considerando el número de pines, y el espaciado de memorias.

- Conexión del microcontrolador con sus respectivos dispositivos

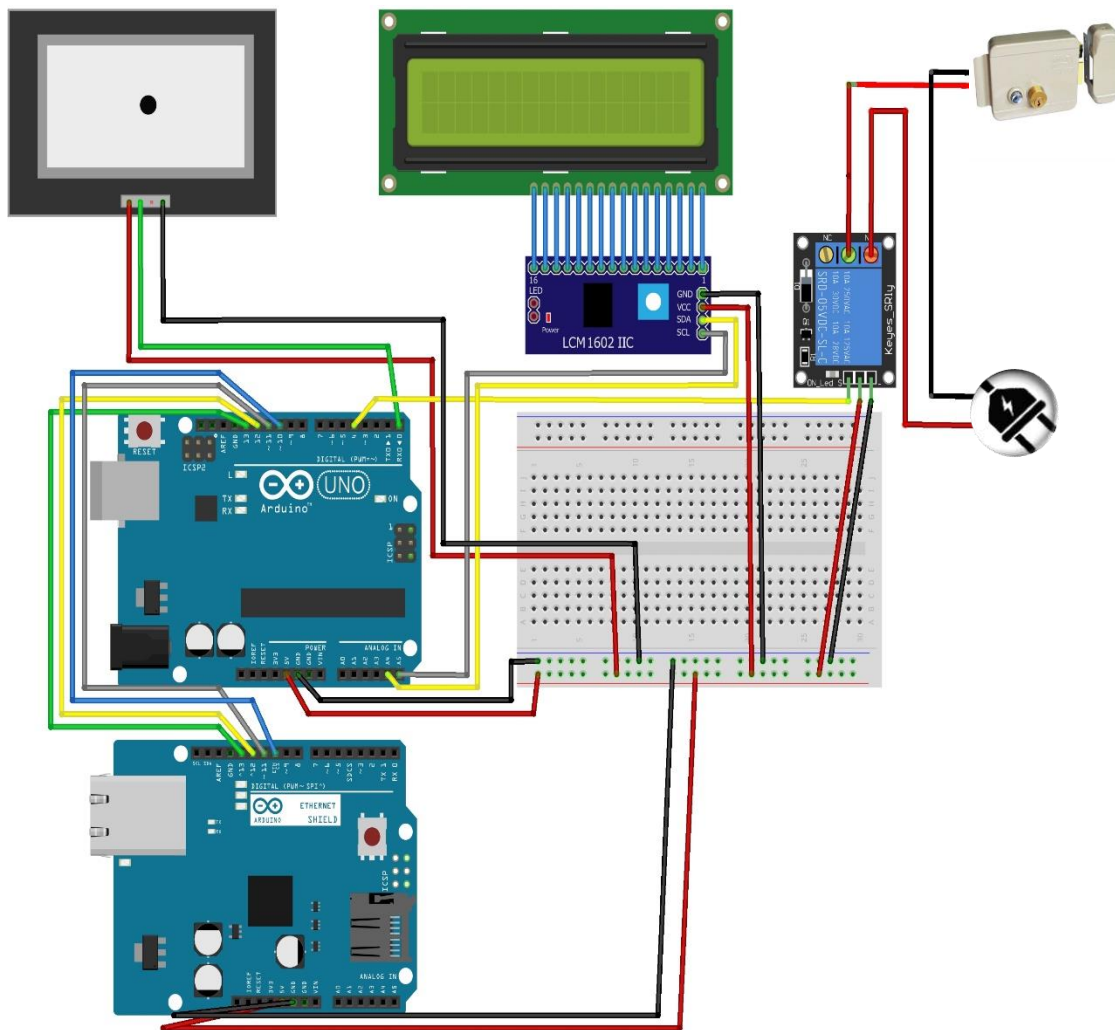


Figura 17. Diagrama final de conexiones del sistema

### 3.2.3. DIAGRAMA ESQUEMÁTICO DE COMPONENTES

A continuación, se muestra un diagrama con todos los elementos o dispositivos electrónicos que se utilizaron en la construcción del sistema de cerradura eléctrica con apertura mediante código QR.

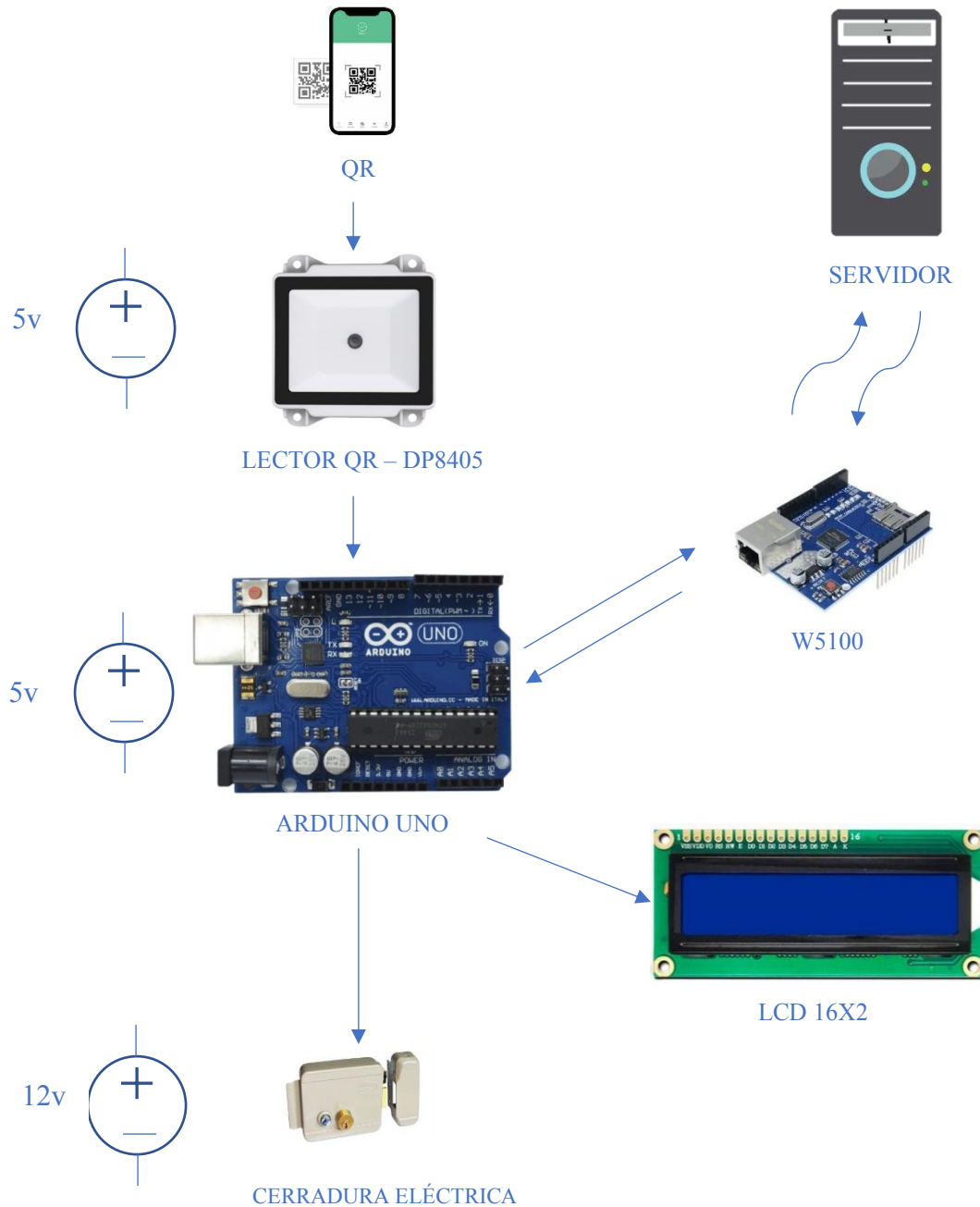
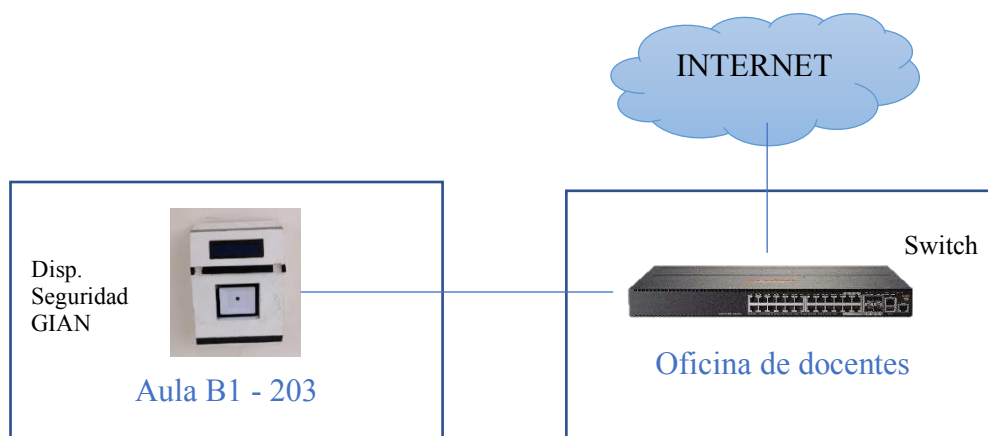
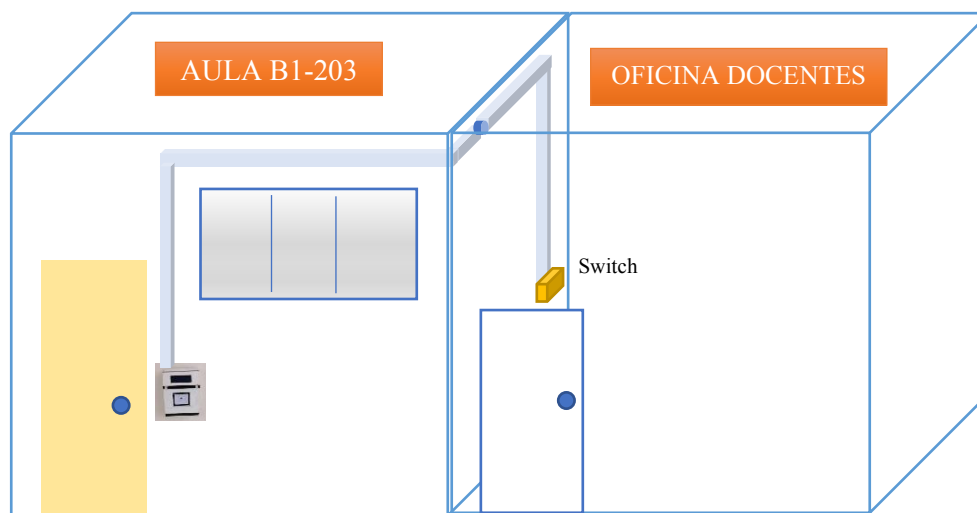


Figura 18. Diagrama esquemático de componentes.

### 3.2.4. DIAGRAMA ESQUEMÁTICO DE RED

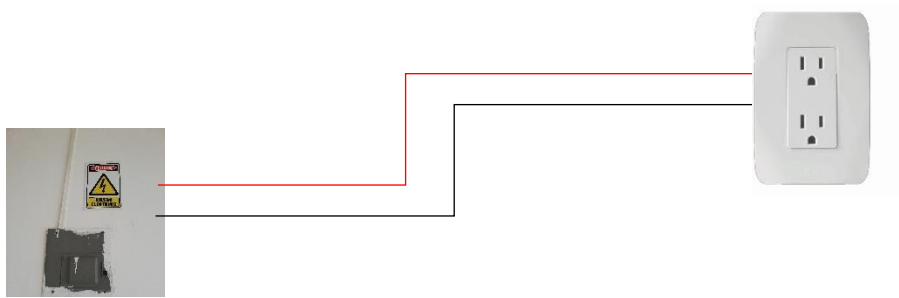


*Figura 19. Diagrama esquemático básico de red.*

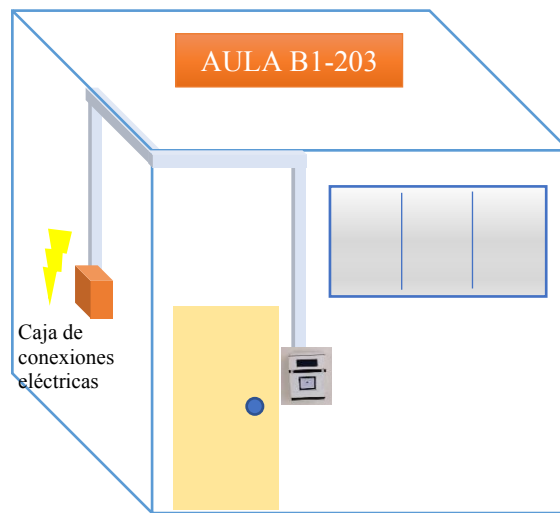


*Figura 20. Diagrama conexión switch – dispositivo de seguridad.*

### 3.2.5. DIAGRAMA ESQUEMÁTICO DE CONEXION ELÉCTRICA



*Figura 21. Diagrama básico - conexión eléctrica.*



*Figura 22. Diagrama esquemático - instalación de cableado eléctrico.*



### 3.2.6. DIAGRAMA DE ACTIVIDAD

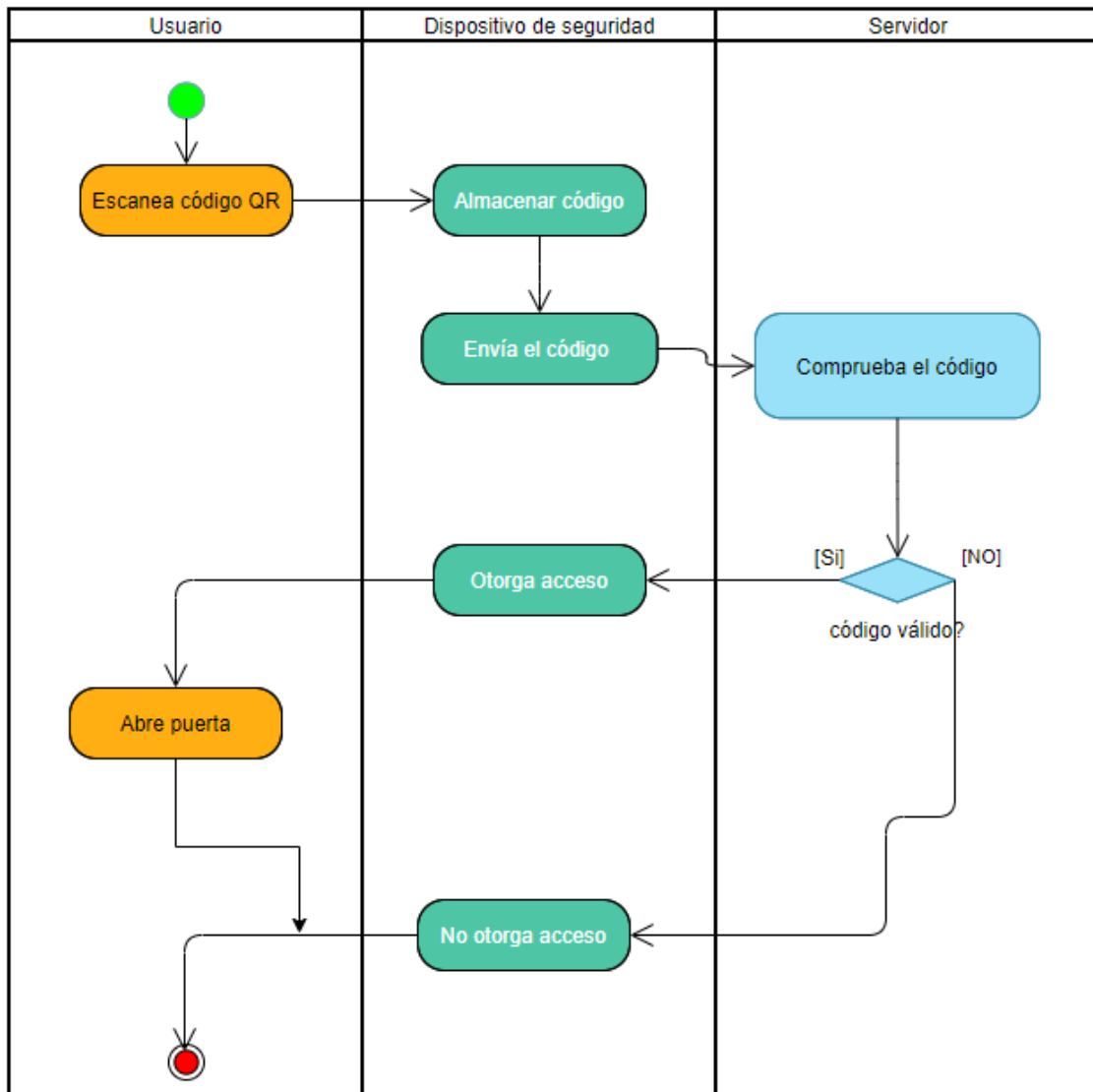


Figura 23. Diagrama de actividad del sistema de seguridad.

### 3.2.7. DIAGRAMAS DE PROCESOS

#### 3.2.7.1. PROCESO DE VALIDACIÓN QR

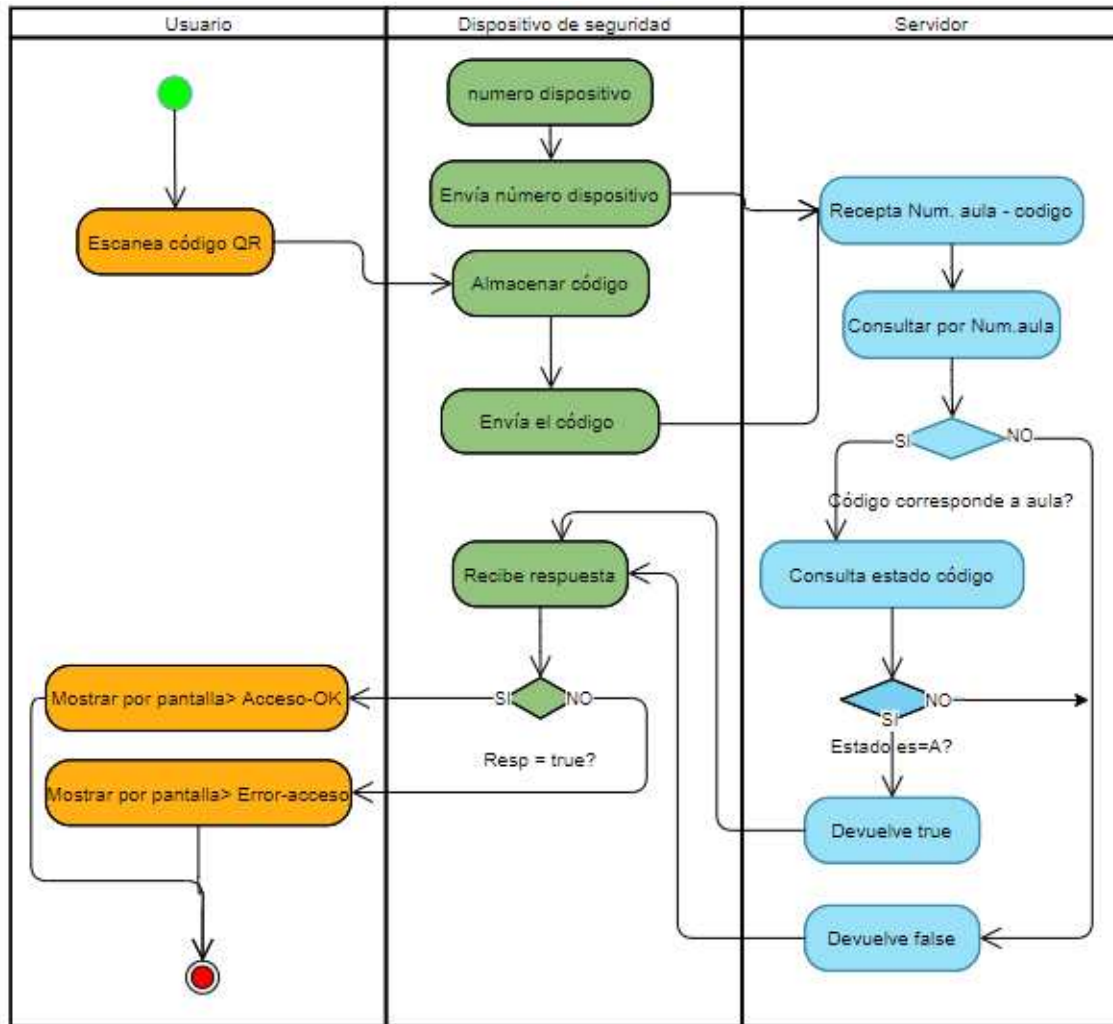


Figura 24. Diagrama de proceso validación de código QR

### 3.2.7.1. PROCESO DE CONEXIÓN A LA RED POR DHCP

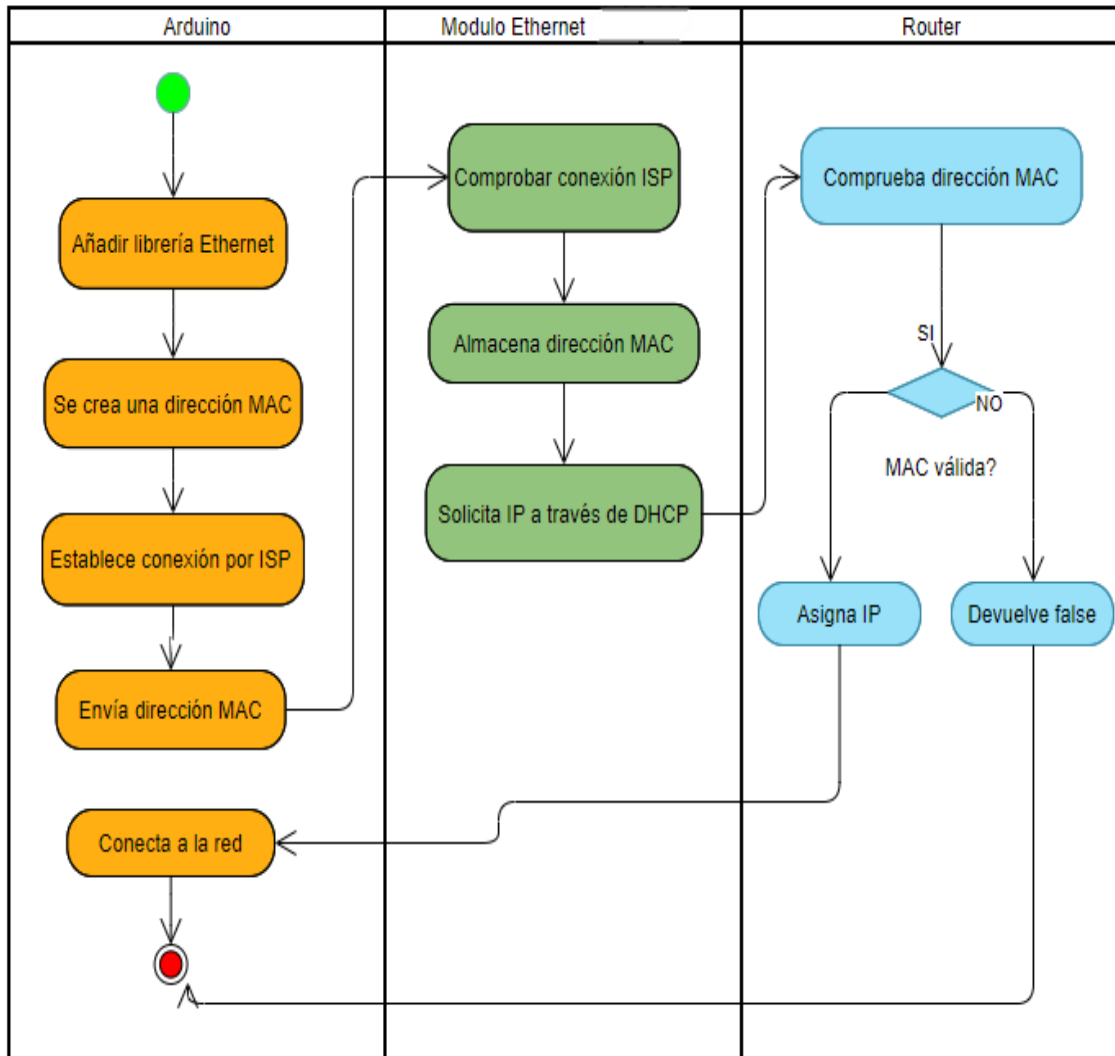


Figura 25. Diagrama de proceso para adquirir dirección IP por DHCP

### 3.2.8. DIAGRAMA DE BLOQUES GENERAL – PROGRAMACIÓN DEL SISTEMA

El sistema de cerradura eléctrica con acceso mediante código QR consta de varios módulos tales como: Entradas, Salidas, Procesamiento e Implementación, dichos módulos encierran de forma general a los componentes electrónicos necesarios para llevar a cabo el funcionamiento del sistema.

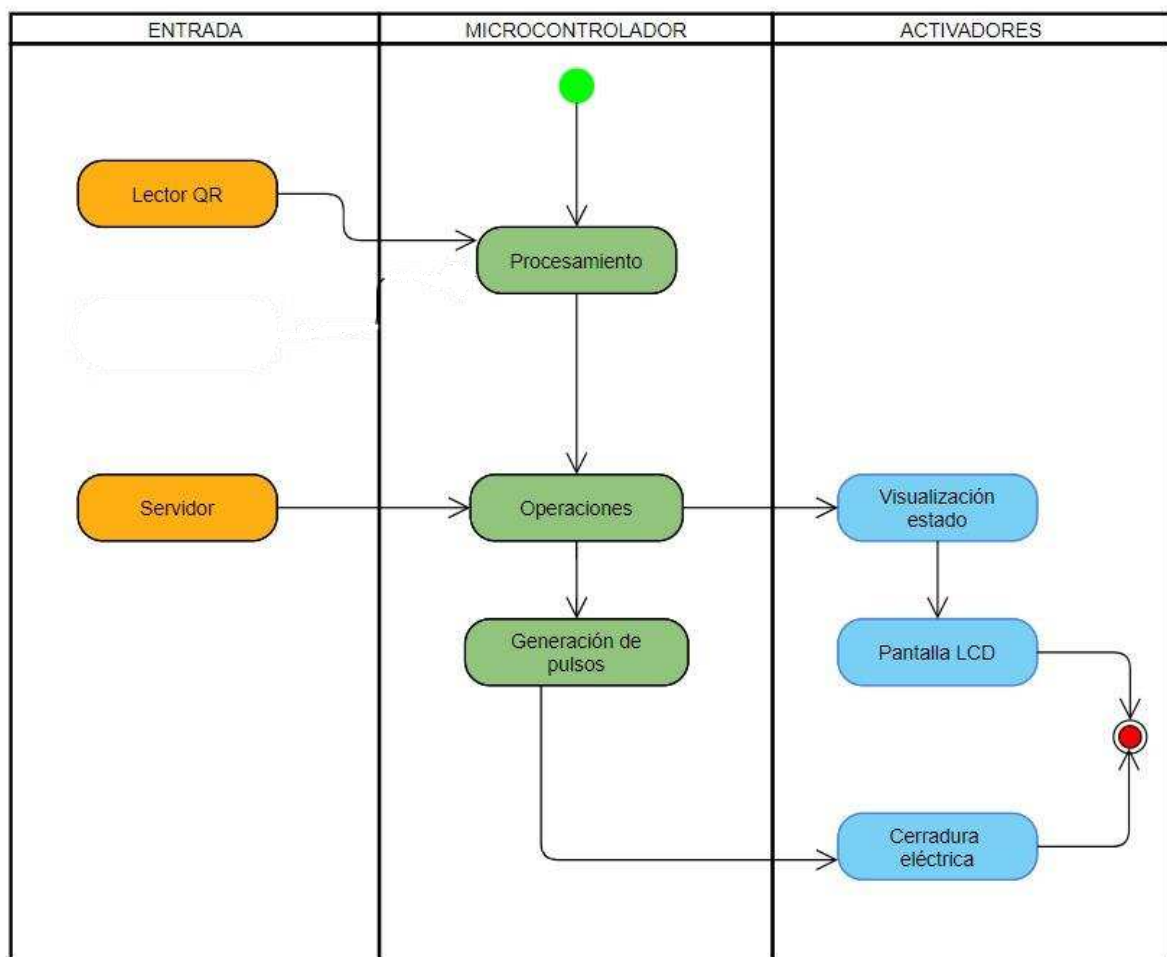


Figura 26. Diagrama de bloques general – Programación del sistema

### 3.2.9. PROGRAMACION DEL MICROCONTROLADOR

La programación del microcontrolador es la parte más importante del sistema, puesto que es el momento en el que se determinan varias condiciones como la forma de trabajo, la gestión e interpretación de los datos.

Por tal motivo se detallarán las etapas de programación acorde a los módulos de la metodología del proyecto.

#### 3.2.9.1. MÓDULO DE COMUNICACIÓN

El módulo de comunicación comprende la inicialización de los componentes que conforman las entradas del sistema: Lector QR y Ethernet Shield.

##### - Velocidad de transmisión de bits (baudios)

La velocidad de transmisión de bits se mide en baudios, en dónde cada baudio equivale a 1 bit/s. Existen diferentes velocidades normalizadas que se encuentran disponible respecto al arduino, como, por ejemplo: 115200, 56700, 38500, 19200, 9600, 2400, 1200, 600 y 300 baudios.

La comunicación entre el lector QR y Arduino se realiza mediante el puerto serial, así como también con la PC, por tal motivo se estableció una sola velocidad de transmisión:

```
Serial.begin (9600); //Velocidad del puerto serie
```

##### - Librerías utilizadas para el módulo de comunicación

Cada componente maneja su propia librería, por tal motivo en principio se descargaron las librerías correspondientes para luego poder utilizarlas en el código del programa.

```
#include <Ethernet.h>
```

*Figura 27. Librerías Módulo de comunicación*

**Ethernet:** Esta biblioteca viene incluida en el IDE de Arduino, y es fácilmente accesible, brinda la posibilidad de configurar el dispositivo como cliente o servidor, mediante los métodos `client` o `server`, en el caso de este proyecto se configura al dispositivo de arduino como cliente, y además se declara una variable que contendrá la dirección del servidor para las consultas posteriores.

#### - Inicialización del módulo Ethernet

Una vez que se haya incluido la librería Ethernet, el siguiente paso es configurar el módulo para que en este caso arduino actúe como cliente o servidor dentro de la red. Siguiendo la arquitectura (cliente-servidor) anteriormente planteada, el dispositivo de seguridad será configurado como tipo cliente, para lo cual se debe establecer una dirección MAC dentro de una variable, lo que permitirá que el dispositivo pueda ser reconocido en la red y a su vez conectarse mediante dirección IP estática o dinámica en el caso de DHCP.

```
static byte mymac[] = {0x74, 0x69, 0x69, 0x2D, 0x31, 0x31};  
IPAddress server(52,233,72,68);  
EthernetClient client;
```

Figura 28. Inicialización del módulo Ethernet

#### - Adquisición de dirección IP mediante protocolo DHCP

Se hace uso del método *Ethernet.begin("mymac")*, al cual se le envía por parámetro la dirección MAC previamente almacenada en un arreglo denominado "mymac". En caso de que se quiera imprimir por pantalla o por el puerto Serial la dirección IP asignada por el router se utilizó el método *Ethernet.localIP()*;

```
Serial.println(F("Adquiriendo dirección IP"));  
Ethernet.begin(mymac);  
Serial.println(Ethernet.localIP());
```

Figura 29. Adquisición de dirección IP por DHCP

#### - Inicialización del lector QR

La placa Arduino uno trae por defecto un puerto serial que es utilizado para realizar la conexión directa con un ordenador, dicho puerto se hace presente en los pines 0 y 1 RX y TX respectivamente. Es importante mencionar que al conectar un dispositivo a estos pines no se podrá hacer uso del puerto serie que viene en la entrada tipo B para establecer conexión con la computadora. Por tal motivo, siempre que se quiera subir un código al arduino, los pines 0 y 1 deberán estar libres de conexiones.

- **Declaración de variables para el uso del lector QR**

```
String codigoleido="";  
bool stringcomplete = false;
```

*Figura 30. Declaración de variables para el uso del lector QR*

Se utilizará una variable de tipo String la cual receptorá los bytes almacenados en el buffer al momento de escanear un código QR, además, se declara una variable de tipo bool que devolverá true cuándo el código QR se haya terminar de leer, actuando como una interrupción del programa.

### 3.2.9.2. MÓDULO DE ALERTAS

- **Librerías**

```
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>
```

*Figura 31. Librerías utilizadas para el módulo de alertas*

**Wire:** Esta biblioteca hace posible la comunicación con dispositivos I2C y por lo general es implementada para la comunicación con pantallas LCD, en el caso de este proyecto se usa la pantalla 16x2 mediante I2c para la reducción del uso de pines digitales. Por defecto viene estipulado en la librería que para las placas de arduino la línea de datos o SDA debe conectarse al pin A4, y la línea de reloj o SCL al pin A5.

**LiquidCrystal\_I2C:** Esta librería sirve para especificar la dirección única de la pantalla que se va a conectar, en ciertos casos viene estipulada en el propio componente, de no ser así, existen programas para Arduino que sirven para proporcionar dicha información de la pantalla LCD. Esta dirección es importante, ya que sin ella no se podrá tener acceso a la pantalla LCD.

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

*Figura 32. Declaración de variable LiquidCrystal\_I2c*

- **Inicialización de pantalla LCD.**

Una vez declarada la variable de tipo LiquidCrystal\_I2C, se procede a inicializar la pantalla LCD en el setup (), mediante los métodos init () y backlight (), el primer método sirve para que arduino reconozca la pantalla Lcd, mientras que el segundo método hace posible de que esta pantalla emita luminosidad.

```
lcd.init();  
lcd.backlight();
```

*Figura 34. Inicialización de la pantalla LCD*

### 3.2.9.3. MÓDULO DE PROCESAMIENTO

Este módulo comprende el tratamiento de las entradas y salidas del sistema, así como también el uso de variables declaradas durante la creación y ejecución del programa.

- **Declaración de variables globales para el uso de millis ()**

```
unsigned long interval = 3000;  
unsigned long interval2 = 4000;  
unsigned long previousMillis1;  
unsigned long previousMillis2;
```

*Figura 35. Variables globales para el uso de millis ()*

Las variables declaradas como se muestra en la *Figura 35*, hacen posible de que se lleve un control de los mili segundos transcurridos durante la ejecución del programa en el microcontrolador, se estable un intervalo de tiempo para que se vuelva a ejecutar un conjunto de procedimientos posteriormente detallados, y además se actualiza el contador de millis con las variables previousMillis.

- **Loop ()**

El loop es un bucle infinito, también conocido como la función madre en el que las funciones hijas se ejecutarán continuamente, aquí se leerán las entradas y se activarán las salidas.



Para que el comportamiento del dispositivo sea fluido, se determinaron **2 macro procesos**, los cuales son: *El proceso que se ejecuta una vez por interrupción*, y *el proceso que se ejecutará constantemente* durante toda la vida del dispositivo.

Cuando se realiza más de un proceso en arduino se hace indispensable utilizar la función `millis ()` para “ejecutar multitareas”.

```
void loop() {
    unsigned long currentMillis = millis();
    // PROCESO EJECUCION SOLO SI SE HA LEIDO CODIGO QR
    if((unsigned long) (currentMillis - previousMillis1) >= interval){

        previousMillis1 = millis();
    }

    // PROCESO CONSTANTE
    if((unsigned long) (currentMillis - previousMillis2) >= interval2){

        previousMillis2 = millis();
    }
    |
    estado_pantalla(1);
}
```

*Figura 36. Estructuración general del loop ()*

Detalle del funcionamiento:

Se comienza declarando una variable `currentMillis` en la que se almacenará el tiempo en ese instante.

Mediante el uso de **if**, se comprueba si la diferencia de tiempos (`currentMillis ()` - `previousMillis ()`) es mayor al intervalo establecido anteriormente, es decir que debe ser mayor a 3 segundos, se ejecutará lo que esté dentro del **if** y se actualiza el tiempo de la variable `previousMillis ()`, caso contrario el programa continúa.

#### - **Proceso macro 1**

Este proceso se ejecutará sólo si el lector QR se encuentra en funcionamiento, una vez que este se accione, se desprenden un conjunto de funciones para el envío de peticiones al servidor, el tratamiento de la respuesta del servidor, la impresión por pantalla Lcd y el accionamiento final de la chapa eléctrica.

```

// PROCESO EJECUCION SOLO SI SE HA LEIDO CODIGO QR
if((unsigned long)(currentMillis - previousMillis) >= interval){
  bool tipo=0;
  bool accion;
  if(stringcomplete){ //INTERRUPCION
    Serial.println("CodigoQR LEIDO");
    codigoleido.remove(32,64);
    delay(500);
    Serial.println(codigoleido);
    if (client.connect(server, 80)) { //CONEXION CON EL SERVIDOR
      Serial.println("->Conectado al servidor");
      String path="prueba.php";
      estado_pantalla(2);
      peticion_servidor(path,codigoleido);
      delay(1500);
      accion=repuesta_servidor();
      accion_puerta(accion,0);
    }else{
      Serial.println("Conecion fallida");
      client.stop();
      client.flush();
    }
  }
  codigoleido = "";
  stringcomplete = false;
  previousMillis = millis();
}

```

*Figura 37. Código de ejecución si se detecta código QR*

Detalle del código:

Variables tipo (establece el tipo de macro proceso) y acción (establece si se debe o no accionar la chapa) son declaradas de tipo bool, posteriormente utilizadas para ser actualizadas y ser enviar por parámetros a la función acción\_puerta ().

La sentencia condicional if (stringcomplete) verifica si se ha leído algún código QR para proceder con las líneas de código. En este punto es importante decir que se hace uso del método remove (), el cual se utiliza para el tratamiento de string, aquí nos aseguramos de que sólo se envíe a la petición un código QR comprendido entre 32bits.

Una segunda sentencia condicional if (client.connect(server,80)) comprueba si hay conexión con el servidor web para posteriormente enviar la petición, leer la respuesta y accionar o no la chapa eléctrica.

En caso de que no se establezca conexión con el servidor, se detendrá la conexión de tipo cliente, y se limpiará el buffer de entrada de datos en el módulo Ethernet Shield.

- **Función de control para el lector QR.**

```
void loop() {  
  
  void serialEvent() {  
    while (Serial.available()>0) {  
      char c = (char)Serial.read();  
      codigoleido += c;  
      if (c == '\n') {  
        stringcomplete = true;  
      }  
    }  
  }  
}
```

Figura 38. Función de control para el lector QR

La función *serialEvent()*, es una función predeterminada de arduino, tiene como principal objeto captar un evento ejecutado por el puerto serie al mismo tiempo de que interrumpe la ejecución del resto del programa, dentro de esta función se hace uso del método *available()* el cual permite conocer si existen datos en el puerto serial.

El detalle de la función sería el siguiente:

Mientras existan datos en el puerto serial, se leerá cada bit mediante el método *Serial.read()* y se lo almacenará en una variable de tipo char denominada “c”, el siguiente paso será concatenar esos bits en una variable global de tipo String denominada “codigoleido”, por último, se verifica si el bit leído es un retorno de carro ‘\n’, en caso de así sea, se da por entendido de que la cadena ha sido leída por completo, cambiando el estado de la variable “stringcomplete=true”.

- **Función para realizar peticiones al servidor**

```
void peticion_servidor(String path,String codigoleido){  
  Serial.println("GET /arduino/"+path+"?valor="+codigoleido+"&aula=B1%20-%20203");  
  client.print("GET /arduino/"+path+"?valor="+codigoleido+"&aula=B1%20-%20203");  
  client.println(" HTTP/1.1");  
  client.println("Host: 52.233.72.68");  
  client.println("Connection: close");  
  client.println();  
  Serial.println(F("->Petición enviada"));  
}
```

Figura 39. Función petición servidor

La función *petición\_servidor()* pasa por parámetros el Path o el nombre del archivo php del servidor que contiene la sentencia mysql a ejecutar, de la misma manera envía el código leído y el nombre del aula (permite identificar los qr para el aula enviada).

## - Función para recibir la respuesta del servidor

```
bool respuesta_servidor(){
    String resp="";
    bool puerta=0;
    while (client.available()) {
        char c = client.read();
        resp += c;
        fin = true;
    }

    if(fin){
        int longitud = resp.length();
        int posicion = resp.indexOf("valor=");
        estado = "";
        for(int i = posicion + 6; i< longitud; i++){
            if(resp[i] == ';'){
                i= longitud;
            }
            else
                estado += resp[i];
        }
        fin=false;
        Serial.println("----- estado:" +estado+ " \n");
        if(estado == "A"){
            puerta=1;
        }
        estado="";
    }

    if(!client.connected()){
        Serial.println("Desconectado");
        client.stop();
        client.flush();
    }

    return puerta;
}
```

*Figura 40. Función respuesta servidor*

Hace uso del método `available ()` para comprobar si existe o no respuesta de la petición enviada por el cliente, posteriormente se cambia el estado de la variable booleana `fin =true`, para establecer que se ha terminado de leer todo el buffer.

Dentro de una sentencia condicional `if()` que se ejecuta una vez el valor de `fin=true`, se procede a tratar al string recibido como respuesta mediante el uso de la sentencia `for ()`, para recorrer el string , se almacena en una variable `estado` la respuesta que queremos para luego mediante una nueva sentencia `if` comprobar si el resultado es igual a “A” si es así se cambia el estado de la variable booleana `puerta` a 1 , por último se retorna el valor de `puerta`, se consulta si el cliente ha sido desconectado, de ser así se detiene la comunicación con el servidor, y se limpia el buffer.

- **Función para el accionamiento de la chapa eléctrica**

```
//ACCIONAMIENTO DE LA PUERTA//
void accion_puerta(bool accion,bool tipo){
  if(accion==1 and tipo==0){
    estado_pantalla(3);
    Serial.println("Abriendo puerta");
    digitalWrite(5,LOW);
    delay(3000);
    digitalWrite(5,HIGH);
  }else if(accion==0 and tipo==0){
    estado_pantalla(4);
    Serial.println("Acceso incorrecto");
    delay(2000);
  }

  if(accion==1 and tipo==1){
    estado_pantalla(3);
    Serial.println("Abriendo puerta");
    digitalWrite(5,LOW);
    delay(3000);
    digitalWrite(5,HIGH);
  }
}
```

*Figura 41. Función para el accionar de la cerradura eléctrica*

Esta función recibe por parámetros los valores de las variables acción y tipo que servirán para diferenciar entre los tipos de macroprocesos que acceden a la función y la orden que estos están enviando.

Para el caso de la primera sentencia condicional se verifica si la acción==1 (respuesta de servidor válida) y si el tipo==0 (primer macro proceso), si ambas condiciones se cumplen se envía una señal digital baja al pin 5 declarado en el setup como salida en alta, adicional a esto se le añade la función estado\_pantalla (3) con paso de parámetro 3, para que como su nombre lo indica, muestre el estado del dispositivo por pantalla. En caso de que la condición verdadera no se cumpla se consulta nuevamente esta vez preguntando si ambos valores de las variables son 0, que indica que no se tuvo respuesta acertada del servidor. Y que corresponde a la petición de tipo primer macro proceso.

Para el segundo caso se verifica si la acción==1 que indica que la respuesta del servidor es positiva, pero además se comprueba que el tipo==1 indicando que quien solicita acceso al método es el segundo macroproceso.

#### - Proceso macro 2

Este procedimiento abarca todas las funciones del primer proceso sin considerar la interrupción generada por el lector QR. A diferencia del primer proceso macro, se va a ejecutar cada 4 segundos enviando una petición hacia el archivo Php correspondiente en el servidor, el mismo que consultará por el estado de la puerta del aula, este proceso está dirigido al accionamiento de la cerradura eléctrica en cualquier momento y desde cualquier lugar desde la app móvil y desde el usuario técnico docente.

```
// PROCESO CONSTANTE
if((unsigned long)(currentMillis - previousMillis2) >= interval2){
  bool tipo=1;
  bool accion;
  if (client.connect(server, 80) {          //CONEXION CON EL SERVIDOR
    Serial.println("->Conectado al servidor");
    String path="puerta.php";
    peticion_servidor(path,codigoleido);
    delay(1500);
    accion=repuesta_servidor();
    accion_puerta(accion,tipo);
    codigoleido = "";
    stringcomplete = false;
  }else{
    Serial.println("Conecion fallida");
    client.stop();
    client.flush();
  }

  previousMillis2 = millis();
}
estado_pantalla(1);
}
```

Figura 42. Código de ejecución petición constante

- **Función para el cambio de estado de la pantalla LCD**

```
void estado_pantalla(int val){
    switch(val){
        case 1:
            lcd.setCursor(0,0);
            lcd.print("  SEGURIDAD  ");
            lcd.setCursor(0,1);
            lcd.print(" =====GIAN=====");
            break;
        case 2:
            lcd.setCursor(0, 0);
            lcd.print("  COMPROBANDO  ");
            lcd.setCursor(0, 1);
            lcd.print("      QR      ");
            break;
        case 3:
            lcd.setCursor(0, 0);
            lcd.print("  ACCESO  ");
            lcd.setCursor(0, 1);
            lcd.print("  CONCEDIDO  ");
            break;
        case 4:
            lcd.setCursor(0, 0);
            lcd.print("  ACCESO  ");
            lcd.setCursor(0, 1);
            lcd.print("  INCORRECTO  ");
            break;
    }
}
```

*Figura 43. Función estados pantalla Lcd*

Cada vez que se llame a la función se deberá pasar por parámetro un número de tipo entero, si éste es comprendido entre los casos que contiene la sentencia switch, la pantalla emitirá un mensaje distinto. Cabe recalcar que el primer estado o case 1, es llamado en el loop desde el inicio como presentación inicial del sistema.

### 3.2.10. MODELADO DE DATOS DEL SISTEMA EN GENERAL

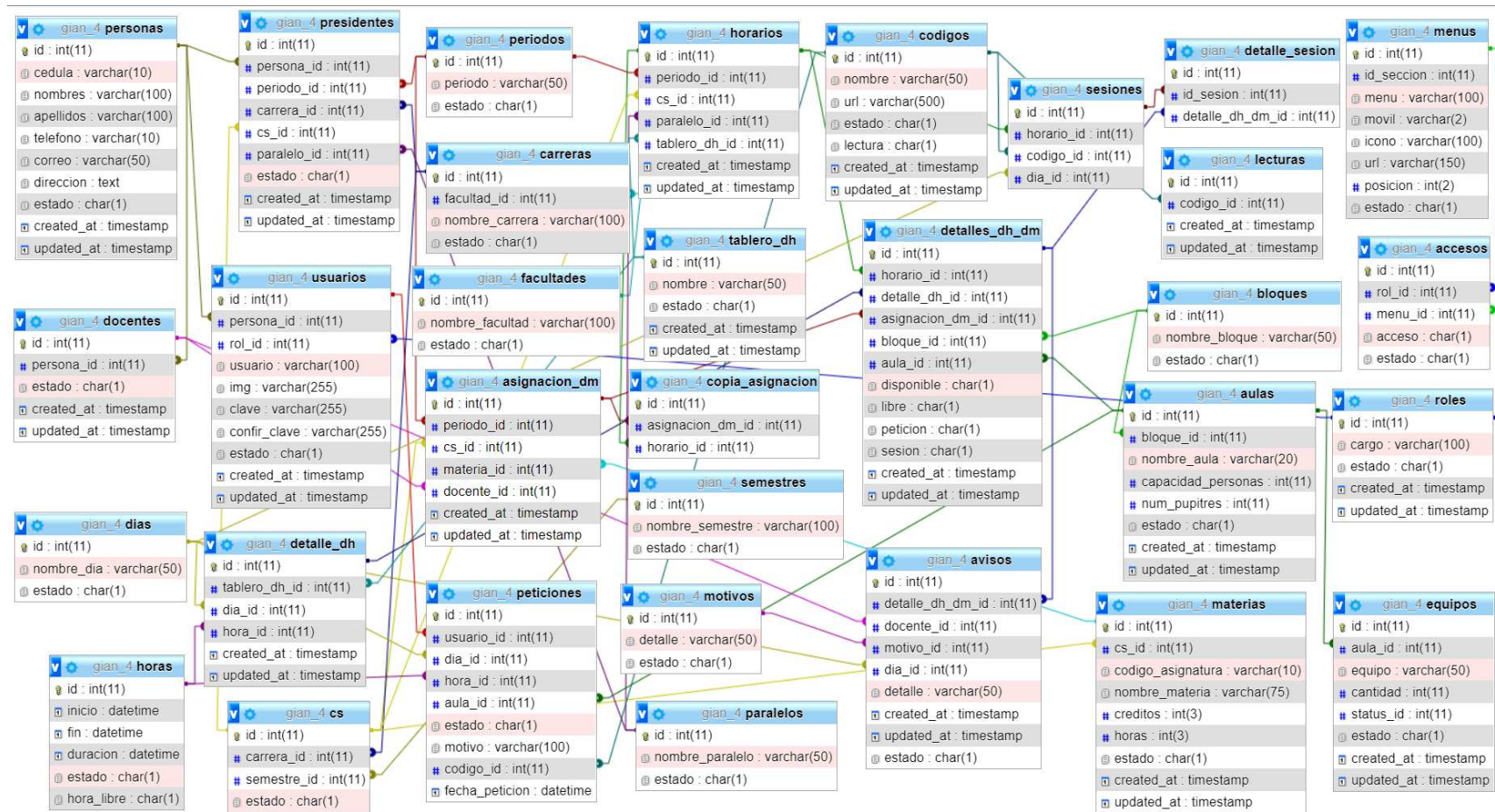


Figura 44. Modelado de datos del sistema en General



### 3.2.11. MODELADO DE DATOS – DISPOSITIVO DE SEGURIDAD

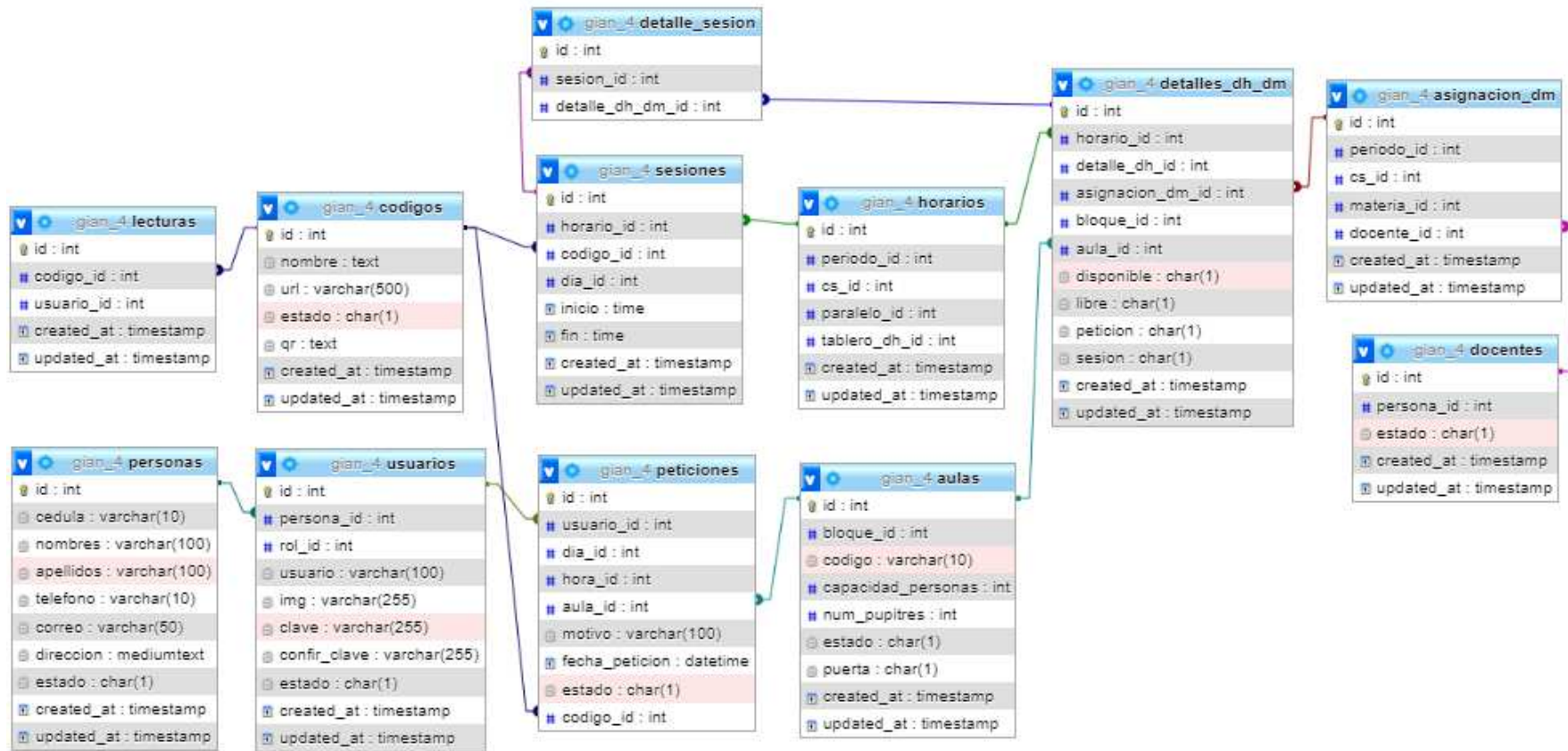


Figura 45. Modelado de datos – Dispositivo de seguridad

### 3.2.11.1. DICCIONARIO DE DATOS

#### ACCESOS

<b>Descripción General:</b>			
<i>Contiene los datos de acceso de usuarios, ingresando a los diversos menús.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>rol_id</b>	Integer	11	Clave foránea de la tabla roles.
<b>menu_id</b>	Integer	11	Clave foránea de la tabla menus.
<b>acceso</b>	Char	1	Estado de acceso. Activo (A), Inactivo (I).
<b>estado</b>	Char	1	Estado de la tabla accesos. Activo (A), inactivo (I).

Tabla 14. Diccionario de datos tabla accesos

#### ASIGNACION\_DM

<b>Descripción General:</b>			
<i>Contiene los datos de las tablas intermedias: periodos, cs, materias y docentes.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>periodo_id</b>	Integer	11	Clave foránea de la tabla periodos.
<b>cs_id</b>	Integer	11	Clave foránea de la tabla cs.
<b>materia_id</b>	Integer	1	Clave foránea de la tabla materias.
<b>docente_id</b>	Integer	1	Clave foránea de la tabla docentes.
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

Tabla 15. Diccionario de datos tabla asignación\_dm

#### AULAS

<b>Descripción General:</b>			
<i>Contiene los datos de las aulas.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>bloque_id</b>	Integer	11	Clave foránea de la tabla bloques.
<b>codigo</b>	Varchar	20	Código del aula.
<b>capacidad_personas</b>	Integer	11	Número de personas que pueden ingresar al aula.
<b>num_pupitres</b>	Integer	11	Número de pupitres que posee el aula.
<b>estado</b>	Char	1	Estado de la tabla aulas. Activo (A), inactivo (I).
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

Tabla 16. Diccionario de datos tabla aulas

## AVISOS

<b>Descripción General:</b>			
<i>Contiene los datos de los avisos de docentes al notificar faltas.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>detalle_dh_dm_id</b>	Integer	11	Clave foránea de la tabla detalles_dh_dm.
<b>docente_id</b>	Integer	11	Clave foránea de la tabla docentes.
<b>motivo_id</b>	Integer	11	Clave foránea de la tabla motivos.
<b>dia_id</b>	Integer	11	Clave foránea de la tabla días.
<b>detalle</b>	Varchar	50	Motivo del docente, por el cual falta a clases.
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.
<b>estado</b>	Char	1	Estado de la tabla avisos. Activo (A), inactivo (I).

*Tabla 17. Diccionario de datos tabla avisos*

## BLOQUES

<b>Descripción General:</b>			
<i>Contiene los datos de los bloques de la facultad.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>facultad_id</b>	Integer	11	Clave foránea de la tabla facultades.
<b>nombre_bloque</b>	Varchar	50	Nombre del bloque.
<b>estado</b>	Char	1	Estado de la tabla bloques. Activo (A), inactivo (I).

*Tabla 18. Diccionario de datos tabla bloques*

## CARRERAS

<b>Descripción General:</b>			
<i>Contiene los datos de las carreras universitarias.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>facultad_id</b>	Integer	11	Clave foránea de la tabla facultades.
<b>nombre_carrera</b>	Varchar	100	Nombre de la carrera.
<b>estado</b>	Char	1	Estado de la tabla carreras. Activo (A), inactivo (I).

*Tabla 19. Diccionario de datos tabla carreras*

## CODIGOS

<b>Descripción General:</b>			
<i>Contiene los datos de los códigos QR, que serán empleados para aperturar las aulas.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>nombre</b>	Text		Nombre de la imagen del código QR.
<b>url</b>	Varchar	50	Ruta donde se encuentra el código QR.
<b>estado</b>	Char	1	Estado de la tabla códigos. Activo (A), inactivo (I).
<b>qr</b>	Text	-	Estado de lectura del código QR. Activo (A), inactivo (I).
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 20. Diccionario de datos tabla códigos*

## COPIA\_ASIGNACION

<b>Descripción General:</b>			
<i>Contiene los datos de las tablas intermedias: asignación dm y horarios.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>asignacion_dm_id</b>	Integer	11	Clave foránea de la tabla asignación dm.
<b>horario_id</b>	Integer	11	Clave foránea de la tabla horarios.

*Tabla 21. Diccionario de datos tabla copia\_asignacion*

## COPIA\_SESION

<b>Descripción General:</b>			
<i>Contiene los datos de las tablas intermedias: horarios, detalles_dh_dm y dias.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>horario_id</b>	Integer	11	Clave foránea de la tabla horarios.
<b>detalles_dh_dm_id</b>	Integer	11	Clave foránea de la tabla detalles_dh_dm.
<b>dia_id</b>	Integer	11	Clave foránea de la tabla dias.
<b>uso</b>	Char	1	Estado de la lectura. Sí (S), No (N).

*Tabla 22. Diccionario de datos tabla copia\_sesion*

## CS

<b>Descripción General:</b>			
<i>Contiene los datos de las tablas intermedias: carreras y semestres.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>carrera_id</b>	Integer	11	Clave foránea de la tabla carreras.
<b>semestre_id</b>	Integer	11	Clave foránea de la tabla semestres.
<b>estado</b>	Char	1	Estado de la tabla cs. Activo (A), inactivo (I).

*Tabla 23. Diccionario de datos tabla cs*

## DETALLES\_DH\_DM

<b>Descripción General:</b>			
<i>Contiene los datos de las tablas intermedias: horarios, detalle_dh, asignación_dm, bloques y aulas.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria
<b>horario_id</b>	Integer	11	Clave foránea de la tabla carreras.
<b>detalle_dh_id</b>	Integer	11	Clave foránea de la tabla semestres.
<b>asignacion_dm_id</b>	Integer	11	Clave foránea de la tabla asignación_dm.
<b>bloque_id</b>	Integer	11	Clave foránea de la tabla bloques.
<b>aula_id</b>	Integer	11	Clave foránea de la tabla aulas.
<b>disponible</b>	Char	1	Estado de la tabla detalles_dh_dm. Sí (S), No (N).
<b>libre</b>	Char	1	Estado de apertura de un aula. Sí (S), No (N).
<b>peticion</b>	Char	1	Estado que indica si se realizó una petición en ese horario. Si (S), No(N).
<b>sesion</b>	Char	1	Estado que indica la sesión. Sí (S), No (N).
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 24. Diccionario de datos tabla detalles\_dh\_dm*

## DETALLE\_DH

<b>Descripción General:</b> <i>Contiene los datos de las tablas intermedias: tablero_dh, dias y horas.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>tablero_dh_id</b>	Integer	11	Clave foránea de la tabla tablero_dh.
<b>dia_id</b>	Integer	11	Clave foránea de la tabla días.
<b>hora_id</b>	Integer	11	Clave foránea de la tabla horas.
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

Tabla 25. Diccionario de datos tabla detalle\_dh

## DETALLE\_SESION

<b>Descripción General:</b> <i>Contiene los datos de las tablas intermedias: sesiones y detalles_dh_dm.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>sesion_id</b>	Integer	11	Clave foránea de la tabla sesiones.
<b>detalle_dh_dm_id</b>	Integer	11	Clave foránea de la tabla detalles_dh_dm.

Tabla 26. Diccionario de datos tabla detalle\_sesion

## DIAS

<b>Descripción General:</b> <i>Contiene los datos de los días.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>nombre_dia</b>	Varchar	50	Nombre del día.
<b>estado</b>	Char	1	Estado de la tabla días. Activo (A), Inactivo (I).

Tabla 27. Diccionario de datos tabla dias

## DOCENTES

<b>Descripción General:</b> <i>Contiene los datos de los docentes, de la facultad.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>persona_id</b>	Integer	11	Clave foránea de la tabla personas.
<b>estado</b>	Char	1	Estado de la tabla docentes. Activo (A), Inactivo (I).
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

Tabla 28. Diccionario de datos tabla docentes

## EQUIPOS

<b>Descripción General:</b>			
<i>Contiene los datos de los equipos informáticos que están en las aulas.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>aula_id</b>	Integer	11	Clave foránea de la tabla aulas.
<b>equipo</b>	Varchar	50	Nombre del equipo.
<b>status_id</b>	Integer	11	Estado.
<b>estado</b>	Char	1	Estado de la tabla equipos. Activo (A), Inactivo (I).
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 29. Diccionario de datos tabla equipos*

## FACULTADES

<b>Descripción General:</b>			
<i>Contiene los datos de las facultades.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>nombre_facultad</b>	Varchar	50	Nombre de la facultad.
<b>estado</b>	Char	1	Estado de la tabla facultades. Activo (A), Inactivo (I).

*Tabla 30. Diccionario de datos tabla facultades*

## HORARIOS

<b>Descripción General:</b>			
<i>Contiene los datos de las tablas intermedias: periodos, cs, paralelos y tablero dh.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>periodo_id</b>	Integer	11	Clave foránea de la tabla periodos.
<b>cs_id</b>	Integer	11	Clave foránea de la tabla cs.
<b>paralelo_id</b>	Integer	11	Clave foránea de la tabla paralelos.
<b>tablero_dh_id</b>	Integer	11	Clave foránea de la tabla tablero_dh.
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 31. Diccionario de datos tabla horarios*

## HORAS

<b>Descripción General:</b>			
<i>Contiene los datos de las horas, que serán empleadas en la creación de los horarios.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>inicio</b>	Time	-	Hora de inicio.
<b>fin</b>	Time	-	Hora de fin.
<b>duracion</b>	Char	5	Duración de la hora.
<b>estado</b>	Char	1	Estado de la tabla horas. Activo (A), Inactivo (I).
<b>hora_libre</b>	Char	1	Estado para saber si una hora está libre. S: (Hora libre), A (Activa), I (Inactiva).

*Tabla 32. Diccionario de datos tabla horas*

## LECTURAS

<b>Descripción General:</b>			
<i>Contiene los datos de las lecturas de los códigos QR al ingresar a las distintas aulas.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>codigo_id</b>	Integer	11	Clave foránea de la tabla códigos.
<b>usuario_id</b>	Integer	11	Clave foránea de la tabla usuarios.
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 33. Diccionario de datos tabla lecturas*

## MATERIAS

<b>Descripción General:</b>			
<b>Contiene los datos de las materias, que serán empleadas en la creación de los horarios.</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>cs_id</b>	Integer	11	Clave foránea de la tabla cs.
<b>codigo_asignatura</b>	Varchar	10	Código de la asignatura.
<b>nombre_materia</b>	Varchar	75	Nombre de la materia.
<b>creditos</b>	Integer	3	Número de créditos.
<b>horas</b>	Integer	3	Número de horas que tiene la asignatura.
<b>estado</b>	Char	1	Estado de la tabla materias. Activo (A), inactivo (I).
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 34. Diccionario de datos tabla materias*



## MENUS

<b>Descripción General:</b>			
<i>Contiene los datos de los menús empleados en los sistemas web y móvil.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>id_seccion</b>	Integer	11	Número de sección.
<b>menu</b>	Varchar	100	Nombre del menú.
<b>movil</b>	Varchar	2	Identifica a que sistema pertenece. Móvil (s), Web (null).
<b>icono</b>	Varchar	100	Nombre del ícono.
<b>url</b>	Varchar	150	Enlace de la página.
<b>posicion</b>	Integer	2	Posición donde estará ubicado el menú.
<b>estado</b>	Char	1	Estado de la tabla menús. Activo (A), inactivo (I).

*Tabla 35. Diccionario de datos tabla menús*

## MOTIVOS

<b>Descripción General:</b>			
<i>Contiene los datos de los motivos que se emplean para notificar un aviso.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>detalle</b>	Varchar	50	Detalle del motivo.
<b>estado</b>	Char	1	Estado de la tabla motivos. Activo (A), Inactivo (I).

*Tabla 36. Diccionario de datos tabla motivos*

## PARALELOS

<b>Descripción General:</b>			
<i>Contiene los datos de los paralelos, que serán empleados en la creación de horarios.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>nombre_paralelo</b>	Varchar	50	Nombre del paralelo.
<b>estado</b>	Char	1	Estado de la tabla paralelos. Activo (A), Inactivo (I).

*Tabla 37. Diccionario de datos tabla paralelos*

## PERIODOS

<b>Descripción General:</b>			
<i>Contiene los datos de los periodos, que serán empleados en la creación de horarios.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>periodo</b>	Varchar	50	Periodo académico.
<b>estado</b>	Char	1	Estado de la tabla periodos. Activo (A), Inactivo (I).

*Tabla 38. Diccionario de datos tabla periodos*

## PERSONAS

<b>Descripción General:</b>			
<i>Contiene los datos de las personas.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>cedula</b>	Varchar	100	Número de cédula de la persona.
<b>nombres</b>	Varchar	100	Nombres de la persona.
<b>apellidos</b>	Varchar	100	Apellidos de la persona.
<b>telefono</b>	Varchar	10	Teléfono de la persona.
<b>correo</b>	Varchar	50	Correo electrónico de la persona.
<b>direccion</b>	Text	-	Dirección de la persona.
<b>estado</b>	Char	1	Estado. Activo (A), Inactivo (I).
<b>created_at</b>	Timestamp	-	Fecha que se almacena.
<b>updated_at</b>	Timestamp	-	Fecha que se actualiza.

*Tabla 39. Diccionario de datos tabla personas*

## PETICIONES

<b>Descripción General:</b>			
<i>Contiene los datos de las peticiones que se realizan en el Bloque 1.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>usuario_id</b>	Integer	11	Clave foránea de la tabla usuarios.
<b>dia_id</b>	Integer	11	Clave foránea de la tabla días.
<b>hora_id</b>	Integer	11	Clave foránea de la tabla horas.
<b>aula_id</b>	Integer	11	Clave foránea de la tabla aulas.
<b>estado</b>	Char	1	Estado. A (Activo), I (Inactivo).
<b>motivo</b>	Varchar	100	Motivo de la petición.
<b>codigo_id</b>	Integer	11	Clave foránea de la tabla códigos.
<b>fecha_peticion</b>	Datetime	-	Fecha en que se realiza la petición.

*Tabla 40. Diccionario de datos tabla peticiones*

## PRESIDENTES

<b>Descripción General:</b>			
<i>Contiene los datos de los presidentes de curso.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>persona_id</b>	Integer	11	Clave foránea de la tabla personas.
<b>periodo_id</b>	Integer	11	Clave foránea de la tabla periodos.
<b>carrera_id</b>	Integer	11	Clave foránea de la tabla carreras.
<b>cs_id</b>	Integer	11	Clave foránea de la tabla cs.
<b>paralelo_id</b>	Integer	11	Clave foránea de la tabla paralelos.
<b>estado</b>	Char	1	Estado. Activo (A), Inactivo (I).
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 41. Diccionario de datos tabla presidentes*

## ROLES

<b>Descripción General:</b> <i>Contiene los datos de los roles.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>cargo</b>	Varchar	100	Nombre del cargo.
<b>estado</b>	Char	1	Estado de la tabla roles. Activo (A), Inactivo (I).
<b>created_at</b>	Timestamp	-	Fecha que se almacena.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 42. Diccionario de datos tabla roles*

## SEMESTRES

<b>Descripción General:</b> <i>Contiene los datos de los semestres, que serán empleados en la creación de los horarios.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>nombre_semestre</b>	Varchar	100	Nombre del semestre.
<b>estado</b>	Char	1	Estado de la tabla semestres. Activo (A), Inactivo (I).

*Tabla 43. Diccionario de datos tabla semestres*

## SESIONES

<b>Descripción General:</b> <i>Contiene los datos de las sesiones, que son creadas a partir de los horarios de clases.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>horario_id</b>	Integer	11	Clave foránea de la tabla horarios.
<b>codigo_id</b>	Integer	11	Clave foránea de la tabla códigos.
<b>dia_id</b>	Integer	11	Clave foránea de la tabla días.
<b>inicio</b>	Time	-	Hora de inicio.
<b>fin</b>	Time	-	Hora de fin.
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 44. Diccionario de datos tabla sesiones*

## TABLERO\_DH

<b>Descripción General:</b>			
<i>Contiene los datos de los tableros que se emplean para crear horarios de clases.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>periodo_id</b>	Integer	11	Clave foránea de la tabla periodos.
<b>carrera_id</b>	Integer	11	Clave foránea de la tabla carreras.
<b>semestre_id</b>	Integer	11	Clave foránea de la tabla semestres.
<b>nombre</b>	Varchar	50	Nombre del tablero.
<b>uso</b>	Char	1	Estado que indica si el tablero está en uso. Si (S), No (N).
<b>estado</b>	Char	1	Estado de la tabla tablero_dh. Activo (A), Inactivo (I).
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 45. Diccionario de datos tabla tablero\_dh*

## USUARIOS

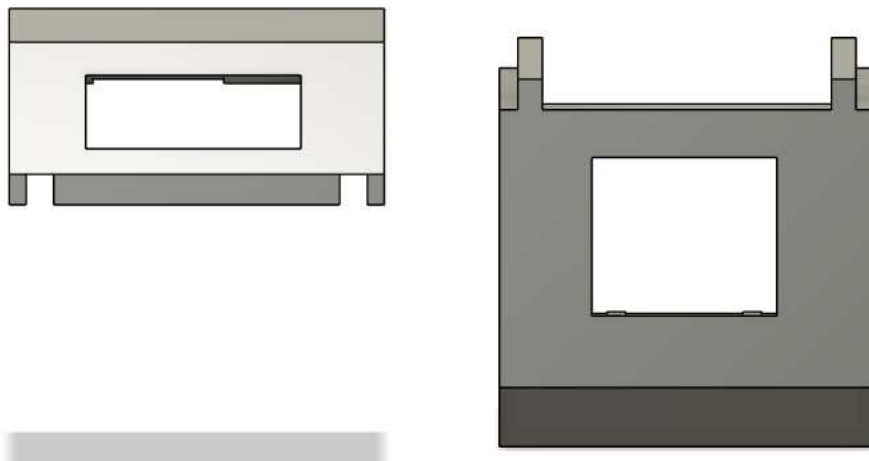
<b>Descripción General:</b>			
<i>Contiene los datos de los usuarios del sistema que servirán para otorgar privilegios.</i>			
<b>Campo</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción.</b>
<b>id</b>	Integer	11	Clave primaria.
<b>persona_id</b>	Integer	11	Clave foránea de la tabla personas.
<b>rol_id</b>	Integer	11	Clave foránea de la tabla roles.
<b>usuario</b>	Varchar	100	Nombre de usuario.
<b>img</b>	Varchar	255	Imagen del usuario.
<b>clave</b>	Varchar	255	Clave del usuario.
<b>confir_clave</b>	Varchar	255	Clave de confirmación.
<b>estado</b>	Char	1	Estado de la tabla usuarios. Activo (A), Inactivo (I).
<b>created_at</b>	Timestamp	-	Fecha de almacenamiento del registro.
<b>updated_at</b>	Timestamp	-	Fecha de actualización del registro.

*Tabla 46. Diccionario de datos tabla usuarios*

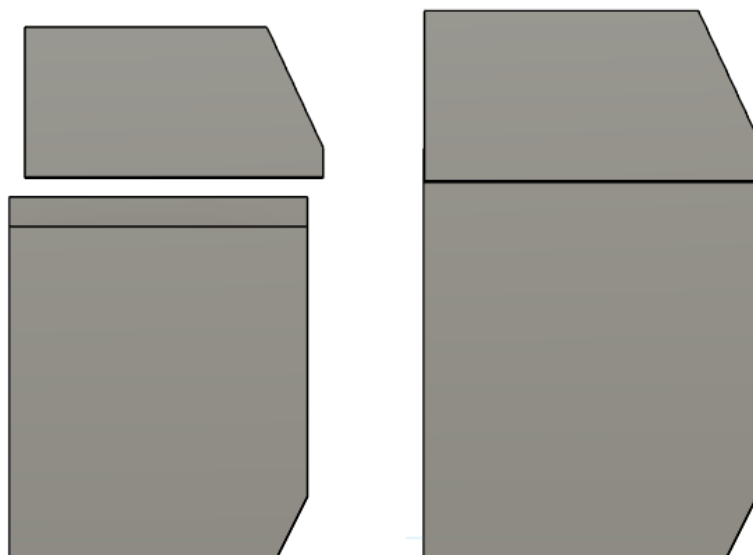
### 3.7. DISEÑO Y CONSTRUCCIÓN DEL CONTENEDOR DEL DISPOSITIVO

Para la construcción del dispositivo se hizo uso del software Fusion 360, mediante el cual se llevó a cabo el diseño en 3D.

La estructura en general comprende 2 partes, una inferior y otra superior, con el fin de que se pueda visualizar y manipular las conexiones eléctricas dentro del contenedor. Una vez instalado todos los componentes eléctricos en la parte inferior, se procede a colocar la parte superior que contendrá la pantalla Lcd.



*Figura 46. Vista frontal, parte superior del diseño en 3D*



*Figura 47. Vista lateral, parte superior del diseño en 3D*

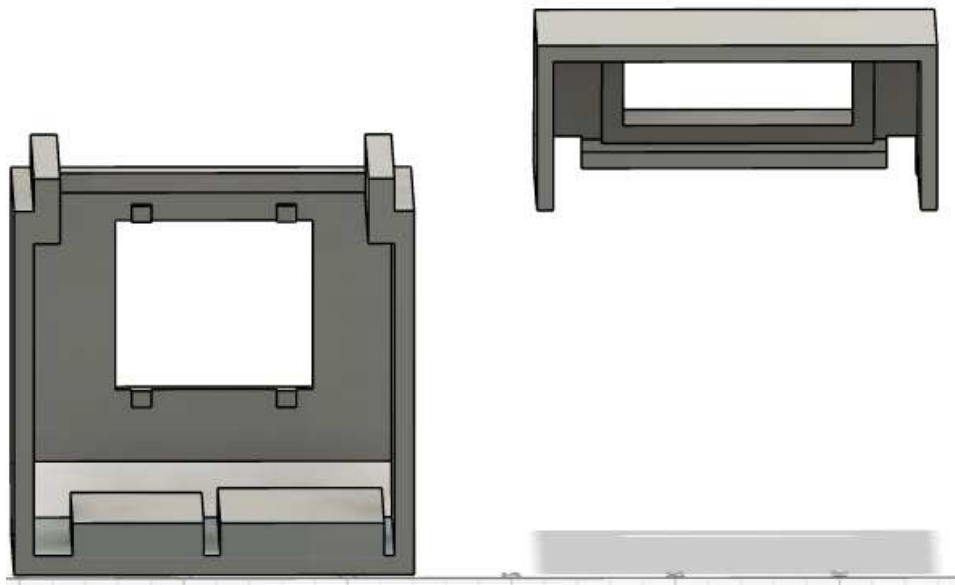


Figura 48. Vista posterior, parte superior del diseño en 3D

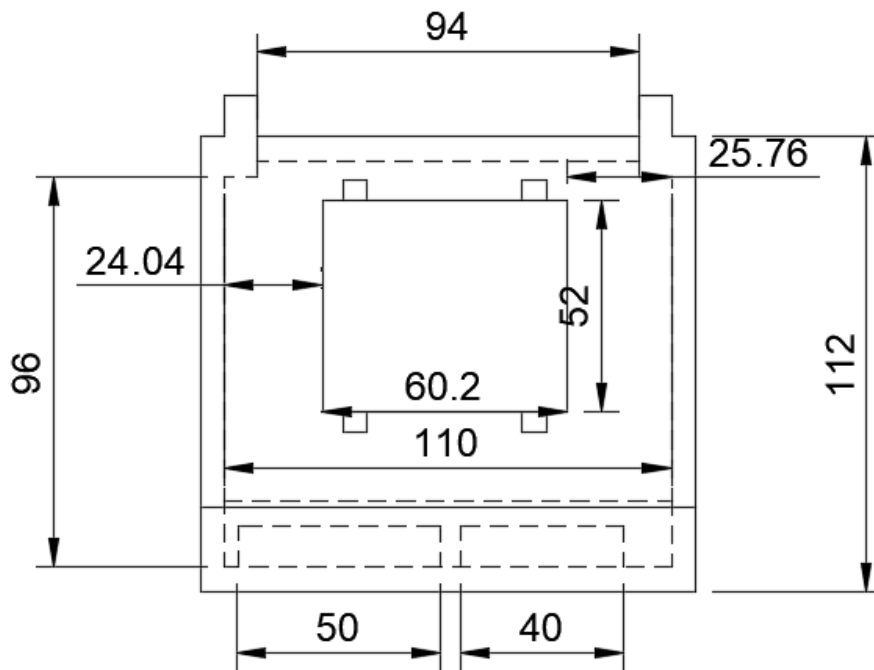


Figura 49. Medidas de la vista posterior, parte superior del diseño en 3D

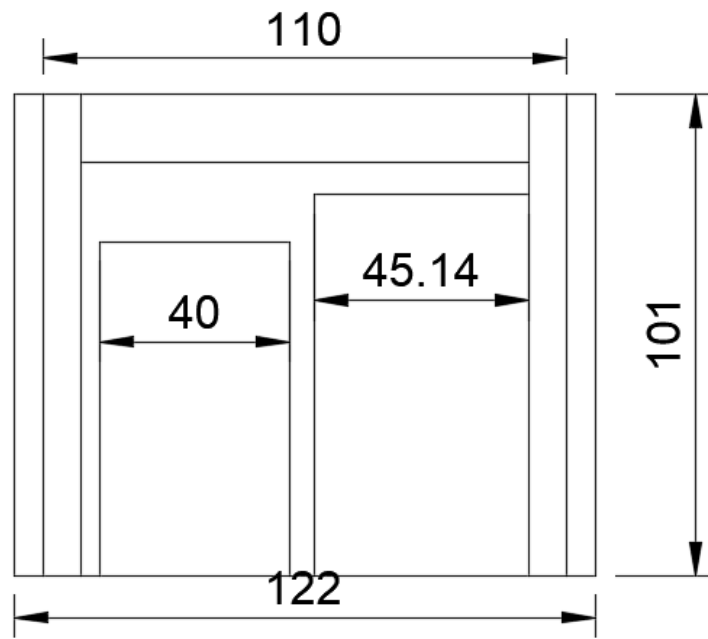


Figura 50. Medidas de la vista superior, parte superior del diseño en 3D

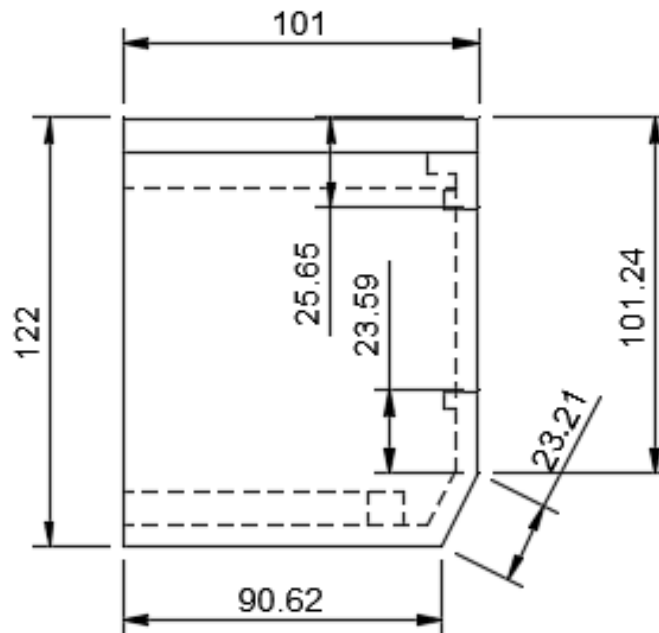


Figura 51. Medidas de la vista lateral, parte inferior del diseño en 3D

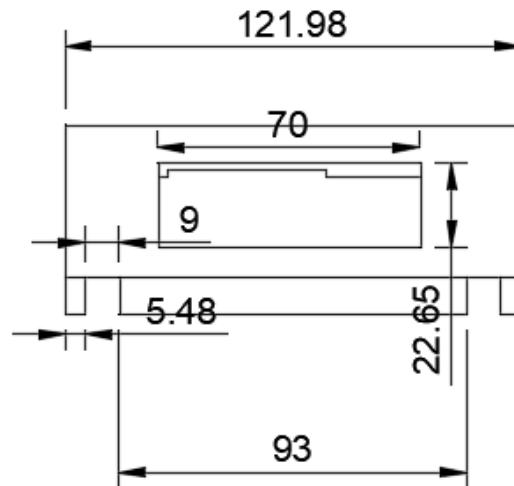


Figura 52. Medidas de la vista frontal, parte superior del diseño en 3D

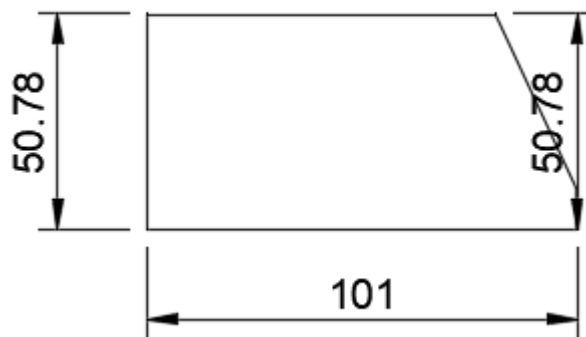


Figura 53. Medidas de la vista lateral, parte superior del diseño en 3D

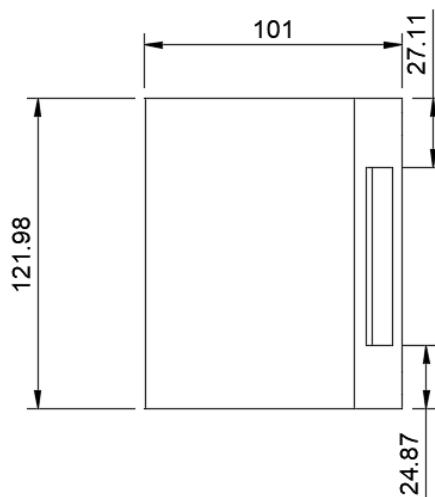
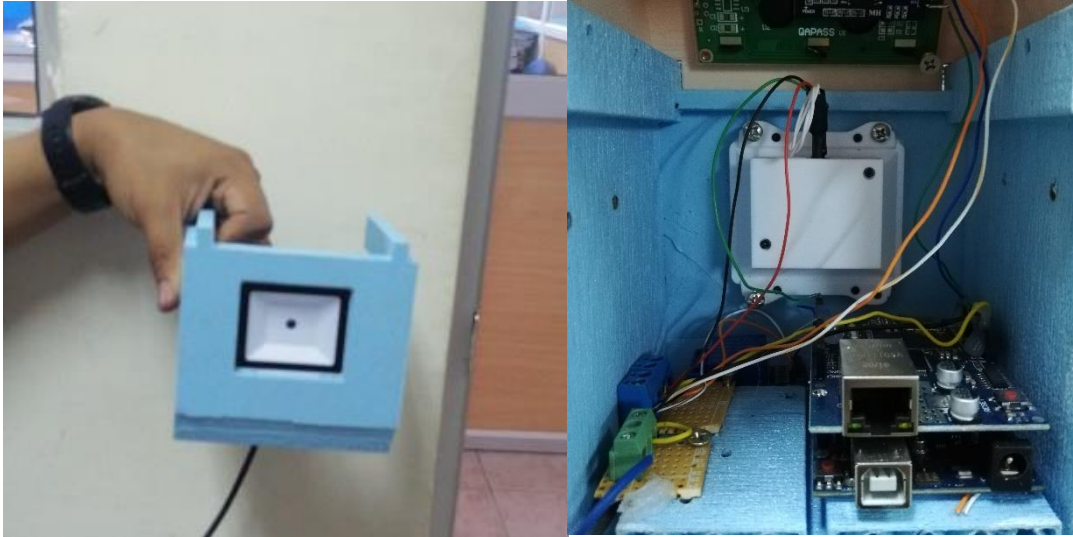
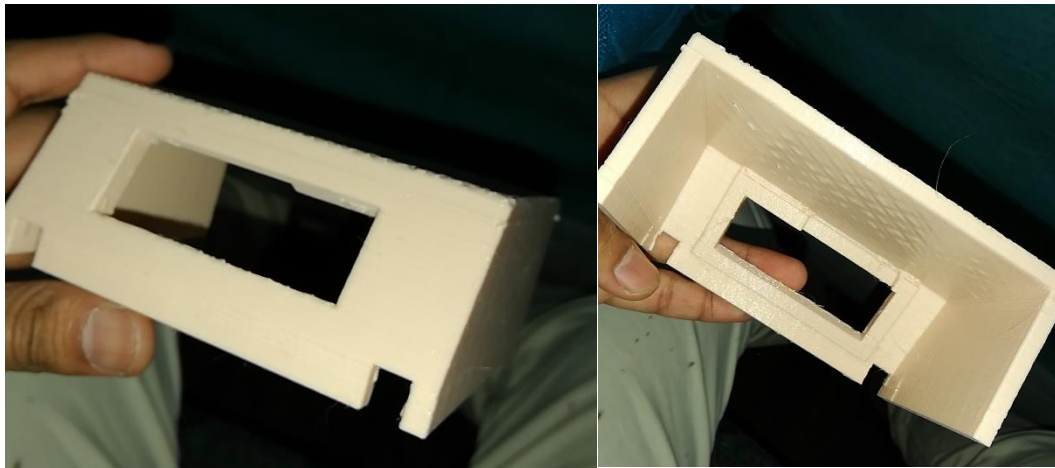


Figura 54. Medidas de la vista superior, parte superior del diseño en 3D





*Figura 55. Impresión en 3D de la parte inferior del diseño*



*Figura 56. Impresión en 3D de la parte superior del diseño*

### 3.4. PRUEBAS

<b>CASO DE PRUEBA No.</b>	001	
<b>CASO DE ESTUDIO:</b>	Validación del código QR	
<b>OBJETIVO DE LA PRUEBA:</b>	Comprobar que el código QR leído se encuentre en la base de datos.	
<p><b>CONDICIONES:</b> El código leído por el dispositivo debe coincidir con el que se encuentra habilitado en la base de datos.</p> <p><b>PASOS A SEGUIR:</b></p> <ol style="list-style-type: none"> <li>1. El usuario escanea el código QR.</li> <li>2. Ingresa al aula.</li> </ol>		
<b>RESULTADOS DE LA PRUEBA</b>		
<b>RESULTADOS ESPERADOS</b>	<b>EVALUACIÓN</b>	
El usuario ingresa de manera correcta, se emitirá un mensaje por la pantalla Lcd indicando que el acceso es correcto.	Exitoso <input checked="" type="checkbox"/>	Fallido <input type="checkbox"/>

Tabla 47. Prueba 1 de funcionamiento para validación de código QR

<b>CASO DE PRUEBA No.</b>	002	
<b>CASO DE ESTUDIO:</b>	Validación del código QR	
<b>OBJETIVO DE LA PRUEBA:</b>	Comprobar que el código QR leído se encuentre habilitado para el horario en el que se desea ingresar al aula.	
<p><b>CONDICIONES:</b> El código leído por el dispositivo debe estar en la tabla códigos con un estado activo, el id del código leído debe estar asignado en la tabla sesiones, y dicha sesión debe estar relacionada en la tabla detalle_sesion con un id de la tabla detalle_dh_dm la cual contiene toda la información de los horarios ocupados y desocupados en el campo disponible.</p> <p><b>PASOS A SEGUIR:</b></p> <ol style="list-style-type: none"> <li>1. El usuario escanea el código QR.</li> <li>2. Ingresa al aula.</li> </ol>		

<b>RESULTADOS DE LA PRUEBA</b>	
<b>RESULTADOS ESPERADOS</b>	<b>EVALUACIÓN</b>
El usuario ingresa de manera correcta, se emite un mensaje por el puerto COM indicando que el acceso es exitoso.	Exitoso <input checked="" type="checkbox"/> Fallido <input type="checkbox"/>

*Tabla 48. Prueba 2 de funcionamiento para validación de código QR*

<b>CASO DE PRUEBA No.</b>	003
<b>CASO DE ESTUDIO:</b>	Validación del código QR
<b>OBJETIVO DE LA PRUEBA:</b>	Comprobar que el código QR leído esté asignado al aula que se quiere acceder en el respectivo horario.
<b>CONDICIONES:</b> Se consideran todas las validaciones del caso de prueba No. 2, adicionando que el aula destinada al dispositivo, se encuentre en estado Inactivo en la tabla aulas, y que además dicha aula se encuentre relacionada a un registro de la tabla detalles_dh_dm.	
<b>PASOS A SEGUIR:</b>	
<ol style="list-style-type: none"> <li>3. El usuario escanea el código QR.</li> <li>4. Ingresa al aula.</li> </ol>	
<b>RESULTADOS DE LA PRUEBA</b>	
<b>RESULTADOS ESPERADOS</b>	<b>EVALUACIÓN</b>
El usuario ingresa de manera correcta, se emite un mensaje por el puerto COM indicando que el acceso es exitoso.	Exitoso <input checked="" type="checkbox"/> Fallido <input type="checkbox"/>

*Tabla 49. Prueba 3 de funcionamiento para validación de código QR*

<b>CASO DE PRUEBA No.</b>	004
<b>CASO DE ESTUDIO:</b>	Validación del código QR
<b>OBJETIVO DE LA PRUEBA:</b>	Comprobar que el código QR leído se encuentre habilitado para el horario de peticiones.
<p><b>CONDICIONES:</b> El código leído por el dispositivo debe coincidir en la tabla códigos, y debe estar activo, seguido de esto se comprueba que el aula asignada al dispositivo sea válida y con estado inactivo(ocupada), para finalizar se comprueba que el código y el aula se encuentren en la tabla peticiones y que el estado de peticiones sea “E” (que está en uso).</p> <p><b>PASOS A SEGUIR:</b></p> <ol style="list-style-type: none"> <li>1. El usuario escanea el código QR.</li> <li>2. Ingresa al aula.</li> </ol>	
<b>RESULTADOS DE LA PRUEBA</b>	
<b>RESULTADOS ESPERADOS</b>	<b>EVALUACIÓN</b>
El usuario ingresa de manera correcta, se emite un mensaje por el puerto COM indicando que el acceso es exitoso.	Exitoso <input checked="" type="checkbox"/> Fallido <input type="checkbox"/>

*Tabla 50. Prueba 4 de funcionamiento para validación de código QR*

<b>CASO DE PRUEBA No.</b>	005
<b>CASO DE ESTUDIO:</b>	Validación del usuario
<b>OBJETIVO DE LA PRUEBA:</b>	Conocer al usuario mediante el código Qr leído.
<p><b>CONDICIONES:</b> El código leído por el dispositivo debe coincidir en la tabla códigos, y debe estar activo, el id del código debe estar contenido en la tabla sesiones, y la sesión debe estar relacionada en detalle_sesion, con un id de detalles_dh_dm, a través de ese id, se verifica el id de asignación_dm, para luego consultar en la tabla asignación_dm por el id_docente, con dicho id se consulta en la tabla docentes, que finalmente tendrá el id_persona correspondiente, el que permitirá saber que persona está accediendo al aula.</p> <p><b>PASOS A SEGUIR:</b></p> <ol style="list-style-type: none"> <li>1. El usuario escanea el código QR.</li> </ol>	

2. Ingresa al aula.	
<b>RESULTADOS DE LA PRUEBA</b>	
<b>RESULTADOS ESPERADOS</b>	<b>EVALUACIÓN</b>
El usuario ingresa de manera correcta, se emite un mensaje por el puerto COM indicando que el acceso es exitoso.	Exitoso <input checked="" type="checkbox"/> Fallido <input type="checkbox"/>

Tabla 51. Prueba 5 de funcionamiento para validación de código QR

<b>CASO DE PRUEBA No.</b>	006
<b>CASO DE ESTUDIO:</b>	Llenar tabla lecturas
<b>OBJETIVO DE LA PRUEBA:</b>	Llenar la tabla lecturas, una vez validado el código qr escaneado.
<b>CONDICIONES:</b> La tabla lecturas es el resultado de todas las validaciones respecto al código qr: Que se encuentre en la base de datos, que pertenezca al aula y horario de clase al que se desea acceder, si no, que esté asignado a las peticiones.	
<b>PASOS A SEGUIR:</b>	
<ol style="list-style-type: none"> <li>1. El usuario escanea el código QR.</li> <li>2. Ingresa al aula.</li> </ol>	
<b>RESULTADOS DE LA PRUEBA</b>	
<b>RESULTADOS ESPERADOS</b>	<b>EVALUACIÓN</b>
El usuario ingresa de manera correcta, se emite un mensaje por el puerto COM indicando que el acceso es exitoso.	Exitoso <input checked="" type="checkbox"/> Fallido <input type="checkbox"/>

Tabla 52. Prueba 6 de funcionamiento llenar tabla lectura

## CONCLUSIONES

- La conexión de módulos externos a la placa de arduino, como en el caso del shield para ethernet, consume cierta cantidad de memoria SRAM extra al momento de incluir sus librerías, un caso relevante es el uso del módulo ENC28J60 con arduino uno ya que elevaba dicha memoria a un tope de 86% lo cual producía desbordamiento de memoria y por lo consiguiente el colapso del microcontrolador, a diferencia del shield de ethernet W5100 el consumo se redujo al 46%, permitiendo que el código fluya en su totalidad.
- La velocidad de descifrado que maneja el Lector Barcode 2D DP8405 es de 50cm/s lo cual hace posible que la interacción entre el dispositivo y el usuario sea de manera rápida, permitiendo el acceso al aula en un lapso de 3 a 7 segundos.
- El modo de accionamiento del dispositivo de seguridad respecto a la cerradura eléctrica es en base al relevador, lo cual permite el accionar de cualquier tipo de cerradura eléctrica que trabaje a 12V, como las cerraduras electromagnéticas, electromecánicas o solenoides.
- El uso de herramientas para el diseñado en 3D, hizo posible que la construcción del contenedor del sistema sea la adecuada, teniendo en cuenta las medidas y dimensiones de cada uno de los componentes intervinientes.
- Los códigos QR, pueden servir para la implementación de un sistema de seguridad, siempre y cuándo se agreguen a dicho código un conjunto de herramientas para mejorar su integridad, como la encriptación o el uso de OTP (One Time Password) también llamado código de un solo uso.

## RECOMENDACIONES

- El diseño eléctrico del dispositivo puede mejorar, considerando conocimientos de electrónica avanzada para el diseño y la fabricación de placas PCB, lo cual permite mayor conservación de los componentes y optimización de recursos en un ambiente de producción en masa.
- Se recomienda el uso de un regulador de voltaje para la placa controladora arduino uno, en caso de que existan variaciones o apagones eléctricos.
- La estructura de la caja contenedora puede mejorar notablemente, para incrementar la durabilidad del dispositivo y asegurar su integridad ante situaciones o ambientes extremos.
- En caso de que el dispositivo sea producido en masa, se debe considerar como parte importante, su programación, siendo el caso de que internamente se define el uso del dispositivo para una determina aula.

## BIBLIOGRAFÍA

- [1] Arqué, «Arqe,» 10 agosto 2018. [En línea]. Available: <https://arqe.com.ec/tienda/2018/08/10/todo-sobre-puertas-para-instituciones-educativas/>.
- [2] P. D. L. C. L. STEFANIA, «Software de reportes de acceso a zonas seguras para la Facultad de,» La Libertad, 2019.
- [3] UPSE, «UPSE,» 26 Junio 2018. [En línea]. Available: [https://www.upse.edu.ec/index.php?option=com\\_content&view=article&id=10&Itemid=166](https://www.upse.edu.ec/index.php?option=com_content&view=article&id=10&Itemid=166).
- [4] UPSE, «upse.edu.ec,» [En línea]. Available: <https://www.upse.edu.ec/>. [Último acceso: 05 12 2020].
- [5] secmotic, «secmotic,» 21 Septiembre 2016. [En línea]. Available: <https://secmotic.com/problemas-cotidianos-lockart/#gref>.
- [6] A. TIC, «Altiria,» [En línea]. Available: <https://www.altiria.com/sistema-de-control-acceso-con-codigos-qr-aplicacion-movil/>.
- [7] L. S. Panchana De La Cruz, «Software de reportes de acceso a zonas seguras para la Facultad de Sistema y Telecomunicaciones de la Universidad Estatal Península de Santa Elena "UPSE",» La Libertad, 2019.
- [8] Facsistel, «Facsistel,» 20 Marzo 2021. [En línea]. Available: [http://facsistel.upse.edu.ec/index.php?option=com\\_content&view=article&id=58&Itemid=463](http://facsistel.upse.edu.ec/index.php?option=com_content&view=article&id=58&Itemid=463).
- [9] D. A. Salazar Augusto y K. L. Cobeña Mite, «Diseño e implementación de un sistema de seguridad a través de cámaras, sensores y alarma.,» Guayaquil, 2015.
- [10] A. U. Cevallos Trelles y E. O. Chillpan Cachago, «Construcción de un control electrónico de una cerradura electrica de 4 digitos con alarma de seguridad par aun departamento.,» Quito, 2008.
- [11] S. N. d. P. y. Desarrollo, «Plan Nacional de Desarrollo 2017- 2021,» Senplades, Quito, 2017.
- [12] J. M. Pearce, «Economic savings for scientific free and open source technology: A review,» *HardwareX*, vol. 8, n° ISSN 2468-0672, 2020.
- [13] Y. Kao, G. Luo, H. Lin, Y. Huang y S. Yuan, «Control de acceso fisico basado en código QR,» de *Conferencia internacional de 2011 sobre computación distribuida*



*habilitada para el ciberespacio y descubrimiento de conocimientos*, Beijing, China, 2011.

- [14] Arduino, «Arduino,» [En línea]. Available: <https://store.arduino.cc/usa/arduino-uno-rev3>.
- [15] Arduino.cc, «Arduino.cc,» 17 06 2020. [En línea]. Available: [https://content.arduino.cc/assets/Pinout-UNOrev3\\_latest.pdf](https://content.arduino.cc/assets/Pinout-UNOrev3_latest.pdf).
- [16] Arduino, «Arduino,» [En línea]. Available: <https://www.arduino.cc/en/software>.
- [17] K. Corporation, «Keyence,» [En línea]. Available: <http://spanish.dyscan.com/sale-10430092-embedded-kiosk-vending-machine-desktop-barcode-reader-2d-barcode-scanner-dp8405.html>.
- [18] J. E. S. Cadena, «CONTROL DE APERTURA Y CIERRE DE LAS PUERTAS DE GARAGE A TRAVÉS DEL INTERNET,» Quito, 2020.
- [19] N. Mechatronics, «Naylampmechatronics,» [En línea]. Available: <https://naylampmechatronics.com/drivers/297-modulo-relay-1-canal-5vdc.html>.
- [20] Wiznet, «Wiznet,» 2008. [En línea]. Available: [https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100\\_Datasheet\\_v1\\_1\\_6.pdf](https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100_Datasheet_v1_1_6.pdf).
- [21] B. S. N. Segovia, «SISTEMA DE TELECONTROL Y ORIENTACION REMOTA DE ANTENAS,» Ambato.
- [22] D. S.L., «Manual de programacion lenguaje c++,» MAD, Valencia, 2005.
- [23] T. P. Group, «php,» 2021. [En línea]. Available: [php.net](http://php.net).
- [24] Autodesk, «Fusion 360,» [En línea]. Available: [https://m.media-amazon.com/images/I/71V3vRzFIYL.\\_AC\\_SY450\\_.jpg](https://m.media-amazon.com/images/I/71V3vRzFIYL._AC_SY450_.jpg).
- [25] U. d. I. Andes, «Profesores Virtual Uniandes,» [En línea]. Available: <https://profesores.virtual.uniandes.edu.co/~isis3710/dokuwiki/lib/exe/fetch.php?media=temas:http-guia.pdf>.
- [26] I. E. L. Pérez, «Protocolo SPI,» Ingeniería en Microcontroladores, México.
- [27] A. M. Fernández-Caparrós, «EL BUS I2C,» Córdoba, España, 2004.
- [28] B. S. N. Segovia, «Sistema de telecontrol y orientación remota de antenas direccionales punto a punto,» Ambato - Ecuador, 2019.

- [29] R. H. Sampieri, C. Fernández Collado y L. Pilar Baptista, Metodología de la investigación, Mexico D.F: McGRAW-HILL/Interamericana editores S.A. DE C.V, 2014.
- [30] S. P. Roger, Ingeniería del Software - Un enfoque práctico, Mexico D.F.: Mc Graw Hill - Interamericana Editores, S.A., 2010.
- [31] Gradle, «gradle.org,» [En línea]. Available: <https://gradle.org/>. [Último acceso: 12 13 2020].

# ANEXOS



## Anexo 1. Entrevista al técnico docente

Fecha: 10/12/2020



Recolección de información en busca de problemáticas que posee el control y seguridad en los cursos del bloque 1 de la Facultad de Sistemas y Telecomunicaciones.

**1. ¿Cuál es su nombre y a que se ha dedicado estos últimos 5 años?**

Enrique, soy técnico docente de la Facultad de Sistemas y Telecomunicaciones.

**2. ¿Cómo ha sido su experiencia cumpliendo el cargo de técnico docente?**

Bueno, mi experiencia es satisfactoria por el trabajo desempeñado hasta ahora.

**3. ¿Qué opina al respecto sobre la falta de control de ingreso a las aulas?**

En ocasiones si hemos tenido inconvenientes por aulas por estar ocupadas por las clases, pero en todo caso se solucionaría con la creación de más bloques de aulas.

**4. ¿Cree usted que los registros que realiza en papel deberían ser automatizados para llevar un mejor control? Sí, no, ¿por qué?**

Pienso que en la actualidad debe ser automatizado todo trabajo para poder llevar con más rapidez el mismo.

**5. ¿Cómo ve la administración que se realiza para poder verificar la disponibilidad de los cursos, y que cree usted que se debería hacer para la mejora de gestión de cursos?**

Bueno, pienso que debe realizarse mediante un sistema para que sea más ágil la disponibilidad de aulas.

**6. ¿Qué tal ha sido su experiencia y cómo ha tratado de conllevar el hecho de tener que buscar docentes o presidentes de cursos por toda la facultad para poder hacer llegar un comunicado?**

La ventaja es que he contado con aulas cercanas a mi oficina y no he tenido molestias al trasladarme.

**7. ¿Cuál es la visión que tiene para el siguiente periodo lectivo?**

Tratar de llevar el trabajo de la mejor manera.

**8. ¿Qué tan seguro cree usted que son las aulas de clases con los activos físicos que contienen cada uno?**

Creo que a las aulas si les falta mayor seguridad.

**9. ¿Cómo controlan la entrada y salida de docentes de los cursos?**

De acuerdo a sus horarios de clases.

**10. ¿En la actualidad cómo realiza la gestión de asignación de cursos y qué hace al momento de haber una disputa en el que no se sabe a quién darle la autorización del uso del aula?**

Se trata de alguna manera de solucionar rápidamente, pero por lo general tratamos de que no se crucen los horarios que de alguna u otra manera si se ha dado.

**Anexo 2. Registro de la técnica de observación aplicada en el Bloque 1 de la  
Facultad de Sistemas y Telecomunicaciones**

<b>Registro descriptivo de la información</b>	
<b>Fecha:</b> 06 de marzo del 2020	
<b>Lugar:</b> Universidad Estatal Península de Santa Elena	
<b># Personas:</b> 3	
<b>Proceso:</b> Control y seguridad en los cursos	
<b>Duración:</b> 2 horas	
<b>Hechos observados</b>	
<ul style="list-style-type: none"><li>• Cerraduras obsoletas, de fácil manipulación y acceso.</li><li>• Varios docentes no respetan el curso que se les asigna, y ocupan un aula sin solicitar al técnico docente.</li><li>• Existen choques de docentes por ocupar la misma aula.</li><li>• Se debe buscar al técnico docente, cuando un docente necesita acceder a un curso en común puesto que es el encargado de las llaves.</li><li>• La mayor parte del tiempo los cursos permanecen abiertos, puesto que los estudiantes o docentes olvidan cerrar las puertas.</li><li>• Debido a que los cursos permanecen siempre abiertos, no existe responsable alguno en caso de pérdida de bienes materiales.</li></ul>	
<b>Resumen:</b>	Se puede verificar que existe un sistema de seguridad rudimentario, y que las instalaciones de cerraduras no son las más seguras, además de que no se cuenta con un control de quienes ingresan o salen de las aulas.
<b>Responsable:</b>	Gary González Castillo

### Anexo 3: Código del proyecto (microcontrolador)

```
#include <Ethernet.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//#include <SoftwareSerial.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
static byte mymac[] = {0x74, 0x69, 0x69, 0x2D, 0x31, 0x31};
IPAddress server(52,233,72,68);
EthernetClient client;
unsigned long interval = 3000;
unsigned long interval2 = 4000;
unsigned long previousMillis1;
unsigned long previousMillis2;
String codigoleido;      //Almacena el valor de qr leido
bool stringcomplete = false; //Verifica que el codigo se ha leido
String estado;          //Almacena la respuesta de las peticiones
bool fin=false;        //Verificar que la respuesta se ha leido completamente
void setup() {
  Serial.begin (9600);
  lcd.init();          // Inicializa la pantalla lcd
  lcd.backlight();     // Pantalla siempre con luz
  delay(500);
  Serial.println(F("Adquiriendo dirección IP"));
  Ethernet.begin(mymac); // Conexion a la red por DHCP
  Serial.println(Ethernet.localIP()); //Muestra la direccion IP obtenida
  codigoleido.reserve(32); //Reserva de buffer para el string
  estado.reserve(1);      //Reserva de buffer para el string
  pinMode(5,OUTPUT);     //Declaracion de pin 5 como salida para el relé
  digitalWrite(5,HIGH);  //Se define el pin en Alto para que no se accione
```

```

    unsigned long previousMillis = millis(); //Medir el tiempo
}

void loop() {
    unsigned long currentMillis = millis();
// PROCESO EJECUCION SOLO SI SE HA LEIDO CODIGO QR
if((unsigned long)(currentMillis - previousMillis) >= interval){
    bool tipo=0;
    bool accion;
    if(stringcomplete){ //INTERRUPCION
        Serial.println("CodigoQR LEIDO");
        codigoleido.remove(32,64);
        delay(500);
        Serial.println(codigoleido);
        if (client.connect(server, 80)) { //CONEXION CON EL SERVIDOR
            Serial.println("->Conectado al servidor");
            String path="prueba.php";
            estado_pantalla(2);
            peticion_servidor(path,codigoleido);
            delay(1500);
            accion=repuesta_servidor();
            accion_puerta(accion,0);
        }else {
            Serial.println("Conecion fallida");
            client.stop();
            client.flush(); }
        }
        codigoleido = "";
        stringcomplete = false;
        previousMillis1 = millis();

```



```

}
// PROCESO CONSTANTE
if((unsigned long)(currentMillis - previousMillis2) >= interval2){
bool tipo=1;
bool accion;
if (client.connect(server, 80)) { //CONEXION CON EL SERVIDOR
Serial.println("->Conectado al servidor");
String path="puerta.php";
peticion_servidor(path,codigoleido);
delay(1500);
accion=repuesta_servidor();
accion_puerta(accion,tipo);
codigoleido = "";
stringcomplete = false;
} else {
Serial.println("Conecion fallida");
client.stop();
client.flush();
}
previousMillis2 = millis();
}
estado_pantalla(1);
}
void serialEvent(){
while (Serial.available()>0) {
char c = (char)Serial.read();
codigoleido += c;
if (c == '\n') {
stringcomplete = true;

```

```

}
}
}
//////////CONSULTA AL SERVIDOR//////////
void peticion_servidor(String path,String codigoleido){
Serial.println("GET /arduino/"+path+"?valor="+codigoleido+"&aula=B1%20-
%20203");
client.print("GET /arduino/"+path+"?valor="+codigoleido+"&aula=B1%20-%20203");
client.println(" HTTP/1.1");
client.println("Host: 52.233.72.68");
client.println("Connection: close" );
client.println();
Serial.println(F("->Peticion enviada"));
}
bool repuesta_servidor(){
String resp="";
bool puerta=0;
while (client.available() {
char c = client.read();
resp += c;
fin = true;
}
if(fin){
int longitud = resp.length();
int posicion = resp.indexOf("valor=");
estado = "";
for(int i = posicion + 6; i< longitud; i++){
if(resp[i] == ';')
i= longitud;
else

```

```
estado += resp[i];
}
fin=false;
Serial.println("----- estado:" +estado+" \n");
if(estado == "A"){
puerta=1;
estado="";
}
if(!client.connected()){
Serial.println("Desconectado");
client.stop();
client.flush();
}
return puerta;
}
//ACCIONAMIENTO DE LA PUERTA//
void accion_puerta(bool accion,bool tipo){
if(accion==1 and tipo==0){
estado_pantalla(3);
Serial.println("Abriendo puerta");
digitalWrite(5,LOW);
delay(3000);
digitalWrite(5,HIGH);
}else if(accion==0 and tipo==0){
estado_pantalla(4);
Serial.println("Acceso incorrecto");
delay(2000);
}
}
```

```

if(accion==1 and tipo==1){
estado_pantalla(3);
Serial.println("Abriendo puerta");
digitalWrite(5,LOW);
delay(3000);
digitalWrite(5,HIGH);
}
}

////////////////////ESTADOS DEL LCD////////////////////

void estado_pantalla(int val){
switch(val){
case 1:
lcd.setCursor(0,0);
lcd.print("  SEGURIDAD  ");
lcd.setCursor(0,1);
lcd.print(" =====GIAN=====");
break;
case 2:
lcd.setCursor(0, 0);
lcd.print("  COMPROBANDO  ");
lcd.setCursor(0, 1);
lcd.print("   QR   ");
break;
case 3:
lcd.setCursor(0, 0);
lcd.print("  ACCESO  ");
lcd.setCursor(0, 1);
lcd.print("  CONCEDIDO  ");
break;

```

case 4:

```
lcd.setCursor(0, 0);
```

```
lcd.print(" ACCESO ");
```

```
lcd.setCursor(0, 1);
```

```
lcd.print(" INCORRECTO ");
```

```
break;
```

```
}
```

```
}
```

#### Anexo 4: Código del proyecto (Web service - Php)

- **Configuración del servidor y las credenciales de la base de datos**

```
<?php
$host = "52.233.72.68:3306";
$username = "servidoriot";
$password = "iot2021";
$db_name = "gian_4";
//Initializes MySQLi
$con = mysqli_init();
// Establish the connection
$con=mysqli_connect($host,$username,$password,$db_name);
//If connection failed, show the error
if (mysqli_connect_errno())
{ die('Failed to connect to MySQL: ' .mysqli_connect_error()); }
?>
```

- **Consulta mysql enviando los parámetros leídos en las variables codigo y aula.**

```
<?php
require("config.php");
$codigo = $_GET['valor'];
$aula = $_GET['aula'];
$id_qr="";
$id_user="";
////////////////////CONSULTA 1 Conocer el ID del codigo escaneado
$query = "SELECT id from codigos where qr='$codigo'";
$respon1 = mysqli_query($con,$query);
if(mysqli_num_rows($respon1)>0){
while($row2 = mysqli_fetch_assoc($respon1)){
$id_qr = $row2["id"];  } }
}
```

////////////////////CONSULTA 2 Conocer el estado del codigo qr escaneado

```
$consulta2_estadoqr = "SELECT DISTINCT c.estado
from codigos c, sesiones s, detalle_sesion ds, detalles_dh_dm dhm,aulas a, peticiones p
WHERE c.qr='$codigo' and c.estado='A' and a.codigo ='$aula' and s.codigo_id=c.id
and ds.sesion_id=s.id and dhm.id = ds.detalle_dh_dm_id and a.id=dhm.aula_id and a.es
tado='I'or (p.codigo_id=c.id and p.estado='E')";

$respcon2 = mysqli_query( $con, $consulta2_estadoqr );

if (mysqli_num_rows ($respcon2) > 0) {
while($row = mysqli_fetch_assoc($respcon2)) {
echo "valor= “. $row["estado"]. ";";}
} else {
echo "0 resultados;";
}
}
```

////////////////////CONSULTA 3 Conocer el id del usuario a traves del codigo escaneado.

```
$consulta3_id_user = "SELECT DISTINCT u.id
from usuarios u, personas p,docentes d, asignacion_dm adm, detalles_dh_dm dhm, codi
gos c,sesiones s, detalle_sesion ds, aulas a, peticiones pe
WHERE u.persona_id = p.id and p.id = d.persona_id and d.id = adm.docente_id and ad
m.id = dhm.asignacion_dm_id and c.id = '$id_qr' and c.estado='A' and a.codigo='$aula'
and s.codigo_id = c.id and ds.sesion_id=s.id and dhm.id = ds.detalle_dh_dm_id and a.id
=dhm.aula_id and a.estado='I' or (pe.codigo_id=c.id and pe.estado='E' and pe.usuario_i
d =u.id)";

$respcon3 = mysqli_query($con,$consulta3_id_user);

if(mysqli_num_rows($respcon3)>0){
while($row3 = mysqli_fetch_assoc($respcon3)){
$id_user = $row3["id"]; } }
}
```

////////////////////CONSULTA 4 Llenar la tabla lecturas.

```
$insertlectura = "INSERT INTO lecturas (codigo_id,usuario_id) VALUES ('$id_qr','$id
_user)";

$llenartabla =mysqli_query ($con,$insertlectura);

mysqli_close($con);

?>
```

- **Consulta mysql cambio de estado en la tabla aulas para el accionamiento de la puerta.**

```
<?php
require("config.php");

$codigo = $_GET['valor'];
$aula = $_GET['aula'];

////////////////////CONSULTA POR EL ESTADO DE LA PUERTA

    $consultapuerta = "SELECT puerta from aulas where codigo='$aula' and
    aulas.estado = 'A' and aulas.puerta='A'";

    $respconsulta = mysqli_query($con,$consultapuerta);

    if(mysqli_num_rows($respconsulta)>0){
        while($row = mysqli_fetch_assoc($respconsulta)){
            echo "valor=A;";
        }
    }else{
        echo "0 resultados;";
    }
    mysqli_close($con);
?>
```



## Anexo 5: Implementación



*Figura 57. Ensamblaje de componentes electrónicos y estructura en 3D.*



*Figura 58. Instalación del dispositivo de seguridad a un costado de la puerta de acceso.*



*Figura 59. Conexión e instalación de cerradura eléctrica*



*Figura 60. Dispositivo final en funcionamiento*