



AUTOMATIZACIÓN DE UN CICLOPARQUEADERO "LOCKBIKERACK"

JUAN PABLO CRUZ CHAPARRO

AUTOMATIZACIÓN DE UN CICLOPARQUEADERO “LOCKBIKERACK”

JUAN PABLO CRUZ CHAPARRO



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

UNIVERSIDAD SANTO TOMÁS

INGENIERÍA ELECTRÓNICA

Bogotá D.C.

2024

AUTOMATIZACIÓN DE UN CICLOPARQUEADERO “LOCKBIKERACK”

JUAN PABLO CRUZ CHAPARRO



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

Proyecto de Grado presentado como requisito para optar al título de
INGENIERO ELECTRÓNICO

Director

M.Sc. Ing. DARIO ALEJANDRO SEGURA TORRES

UNIVERSIDAD SANTO TOMÁS

INGENIERÍA ELECTRÓNICA

Bogotá D.C.

2024

DECLARACIÓN DE AUTORÍA

Autoridades de la universidad

RECTOR GENERAL

FRAY ÁLVARO JOSÉ ARANGO RESTREPO, O.P.

VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL

FRAY HERNÁN YESID RIVERA ROBERTO, O.P.

VICERRECTOR ACADÉMICO GENERAL

FRAY MAURICIO ANTONIO CORTES GALLEGO, O.P.

SECRETARIA GENERAL

DOCTORA LUCERO GALVIS CANO

DECANO DIVISIÓN DE INGENIERÍAS

FRAY JAVIER ANTONIO HINCAPIÉ ARDILA, O.P.

SECRETARIA DE DIVISIÓN

DOCTORA LUZ PATRICIA ROCHA CAICEDO

DECANO FACULTAD DE INGENIERÍA ELECTRÓNICA

PH. D. CARLOS ANDRÉS TORRES PINZÓN

NOTA DE ACEPTACIÓN

Firma del director

Firma del jurado

Firma del jurado

Bogotá D.C. noviembre de 2024

ADVERTENCIA

La Universidad Santo Tomás no se hace responsable de las opiniones y conceptos expresados en el trabajo de grado, solo velará por qué no se publique nada contrario al dogma ni a la moral católica y porque el trabajo no tenga ataques personales y únicamente se vea el anhelo de buscar la verdad científica.

Capítulo III –Art. 46 del Reglamento de la Universidad Santo Tomás.

RESUMEN

En el presente proyecto se desarrolla el diseño e implementación de un sistema de candados por control electrónico para un puesto de anclaje de bicicletas, con la finalidad de brindar un servicio automatizado que represente eficiencia y confianza tanto para el usuario, como para el propietario del cicloparqueadero. Mediante la automatización y la programación, como áreas extensas de la Ingeniería Electrónica, se realiza el aplicativo que determina la disponibilidad de cupos en el cicloparqueadero de la Universidad Santo Tomás; los puestos individuales están equipados con un mecanismo de pestillos, cuya activación es determinada por la lectura de tarjetas incorporadas con un chip NFC, para validar la existencia del usuario en la base de datos del sistema. A través del uso de un mecanismo de engranajes de piñón a cremallera se transmite el movimiento del motor eléctrico hacia el pestillo; se realizó un programa de Arduino para así tener el control de acceso y activación del sistema, de este modo se garantiza la entrada únicamente a los usuarios registrados. Para llevar a cabo lo anterior, se empleó una metodología por etapas con un enfoque cualitativo que permitió dar cumplimiento a los objetivos establecidos y así obtener un producto que cumple con las características planteadas.

ABSTRACT

In this project, the design and implementation of an electronically controlled padlock system for a bicycle rack is developed, with the aim of providing an automated service that represents efficiency and confidence for both the user and the owner of the bicycle parking lot. Through automation and programming, as extensive areas of Electronic Engineering, the application that determines the availability of quotas in the bicycle parking lot of the Santo Tomás University is carried out; the individual posts are equipped with a latch mechanism, the activation of which is determined by the reading of built-in cards with an NFC chip, to validate the existence of the user in the system's database. By using a rack and pinion gear mechanism, the movement of the electric motor is transmitted to the lock latch; an Arduino program was made to have access control and activation of the system, thus guaranteeing entry only to registered users. To carry out the above, a methodology was used in stages with a qualitative approach that allowed completing the established objectives and thus obtaining a product that meets the proposed characteristics.

DEDICATORIA

En dedicación a los jóvenes que ven en el aprendizaje una oportunidad para crecer y conseguir sus propósitos profesionales; a quienes dedican su vida a formarse para construir una sociedad más capacitada y a todos aquellos que optan por un medio de transporte menos contaminante y requieran de espacios seguros para transitar.

AGRADECIMIENTOS

Agradezco a mis padres, Andrés Cruz y Andrea Chaparro, por su inmenso apoyo a lo largo de mi carrera y de cada uno de mis objetivos académicos; además de brindarme su confianza incondicional en las decisiones tomadas para mi vida.

Agradezco a mi hermana Laura por su cariño en los momentos de dificultad, quien ha significado para mí estabilidad y motivación para continuar con mis proyectos personales.

Agradezco también a mis compañeros de profesión y amigos, quienes a través de experiencias vividas me han enseñado y han reforzado las virtudes más humanas que tengo.

Gracias por todo,
Juan Pablo Cruz Chaparro.

Agradezco ampliamente a la Universidad Santo Tomás de Bogotá, a sus trabajadores por recibirme de manera cordial y brindar su atención en el diario vivir; a sus docentes por compartir de forma eficaz los conocimientos necesarios para hacer de mí una persona íntegra y un profesional apto para forjar el futuro de nuestro país.

ÍNDICE

Capítulo 1	Introducción.....	1
Capítulo 2	Planteamiento del problema.....	2
Capítulo 3	Justificación	3
Capítulo 4	Impacto social	4
Capítulo 5	Objetivos	5
5.1	General	5
5.2	Específicos	5
Capítulo 6	Marco teórico	6
6.1.	Automatización.....	6
6.1.1.	Sistemas digitales.....	6
6.2.	Programación.....	7
6.2.1.	Algoritmos	8
6.2.2.	Arduino IDE.....	8
6.2.2.1.	Wi-Fi.....	8
6.2.2.1.1.	ESP8266 – ESP01	8
6.2.2.2.	RFID.....	9
6.2.2.2.1.	RFID – RC522	9
6.2.3.	Microsoft Visual Studio.....	9
6.2.3.1.	Aplicativos Web.....	9
6.2.3.1.1.	Servidores.....	9
6.2.3.1.2.	HTML 5	10
6.3.	Bases de datos	10
6.3.1.	Microsoft SQL Server.....	10
6.4.	PCB	10
6.4.1.	Proteus Design Suit.....	11
6.5.	Diseño asistido por ordenador (CAD)	11
6.5.1.	Autodesk Inventor	11
6.5.2.	Autodesk Fusion 360.....	12
6.5.3.	Autodesk Tinkercad.....	12
Capítulo 7	Metodología.....	13
7.1.	Revisión del estado del arte.....	13
7.2.	Diseño de circuitos electrónicos y base de datos.....	14
7.3.	Diseño de componentes y estructura física.....	14

7.4.	Fabricación de piezas	14
7.5.	Montaje de la versión del prototipo	14
7.6.	Diseño de la página web.....	14
7.7.	Pruebas de funcionamiento y validación del prototipo	15
7.8.	Retroalimentación de optimización de versiones	15
7.9.	Construcción del documento.....	15
Capítulo 8	Diseño	16
8.1.	Diseño e implementación de sistemas electrónicos	17
8.1.1.	Diseño circuito principal.....	17
8.1.2.	Simulación del circuito principal.....	19
8.1.3.	Diseño de la PCB.....	23
8.2.	Submódulo conector y adaptador de corriente continua.....	28
8.3.	Submódulo controlador motores (Driver)	29
8.3.1.	Simulación submódulo controlador de motores (Driver)	31
8.4.	Submódulo Wi-Fi ESP8266	32
8.4.1.	Simulación submódulo Wi-Fi ESP8266	33
8.5.	Submódulo lector de tarjetas RFID RC522 NFC	36
8.5.1.	Simulación submódulo lector tarjetas RFID RC 522 NFC.....	37
8.6.	Submódulo de indicadores.....	41
8.6.1.	Simulación submódulo de indicadores	42
8.7.	Submódulo de sensores.....	43
8.7.1.	Simulación submódulo de sensores	45
8.8.	Integración de componentes del sistema LockBikeRack.....	48
8.8.1.	Circuito principal	48
8.9.	Diseño de piezas del prototipo	52
8.9.1.	Diseño estructura general	53
8.9.2.	Diseño divisiones	54
8.9.3.	Diseño de la estructura soporte del pestillo	56
8.10.	Diseño de aplicación y servicios web.....	57
8.10.1.	Servicios	57
8.10.2.	Aplicación.....	60
Capítulo 9	Resultados.....	63
9.1.	Circuitos electrónicos	64
9.2.	Piezas estructurales	65
9.2.1.	Soportes del mecanismo de los pestillos.....	65

9.2.2.	Cubierta estructural.....	65
9.3.	Base de datos	66
9.4.	Página web.....	69
9.5.	Montaje final del prototipo	71
Capítulo 10	Conclusiones.....	73
Capítulo 11	Trabajos Futuros	74

LISTA DE FIGURAS

Figura 1. Diagrama metodológico Fuente: Autor	13
Figura 2. Diagrama de bloques del sistema electrónico - LockBikeRack Fuente: Autor	17
Figura 3. Diseño final PCB, Proteus 8 Profesional - LockBikeRack Fuente: Autor	18
Figura 4. Diagrama 2D del sistema electrónico - LockBikeRack Fuente: Autor	19
Figura 5. Diagrama de flujo del algoritmo de prueba del circuito principal - LockBikeRack Fuente: Autor	21
Figura 6. Simulación del algoritmo de prueba del circuito principal – LockBikeRack Fuente: Autor	22
Figura 7. Esquema del sistema electrónico - LockBikeRack Fuente: Autor	23
Figura 8. Holgura de las pistas, Proteus 8 Professional - LockBikeRack. Fuente: Autor	24
Figura 9. Diseño capa superior PCB, Proteus 8 Profesional – LockBikeRack Fuente: Autor	25
Figura 10. Diseño conexiones capa inferior PCB, impresión – LockBikeRack Fuente: Autor	25
Figura 11. Diseño capa inferior PCB, Proteus 8 Profesional - LockBikeRack Fuente: Autor	26
Figura 12. Diseño capa superior esquemático PCB, Proteus 8 Profesional – LockBikeRack Fuente: Autor	26
Figura 13. Diseño para impresión física capa inferior PCB, impresión – LockBikeRack Fuente: Autor	27
Figura 14. Diseño impresión física capa superior PCB, impresión – LockBikeRack Fuente: Autor	27
Figura 15. Módulo jack de corriente continua – LockBikeRack Fuente: Autor	28
Figura 16. Esquemático del submódulo conector jack DC – LockBikeRack Fuente: Autor	28
Figura 17. Esquemático submódulo controlador motores – LockBikeRack Fuente: Autor	30
Figura 18. Modelo 3D submódulo controlador motores (driver) y terminales para los motores M1 y M2 – LockBikeRack Fuente: Autor	30
Figura 19. Simulación 2D submódulo controlador motores (driver) – LockBikeRack Fuente: Autor	31
Figura 20. Submódulo controlador de motores (driver) – LockBikeRack Fuente: Autor	32
Figura 21. Montaje del motor con mecanismo de piñón a cremallera – LockBikeRack Fuente: Autor	32
Figura 22. Esquemático submódulo Wi-Fi ESP8266 – LockBikeRack Fuente: Autor	33
Figura 23. Simulación submódulo Wi-Fi ESP8266 con regulador de voltaje – LockBikeRack Fuente: Autor	34
Figura 24. Adaptador USB CH340 – LockBikeRack Fuente: Autor	34
Figura 25. Módulo Wi-Fi ESP8266 – LockBikeRack Fuente: Autor	34
Figura 26. Diagrama de flujo del algoritmo de prueba del módulo ESP8266 - LockBikeRack Fuente: Autor	35
Figura 27. Salida monitor serie Arduino, submódulo Wi-Fi ESP8266 – LockBikeRack Fuente: Autor	36
Figura 28. Esquemático submódulo lector de tarjetas RFID RC522 – LockBikeRack Fuente: Autor	37
Figura 29. Simulación 2D lector de tarjetas RFID RC522 – LockBikeRack Fuente: Autor	38
Figura 30. Diagrama de flujo del algoritmo de prueba del submódulo RFID-RC522 – LockBikeRack Fuente: Autor	39
Figura 31. Submódulo lector de tarjetas RFID RC522 – LockBikeRack Fuente: Autor	40
Figura 32. Tarjeta para lector RFID RC522 – LockBikeRack Fuente: Autor	40
Figura 33. Salida monitor serie Arduino, lector de tarjetas RFID RC522 – LockBikeRack Fuente: Autor	41
Figura 34. Esquemático submódulo indicadores – LockBikeRack Fuente: Autor	42
Figura 35. Simulación 2D submódulo indicadores – LockBikeRack Fuente: Autor	42

Figura 36. Diagrama de flujo del algoritmo de prueba del submódulo de indicadores – LockBikeRack Fuente: Autor.....	43
Figura 37. Esquemático submódulo sensores – LockBikeRack Fuente: Autor	44
Figura 38. Modelo 3D del sensor de final de carrera – LockBikeRack Fuente: Autor	45
Figura 39. Simulación 2D submódulo sensores – LockBikeRack Fuente: Autor	46
Figura 40. Diagrama de flujo del algoritmo de prueba del submódulo de sensores – LockBikeRack Fuente: Autor.....	47
Figura 41. Sensor de final de carrera fijado al soporte de pestillo – LockBikeRack Fuente: Autor	48
Figura 42. Programación final, variables del sistema – LockBikeRack Fuente: Autor.....	49
Figura 43. Programación final, ajustes de variables – LockBikeRack Fuente: Autor	50
Figura 44. Programación final - LockBikeRack Fuente: Autor	51
Figura 45. Programación final, funciones – LockBikeRack Fuente: Autor	52
Figura 46. Programación final, funciones – LockBikeRack Fuente: Autor	54
Figura 47. Simulación división pisos – LockBikeRack Fuente: Autor	54
Figura 48. Simulación división pisos estructura general – LockBikeRack Fuente: Autor	55
Figura 49. Simulación división pisos circuitos electrónicos – LockBikeRack Fuente: Autor	55
Figura 50. Simulación de soporte pestillo – LockBikeRack Fuente: Autor	56
Figura 51. Esquema de servicios y aplicación web - LockBikeRack Fuente: Autor.....	57
Figura 52. Diagrama de flujo del algoritmo de registro de ingreso - LockBikeRack Fuente: Autor	58
Figura 53. Diagrama de flujo del algoritmo de registro de salida - LockBikeRack Fuente: Autor	59
Figura 54. Diagrama de flujo del algoritmo de consulta inicial - LockBikeRack Fuente: Autor	60
Figura 55. Formulario de inicio de sesión - LockBikeRack Fuente: Autor	61
Figura 56. Formulario de registro - LockBikeRack Fuente: Autor	61
Figura 57. Página principal y estado del cicloparqueadero - LockBikeRack Fuente: Autor	62
Figura 58. Diagrama de flujo del algoritmo del servidor - LockBikeRack Fuente: Autor	63
Figura 59. PCB del circuito principal superior – LockBikeRack Fuente: Autor	64
Figura 60. PCB del circuito principal inferior – LockBikeRack Fuente: Autor	64
Figura 61. Impresión 3D soporte de pestillo perspectiva – LockBikeRack Fuente: Autor.....	65
Figura 62. Impresión 3D soporte de pestillo superior – LockBikeRack Fuente: Autor	65
Figura 63. Cubierta estructural perspectiva – LockBikeRack Fuente: Autor.....	66
Figura 64. Cubierta estructural frontal – LockBikeRack Fuente: Autor.....	66
Figura 65. Diagrama de base de datos – LockBikeRack Fuente: Autor	67
Figura 66. Ficheros de base de datos – LockBikeRack Fuente: Autor.....	67
Figura 67. Tabla Bici_Usuario base de datos – LockBikeRack Fuente: Autor	68
Figura 68. Tabla Estacion_Guardado base de datos – LockBikeRack Fuente: Autor	68
Figura 69. Tabla Parqueadero base de datos – LockBikeRack Fuente: Autor	68
Figura 70. Tabla Usuarios base de datos – LockBikeRack Fuente: Autor.....	68
Figura 71. Consulta de contenido de tablas, base de datos – LockBikeRack Fuente: Autor	69
Figura 72. Ingreso, página web – LockBikeRack Fuente: Autor	70
Figura 73. Registro, página web – LockBikeRack Fuente: Autor	70
Figura 74. Estado del cicloparqueadero, página web – LockBikeRack Fuente: Autor	71
Figura 75. Montaje final – LockBikeRack Fuente: Autor	71
Figura 76. Estado disponible, página web – LockBikeRack Fuente: Autor.....	72
Figura 77. Estado ocupado, página web – LockBikeRack Fuente: Autor.....	72

LISTA DE TABLAS

Tabla 1. Posibles combinaciones lógicas controlador motores - LockBikeRack Fuente: Autor.....	29
---	----

LISTA DE ANEXOS

Anexo 1. Diagrama de sistemas electrónicos	1
Anexo 2. Diagrama de flujo del sistema.....	2
Anexo 3. Planos del circuito principal.....	3
Anexo 4. Esquema de la base de datos.....	7
Anexo 5. Diseño del logotipo.....	8
Anexo 6. Modelos 3D	9
Anexo 7. Planos de la cubierta y soporte de pestillo	20
Anexo 8. Capturas de la página web	22
Anexo 9. Códigos de programación	23

Capítulo 1

Introducción

La ciudad de Bogotá D.C., durante los últimos años ha enfrentado una problemática de seguridad con implicaciones directas sobre los usuarios de bicicleta; la delincuencia y el hurto a mano armada. Para el año 2021 la cantidad de robos de bicicleta reportados alcanzaría la cifra de 8.200 bicicletas [1], lo cual no solamente involucra los daños y perjuicios por el daño material, sino que también pondría en riesgo la vida de las víctimas. Hoy en día se han tomado medidas por parte de la secretaría distrital de movilidad con la finalidad de identificar al propietario de una bicicleta mediante el registro de sus datos [2], lo cual parece favorecer la devolución del vehículo si posterior al robo se encuentra la bicicleta; sin embargo, este tipo de soluciones no contemplan la prevención ante la sustracción, el hurto y el robo de dicho medio de transporte, por lo cual, es necesario generar distintas soluciones que permitan al usuario contar con un mayor soporte y disponer de lugares de parqueo, los cuales le brinden mayor seguridad y confianza.

La solución propuesta plantea el desarrollo y fabricación de un sistema de seguridad automatizado que, por medio de la validación de identidad, permita al usuario enganchar su bicicleta a los soportes de un cicloparqueadero con el uso de correas de sujeción. También contempla la implementación de una base de datos que respalde la información de usuario y los registros de ingreso y salida del ciclo- parqueadero; con el fin de permitir que las personas registradas en la plataforma revisen la disponibilidad de cupos dentro del parqueadero consultando la página web.

En este documento se presentará el trabajo desarrollado para llevar a cabo la solución propuesta, enfocado en el montaje del sistema LockBikeRack en un puesto de anclaje de bicicletas ubicado en el parqueadero de la sede central de la Universidad Santo Tomás, Bogotá. A modo de aplicación de áreas de la ingeniería electrónica, tales como la Automatización industrial y la Programación Web.

Capítulo 2

Planteamiento del problema

El hurto de bicicletas en Colombia corresponde al delito más reportado para el periodo 2020-2021 representando al 5,8% de propietarios de bicicleta mayores de 14 años. Luego del atraco, las modalidades de hurto de bicicleta más reportadas son por descuido y violentar la bicicleta [3].

Según el último boletín mensual de indicadores de seguridad y convivencia de Bogotá, publicado por la Oficina de Análisis de Información y Estudios Estratégicos (OAIEE), se registraron un total de 5750 casos de hurto de bicicletas entre enero y septiembre del 2023, siendo Kennedy la localidad con más casos reportados, seguida por Engativá y Suba. [4]

De acuerdo con las estadísticas recopiladas por el Sistema Estadístico de la Policía Nacional (SIEDCO), durante el mes de enero, desde el primer día hasta el último, se reportaron 551 casos de robo de bicicletas en la ciudad de Bogotá. Asimismo, según las cifras oficiales proporcionadas por las autoridades, se estima que, en promedio, ocurre un robo de bicicleta cada 42 minutos en la ciudad. [5]

“La estrategia de bici denuncias de la Secretaría de Seguridad, es una medida de criminalización secundaria; no hay registros acerca del número de investigaciones y juicios; tampoco se evidenció en los tramos priorizados presencia de la Policía ni efectividad en los consejos de autocuidado”.

Teniendo en cuenta los datos descritos anteriormente, se plantea una inquietud: ¿De qué manera contribuir a la seguridad social referente a la problemática del hurto de bicicletas en la ciudad de Bogotá?

Capítulo 3

Justificación

Bogotá ha sido clasificada como una de las 20 ciudades con mayor promoción del uso de la bicicleta en el mundo para el año 2019, la organización Copenhagenize ha realizado un estudio cada dos años desde el 2011 hasta el 2019 analizando a detalle características como la infraestructura destinada al uso activo de la bicicleta, así como también la proyección social y cultural de este medio.

Empero a las facilidades que brinda la ciudad, Bogotá enfrenta un alto índice de hurto de bicicletas; tal como registra el periódico El Espectador, en el año 2018 a mes de agosto, habrían sido hurtadas 4011 bicicletas, aumentando en un 72% la cifra de hurtos de bicicletas a comparación del año 2017; además de dar como resultado 38 fallecidos y 128 heridos a causa de tal delito.

La problemática en Bogotá, expresamente hablando del hurto de bicicletas, tiene mayores repercusiones que las mencionadas previamente, además de significar un riesgo para la seguridad y el bienestar de los biciusuarios, se convierte en un negocio cíclico; el mercado ilegal de bicicletas se lucra al aprovechar el auge de dicha actividad, generando ingresos también desde la reventa de componentes de bicicleta.

Tal como asegura la personería de Bogotá, en los últimos tres años (2016-2019), se ha presentado un aumento del 419% en el hurto de bicicletas. A raíz de esto y dados los abruptos aumentos en la inseguridad, es propuesto un mecanismo que brinde una mayor seguridad a los cicloparqueaderos de Bogotá, mediante el cual, sea validada la identidad del usuario y permita únicamente retirar una bicicleta por el propietario.

Capítulo 4

Impacto social

Con base en la proyección social de este proyecto, se pretende generar nuevas ideas que permitan inspirar a los jóvenes a mejorar su calidad de vida e implementar una comunidad del progreso por medio de proyectos de aplicación real. La automatización empleada durante el desarrollo de este proyecto no solo permite mejorar los sistemas de seguridad, sino que también, busca impulsar a la construcción de una sociedad que vele por los derechos humanos y por salvaguardar el medio ambiente. Se pretende incentivar al uso de la bicicleta además de brindar ayuda a sus usuarios con sistemas que permitan expandir los espacios seguros, y simplifiquen el proceso de estacionamiento representando una mayor confianza en el uso de este transporte para el día a día.

Con este proyecto las personas que se movilicen en bicicleta tendrán la posibilidad de obtener oportunidades para hacer de su transporte, un instante de comodidad y libertad en su rutina diaria, la cual es clave al momento de promover su tranquilidad a la hora de movilizarse en bicicleta.

Capítulo 5

Objetivos

5.1 General

Implementar un sistema para un cicloparqueadero basado en un mecanismo de candado flexible controlado electrónicamente empleando tarjetas con chip NFC para contribuir con la seguridad de los biciusuarios.

5.2 Específicos

- Implementar un candado flexible con control electrónico y validación por medio de tarjeta NFC para una barricada de un cicloparqueadero.
- Implementar un software que permita el registro de usuarios y consulta de cupos disponibles para la autenticación y uso del cicloparqueadero.
- Implementar un prototipo del sistema en un cicloparqueadero con el fin de validar su correcto funcionamiento.

Capítulo 6

Marco teórico

Para el desarrollo del proyecto es necesario conocer una serie de temáticas y bases teóricas, las cuales tienen una implicación directa en el entendimiento de este; por ello, se describen a continuación las áreas del conocimiento pertinentes, haciendo énfasis en conceptos particulares y programas propios del diseño establecido.

6.1. Automatización

La automatización es el proceso de usar componentes mecánicos, electrónicos y/o de software, que trabajen juntos para realizar tareas que normalmente requieren intervención humana. Ésta se originó en la revolución industrial, cuando se introdujeron las máquinas a vapor y otras tecnologías para mejorar la eficiencia de los procesos manufactureros. A medida que avanzó la tecnología, la automatización se extendió a otras industrias y se convirtió en parte integral de muchos procesos de producción y operaciones. [5]

Uno de los usos más frecuentes de la automatización está relacionado a los procesos de negocios, mediante el cual se utilizan sistemas digitales para automatizar tareas y procesos empresariales. La automatización de empresas se puede llevar a cabo mediante la implementación de software de gestión empresarial, sistemas de información integrados o sistemas de automatización de procesos. La automatización de empresas permite a las empresas mejorar su eficiencia y reducir sus costes.

En el contexto del sistema de candados por control electrónico, la automatización se refiere a la capacidad del sistema para activar y desactivar los pestillos de forma automática, basándose en la lectura de las tarjetas NFC y en la disponibilidad de cupos en el cicloparqueadero. Esto garantiza un servicio continuo y sin necesidad de intervención humana constante.

6.1.1. *Sistemas digitales*

Un sistema digital hace parte de un sistema de información en el que los datos se representan mediante un conjunto de valores discretos. Estos valores pueden ser números, letras o símbolos, y se codifican mediante un código binario. Los sistemas digitales se caracterizan por ser fiables, precisos y por su capacidad de almacenar y procesar grandes cantidades de datos. [6]

La lógica digital se ocupa de los circuitos digitales y de su análisis; se centra en la forma en la que los circuitos pueden ser diseñados y analizados usando modelos matemáticos. Mediante la organización de entradas en un sistema, es posible determinar el comportamiento de su salida según una serie de combinaciones en los parámetros de entrada del sistema y corresponderlos a una respuesta esperada.

- **Sensores**

Un sensor es un dispositivo que detecta una señal o una magnitud física y la convierte en una señal eléctrica. Se utilizan en una amplia variedad de aplicaciones, desde la medición de la temperatura y la humedad hasta la detección de objetos y el seguimiento de movimientos. También se utilizan en sistemas de control y automatización para detectar y medir una amplia variedad de parámetros físicos.

- **Actuadores**

Un actuador es un dispositivo que convierte energía de un tipo determinado a energía de movimiento. Los actuadores eléctricos convierten energía eléctrica en energía de movimiento; se usan en una amplia variedad de aplicaciones, desde sistemas de control de procesos industriales hasta robots y sistemas automatizados. En términos generales, son utilizados para generar fuerza o movimiento en respuesta a una señal de control.

6.2. Programación

La programación es el proceso de diseñar un conjunto de instrucciones para controlar el comportamiento de un dispositivo electrónico. Las instrucciones pueden especificar cosas como qué datos procesar, cómo procesarlos y qué acciones realizar como resultado.

La programación tiene sus raíces en la ingeniería eléctrica. Los primeros dispositivos programables fueron los telégrafos, que fueron inventados en la década de 1830, los cuales utilizaban una serie de cables para enviar señales eléctricas a través de largas distancias, y los operadores podían controlar el dispositivo mediante el uso de interruptores para conectar o desconectar los cables.

A mediados de 1800, se inventó la máquina de telar Jacquard, que utilizaba tarjetas perforadas para controlar el telar. Esta máquina fue una de las primeras en utilizar una forma de programación, y se considera un precursor de las modernas computadoras. [7]

En la programación informática se utiliza un lenguaje que representa el conjunto de símbolos y reglas a modo de estructura sintáctica. Los lenguajes de programación permiten a los desarrolladores crear instrucciones que pueden controlar el comportamiento de un dispositivo electrónico. Algunos ejemplos de lenguajes de programación comunes son C, C++, Java y Python.

La programación es el corazón del sistema, ya que es la que permite controlar todas las acciones y procesos de este. En este proyecto, se utilizó la programación de Arduino para desarrollar el software que gestiona el acceso de los usuarios y controla la activación de los candados electrónicos. La programación adecuada garantiza un funcionamiento preciso y seguro del sistema, además de permitir futuras mejoras y actualizaciones.

6.2.1. Algoritmos

Un algoritmo es un conjunto de pasos definidos que permiten resolver un problema. En la programación, los algoritmos se traducen en código que puede ser ejecutado por una computadora, representables mediante la sucesión de instrucciones organizadas de manera secuencial o paralela.

Los algoritmos se pueden clasificar de varias maneras, siendo la más común la clasificación en función del tiempo de ejecución y el espacio utilizado por el programa. En cuanto al tiempo de ejecución, los algoritmos se pueden dividir en deterministas y no deterministas. Los primeros son aquellos que siempre producen un resultado correcto dentro de un tiempo finito, independientemente del ordenador o las condiciones iniciales; y los segundos son aquellos que no garantizan obtener un resultado correcto dentro de un tiempo finito o bien dependiendo del ordenador o de las condiciones iniciales. [8]

6.2.2. Arduino IDE

Es un entorno de desarrollo integrado libre y multiplataforma para programar en lenguaje C/C++. Fue creado originalmente para el microcontrolador Arduino, pero se puede utilizar con otros microcontroladores.

El IDE de Arduino tiene como principal objetivo simplificar el proceso de codificación y programación para los usuarios que desean crear sus propias aplicaciones basadas en Arduino. Proporciona un entorno visual intuitivo que permite a los usuarios concentrarse más en su lógica de programación y menos en la sintaxis. [9]

6.2.2.1. Wi-Fi

La tecnología Wi-Fi se refiere a la conectividad inalámbrica a una red de datos. Se utiliza en dispositivos móviles, como ordenadores portátiles y teléfonos inteligentes, así como en puntos de acceso que permiten el acceso a Internet en áreas cercanas.

Un módulo Wi-Fi es un módulo de hardware que se conecta a un microcontrolador y le permite comunicarse con otros dispositivos a través de una red inalámbrica. Existen diferentes tipos de módulos Wi-Fi, cada uno diseñado para cumplir con un propósito específico. Algunos módulos son capaces de actuar como punto de acceso (AP), permitiendo que otros dispositivos se conecten a ellos y se comuniquen directamente. Otros módulos pueden funcionar como estaciones base (STA), lo que les permite conectarse directamente a un punto de acceso y enviar o recibir datos a través de él. [10]

6.2.2.1.1. ESP8266 – ESP01

El ESP8266 corresponde a un SoC (System on Chip) encargado de realizar procesos de escritura y lectura mediante ondas de radiofrecuencia. Está equipado con un microcontrolador con una arquitectura de 32 bits, permitiéndole funcionalidades de conectividad tales como punto de acceso o estación Wi-Fi.

El módulo ESP-01 permite equipar las funciones del chip ESP8266 a microcontroladores de uso práctico como los de Arduino, permitiendo su programación desde un IDE más versátil y cómodo.

6.2.2.2. RFID

RFID es el acrónimo de Radio Frequency Identification (Identificación por Radiofrecuencia). Se trata de una tecnología inalámbrica que permite la identificación y el seguimiento de objetos, personas o animales mediante el uso de etiquetas RFID. Las etiquetas contienen chips que emiten señales de radio que pueden ser leídas por un dispositivo receptor. Esta tecnología está siendo cada vez más utilizada en aplicaciones comerciales y militares, ya que ofrece un modo eficiente y preciso de rastrear objetos. [11]

Existen dos principales categorías de RFID: activo y pasivo. Los dispositivos RFID activos utilizan una fuente externa de energía, como una batería, para alimentar el chip y la etiqueta. Esto les permite emitir señales de radio más potentes y leer las etiquetas a mayores distancias. Los dispositivos RFID pasivos, por otro lado, no tienen ninguna fuente externa de energía; en su lugar, recargan su batería mediante el campo electromagnético generado por el lector. Debido a que están alimentados de forma inalámbrica, los dispositivos RFID pasivos son generalmente más pequeños y menos costosos que los activos.

6.2.2.2.1. RFID – RC522

Se trata de un módulo de comunicación de lectura y escritura sin contacto; utiliza señales de radiofrecuencia de baja potencia a 13.56 MHz, normalmente utilizado en aplicaciones de tecnología NFC (Near Field Communication) para la lectura de etiquetas (tags) como remplazo a las tecnologías de lectura de código de barras. Mediante el uso de librerías especializadas es posible implementarlo en distintas aplicaciones junto con un microcontrolador Arduino.

6.2.3. Microsoft Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) de aplicaciones para Windows, MacOS, Android y iOS que permite a los desarrolladores crear y editar código fuente, así como compilar e implementar aplicaciones. El IDE incluye herramientas para el control de versiones, depuración, análisis y pruebas de rendimiento. También brinda soporte para la creación de paquetes MSI y MSIX para la distribución de aplicaciones. [12]

6.2.3.1. Aplicativos Web

Los aplicativos webs se ejecutan en un servidor y se accede a ellos a través de un navegador web. Este tipo de aplicaciones está escrito usando lenguajes de scripting del lado del servidor, como PHP, ASP y Java. Los navegadores web descargan los archivos necesarios para ejecutar la aplicación desde el servidor y luego procesan las solicitudes del usuario localmente.

6.2.3.1.1. Servidores

Los servidores son computadoras que proporcionan servicios a otras computadoras conectadas a una red. Algunos ejemplos de servicios proporcionados por los servidores son el alojamiento de sitios web, la ejecución de base de datos y el envío de correos electrónicos. Los servidores suelen estar equipados con más recursos y capacidades que las computadoras personales, ya que deben ser capaces de manejar el tráfico elevado y las solicitudes concurrentes.

6.2.3.1.2. HTML 5

Se trata de la quinta versión del lenguaje estándar de programación vigente en estructura para el World Wide Web, el cual unifica funciones de HTML y XML. Cuenta con atributos para el desarrollo y composición de páginas web, que permiten incorporar el texto, imagen y medios dentro de un servidor determinado.

6.3. Bases de datos

Las bases de datos son un conjunto de datos organizados de manera estructurada con la finalidad de aprovechar la información de una manera activa, permiten almacenar, organizar y recuperar información. Los usuarios pueden acceder a la información en las bases de datos a través de interfaces gráficas o líneas de comandos.

Normalmente se emplea un gestor de bases de datos para controlar las tablas y registros, organizados a modo de columnas y filas, permitiendo generar una relación directa entre los datos otorgándole un valor contextual a la información.

Las bases de datos son fundamentales para almacenar y gestionar la información de los usuarios registrados en el sistema. En este proyecto, se utiliza una base de datos para almacenar la información de los usuarios autorizados a acceder al cicloparqueadero. La lectura de las tarjetas NFC se compara con los registros de la base de datos para validar la existencia del usuario y permitirle el acceso. Una base de datos bien estructurada y segura garantiza la integridad de la información y la confiabilidad del sistema.

6.3.1. Microsoft SQL Server

Es un programa de base de datos relacional que permite a los usuarios almacenar y recuperar información a través de consultas estructuradas. El programa también ofrece herramientas para la administración de bases de datos, el diseño de consultas y el análisis de datos.

6.4. PCB

Una PCB es una tarjeta de circuito impreso. Se utiliza para conectar electrónicamente los componentes de un dispositivo y permitir que la energía eléctrica fluya a través del circuito. Las PCB se pueden encontrar en una amplia variedad de dispositivos, desde ordenadores y teléfonos inteligentes hasta automóviles y electrodomésticos.

Las PCB están hechas de una capa delgada de material aislante, como el papel o la fibra de vidrio, cubierta con un patrón de conductores metálicos. El material aislante se llama sustrato y los conductores se llaman pistas. Las pistas están conectadas entre sí y a los componentes electrónicos mediante soldadura o pernos. Su principal ventaja frente a otros tipos de circuitos es su tamaño, permitiendo optimizar la disposición de los elementos electrónicos según su finalidad y requerimientos físicos. Las PCB también son más económicas que otros tipos de circuitos, lo que las hace ideales para aplicaciones en masa.

Las PCB son cruciales en la fabricación de los circuitos electrónicos que componen el sistema de candados por control electrónico. Permiten la interconexión de los componentes electrónicos de manera ordenada y compacta, lo que facilita su integración en el diseño final. En este proyecto, se diseñaron y fabricaron PCB personalizadas para alojar los componentes electrónicos del sistema, lo que optimiza el espacio y asegura un funcionamiento eficiente.

6.4.1. *Proteus Design Suit*

Proteus Design Suit es un programa informático de diseño electrónico que permite a los usuarios crear y simular circuitos integrados. El software se utiliza en entornos educativos para ayudar a los estudiantes a comprender mejor cómo funcionan los circuitos, así como para permitirles experimentar con diferentes diseños antes de construirlos físicamente. También se utiliza en el mundo empresarial para permitir a los ingenieros probar nuevos diseños antes de fabricarlos, lo que puede ayudar a reducir costes y tiempos de desarrollo.

6.5. Diseño asistido por ordenador (CAD)

El diseño asistido por ordenador (CAD) es una técnica de diseño en la que se utilizan ordenadores para ayudar a los diseñadores a crear y modificar modelos tridimensionales de objetos. CAD se puede usar para crear una variedad de objetos, desde piezas mecánicas simples hasta edificios complejos.

Este tipo de herramientas usualmente permiten al usuario diversas funciones en cuanto al diseño de estructuras y mecanismos, brindando modelos tridimensionales y pruebas de estrés mecánico para el desarrollo de aplicaciones aptas para el mundo real.

El CAD es una herramienta fundamental en el diseño y desarrollo de productos electrónicos. En este proyecto, se utilizó software de CAD para diseñar los componentes mecánicos del sistema, como los soportes de pestillos y motores eléctricos. El CAD permite modelar y simular los componentes antes de su fabricación, lo que facilita la detección de posibles fallos y la optimización del diseño final.

6.5.1. *Autodesk Inventor*

Inventor es un software de diseño asistido por ordenador de la compañía Autodesk que provee herramientas de diseño mecánico profesional y simulación 3D. El programa cuenta con técnicas de modelado y simulación 3D, capacitadas para el diseño y la optimización del desempeño

de un producto; permite crear configuraciones personalizables para procesos de automatización de tareas repetitivas. [13]

6.5.2. *Autodesk Fusion 360*

Fusion 360 es un programa de Autodesk enfocado a funciones de modelado y diseño 3D, con una tecnología basada en el almacenamiento en la nube. Este permite recrear las interacciones de un sistema mecánico, diseñar placas de circuitos impresos e incluso realizar pruebas de diseño para asegurar las capacidades ante distintas condiciones. [14]

6.5.3. *Autodesk Tinkercad*

Tinkercad es una aplicación web diseñada por Autodesk la cual permite realizar modelos de estructuras y sistemas principalmente enfocados al aprendizaje, la enseñanza y la innovación. Cuenta con una herramienta de diseño 3D, simulación de circuitos electrónicos y codificación en bloques mediante algoritmos dinámicos de diseño. [15]

Capítulo 7

Metodología

Para el proyecto se diseñó una metodología con un enfoque cualitativo debido a la naturaleza subjetiva de su aplicación. Se decide implementar un diseño por etapas como se describe en el diagrama de la Figura 1; en el cual se lleva a cabo de manera secuencial con un lazo de retroalimentación que permite realizar mejoras en el diseño de piezas estructurales para su montaje físico.

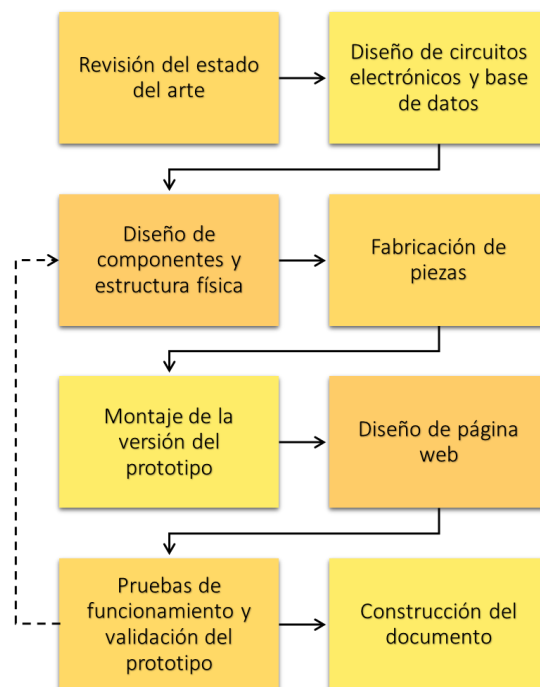


Figura 1. Diagrama metodológico

Fuente: Autor

7.1. Revisión del estado del arte

Esta primera etapa corresponde a realizar todos los procesos de investigación correspondiente a la automatización de sistemas eléctricos en cicloparqueaderos públicos y privados, tanto a nivel nacional como internacional; además de evaluar el nivel de automatismo acorde a las necesidades y a la visión del proyecto.

7.2. Diseño de circuitos electrónicos y base de datos

Al determinar las funcionalidades del proyecto, se selecciona el hardware necesario para su desarrollo; el cual corresponde a un microcontrolador, módulos de comunicaciones y un circuito que permita interconectar estos componentes a los dispositivos de seguridad.

Además, se incluye la elaboración de una base de datos que describa el estado del sistema y registre las modificaciones efectuadas en los puestos del cicloparqueadero.

7.3. Diseño de componentes y estructura física

Se diseñaron los soportes y la cubierta para el prototipo, en los cuales serán ubicados los circuitos y mecanismos encargados del cierre de las correas de sujeción.

7.4. Fabricación de piezas

En esta etapa son realizados los procesos de fabricación necesarios para la construcción de estructuras físicas del prototipo; para ello se modelaron figuras 3D de cada una de las piezas.

En la última versión se empleó una impresora 3D encargada de la construcción de soportes en filamento; además, para la cubierta y los niveles de la estructura, se realizó su fabricación en metal.

7.5. Montaje de la versión del prototipo

Comprende el ensamble de sistemas electrónicos junto con las piezas previamente diseñadas, las cuales, en conjunto, permitan llevar a cabo las funciones determinadas para el sistema "LockBikeRack".

7.6. Diseño de la página web

Con el propósito de implementar un aplicativo que monitoree la disponibilidad en las plazas de parqueo, se diseña una página web como interfaz, la cual reconozca el estado de los cupos de bicicleta y los muestre de manera ordenada e intuitiva.

7.7. Pruebas de funcionamiento y validación del prototipo

En búsqueda del cumplimiento de estándares de calidad, se realizan pruebas que validen el funcionamiento del sistema, para su posterior anclaje a los soportes del cicloparqueadero.

Instalado el prototipo sobre las barricadas, se corrobora el desempeño del sistema “LockBikeRack” a modo de evaluación de comportamiento ante una situación real sistema-usuario.

7.8. Retroalimentación de optimización de versiones

Este lazo de realimentación permite determinar posibles mejoras en la versión del montaje, las cuales contemplan criterios estructurales tales como el cambio de sus proporciones y el remplazo del material de fabricación utilizado. Esto con la finalidad de que se adecúe aún mejor al estrés mecánico y a las condiciones ambientales ante las cuales se exponga.

7.9. Construcción del documento

A través de la redacción de un documento se busca generar un compilado que evidencie los procesos de investigación, diseño y fabricación implicados en el presente proyecto, además de incluir los resultados obtenidos en el proceso y los posibles trabajos futuros que continúen con el desarrollo de sistemas tecnológicos con fines de seguridad.

Capítulo 8

Diseño

El sistema LockBikeRack fue concebido para facilitar el acceso de los usuarios a un cicloparqueadero, permitiéndoles estacionar sus bicicletas de manera segura. Para lograr esto, se ha diseñado un sistema que integra varios módulos y componentes electrónicos especializados.

Inicialmente, se ha identificado la necesidad de un submódulo adaptador de corriente continua, encargado de proporcionar la energía necesaria para el funcionamiento del sistema. Este submódulo garantiza una alimentación estable y segura para los demás componentes electrónicos.

El submódulo controlador de motores (Driver) juega un papel crucial en el sistema, ya que es responsable de activar y desactivar los mecanismos de seguridad que aseguran las bicicletas en su lugar. Este componente se encarga de controlar los motores eléctricos que accionan los pestillos de los cables metálicos, permitiendo su bloqueo y desbloqueo según la autorización del usuario.

La inclusión del submódulo Wi-Fi ESP8266 facilita la comunicación del sistema con la base de datos, permitiendo la validación de las tarjetas NFC utilizadas por los usuarios. Esta conexión inalámbrica asegura una rápida y confiable transmisión de datos, garantizando un acceso al cicloparqueadero.

El submódulo lector de tarjetas RFID RC522 NFC permite la identificación de los usuarios autorizados a acceder al cicloparqueadero. Al acercar una tarjeta con chip NFC al lector, el sistema verifica la información del usuario en la base de datos y activa los mecanismos de seguridad correspondientes.

Además de estos submódulos principales, se han diseñado submódulos adicionales para indicadores visuales y sensores que proporcionan información sobre el estado del sistema y detectan posibles anomalías.

El diseño del sistema LockBikeRack se basa en una estructura metálica diseñada específicamente para soportar el peso del sistema electrónico y los candados. Esta estructura proporciona una base sólida y segura para el funcionamiento del sistema.

En esta sección se describirá detalladamente el diseño de cada uno de los componentes electrónicos, así como las imágenes de simulaciones en 3D de las piezas que componen el sistema. Se incluirán planos electrónicos y detalles del aplicativo desarrollado para el funcionamiento del sistema. Además, se realizarán pruebas para garantizar el correcto funcionamiento del sistema LockBikeRack.

8.1. Diseño e implementación de sistemas electrónicos

El circuito principal está distribuido, tal como se representa en Figura 2. El circuito principal, consta de seis submódulos y dos módulos principales. En el módulo principal se encuentran regletas que redireccionan a los submódulos: lector de tarjetas (RFID RC522), sensores fin de carrera e indicadores, los cuales, no se encuentran incorporados directamente a la base, sin embargo, se encuentran cableados desde su posición de la estructura; los submódulos Alimentación 12V, controlador de motores eléctricos (L293D) y Modulo Wi-Fi (ESP8266) se encuentran incorporados directamente a la tarjeta principal. El segundo modulo principal consta de un Arduino.

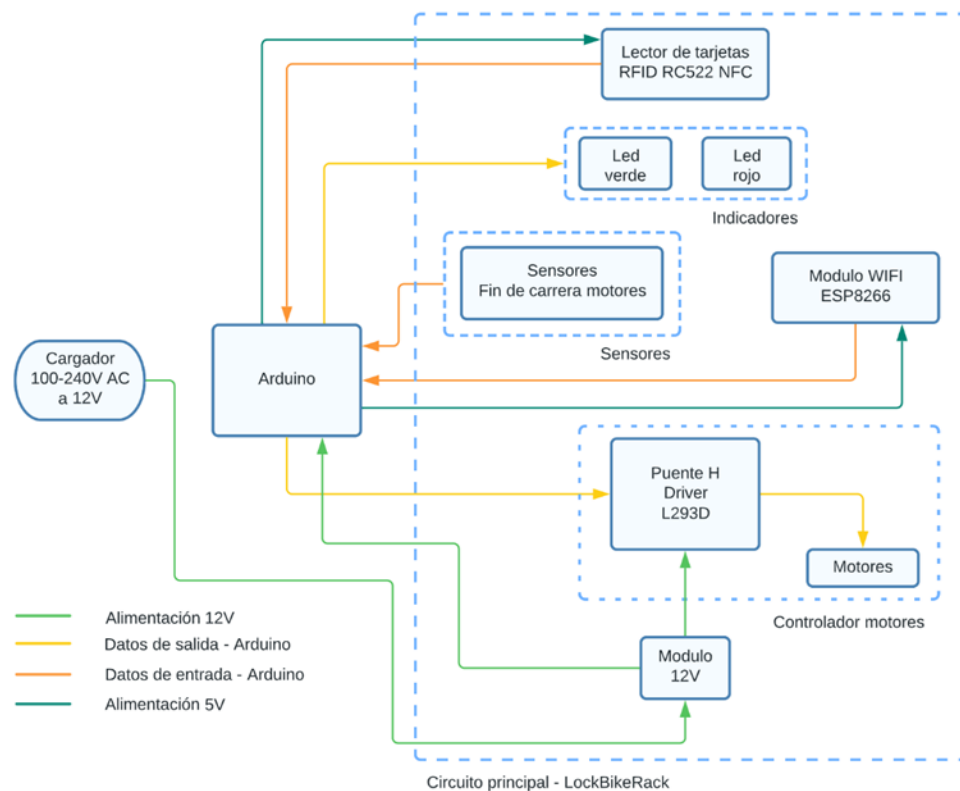


Figura 2. Diagrama de bloques del sistema electrónico - LockBikeRack

Fuente: Autor

8.1.1. Diseño circuito principal

Para el diseño de la placa principal, se otorgó una organización directamente proporcional a su funcionamiento y posición en la estructura. El circuito principal contiene:

1. **Submódulo Jack adaptador de corriente continua Arduino:** A la placa ingresan 12V DC y 1.5Amp con su respectivo GND, de un cargador directamente conectado a un tomacorriente de 100 - 240V AC ~ 50-60Hz, los cuales son compatibles con los voltajes de alimentación establecidos por los submódulos.

2. **Submódulo controlador motores:** En el cual se encontrarán los pines con su respectivo nombre, correspondientes a las salidas de los motores. El controlador L293D que se encarga del movimiento y dirección de estos.
3. **Submódulo Wi-Fi (ESP8266):** Ubicado en la parte lateral derecha de la placa, permite la comunicación vía Wi-Fi entre el módulo Arduino y la base de datos, validando la identidad del usuario y permitiendo el uso del sistema.
4. **Regleta submódulo sensores fin de carrera:** Asentado en la parte lateral izquierda del circuito, con el fin de comunicar los sensores externos con el circuito.
5. **Regleta Arduino:** Situado en el centro de la placa. Llevando la comunicación de las entradas de los submódulos al módulo Arduino.
6. **Regleta submódulo indicadores:** Contemplado en la superficie lateral superior izquierda, comunicando los leds alojados en la estructura.
7. **Submódulo lector de tarjetas RFID RC522:** Localizado en la parte superior de la placa, por la posición del lector en la estructura.

El diseño de este circuito priorizo la organización de los dispositivos electrónicos, que influye directamente en el funcionamiento y posición dentro de la estructura principal, llevando así un orden dentro del mismo sistema como se visualiza en la Figura 3. La función principal de la placa es la transmisión de señales digitales, al contener todos los elementos del circuito dentro de este. Como segunda funcionalidad, contiene la alimentación de los dos módulos del sistema, llevando en si la alimentación de 12 V DC/DC. Las regletas hembra incorporadas para la facilidad de conexión entre el módulo principal y el módulo Arduino.

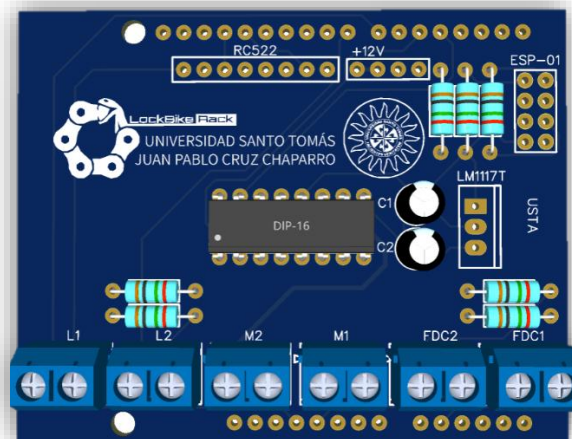


Figura 3. Diseño final PCB, Proteus 8 Profesional - LockBikeRack
Fuente: Autor

8.1.2. Simulación del circuito principal

La simulación para el circuito principal se realizó en el programa Autodesk Tinkercad web, el cual contiene librerías de Arduino, y circuitos electrónicos básicos. Se eligió el programa Tinkercad por contar con librerías de Arduino incluidas por defecto y la integración de programación al módulo Arduino para su simulación en tiempo real. La simulación otorga el comportamiento del circuito en PCB, con la finalidad de contener valores de alimentación, corrientes integradas en el circuito, estructura, y visualización de su comportamiento como se observa en la Figura 4. En el circuito principal se encuentra simulado el módulo Arduino, el submódulo controlador motores, indicadores, submódulo adaptador de corriente continua 12V y submódulo sensores. Se establece una fuente de alimentación de 12 V a 1 A, con la finalidad de conocer una aproximación de la corriente requerida por los motores en funcionamiento.

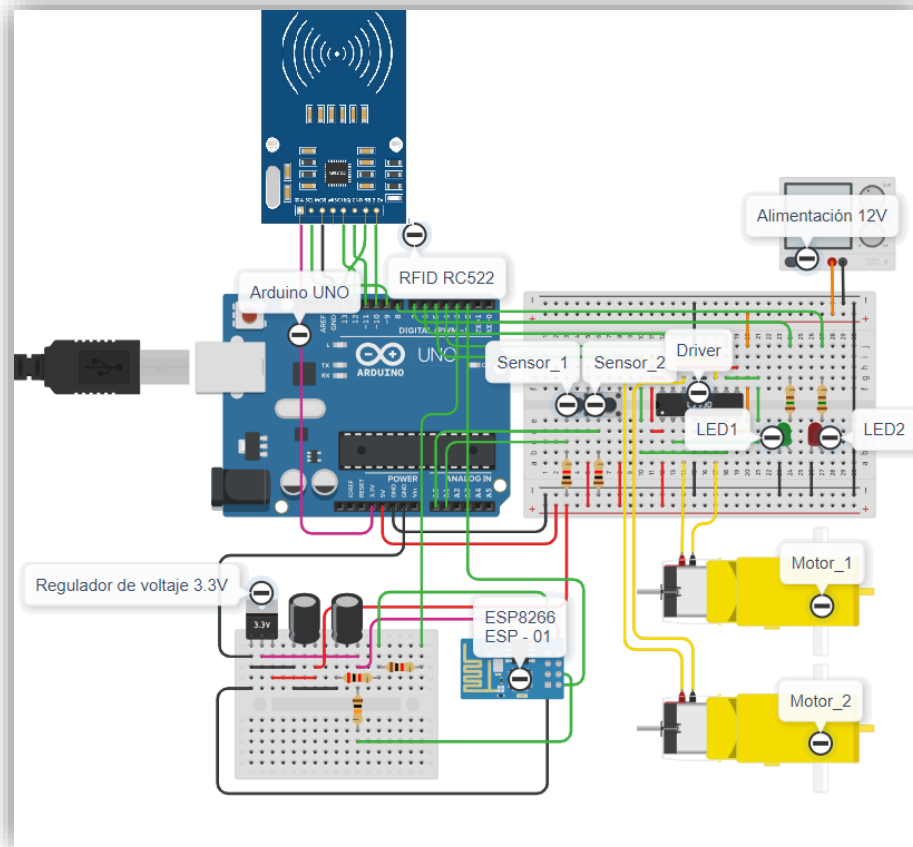


Figura 4. Diagrama 2D del sistema electrónico - LockBikeRack
Fuente: Autor

La simulación del circuito principal se lleva a cabo para el funcionamiento de los componentes en conjunto. Por este motivo se definen 2 motores con voltaje nominal de 9V, driver L293D el cual establece el control de giro de los motores, los indicadores de estado del parqueadero, dependiendo del giro de los motores y sensores de entrada, como lo son los pulsadores que simularán el

comportamiento de sensores de final de carrera, que al detectar la señal de entrada envían una señal al circuito principal y esta, comunicándose con el módulo Arduino para el paso a seguir.

Se procede a generar la programación del Arduino, el cual realizará la acción de lectura y escritura en la simulación. El programa Autodesk Tinkercad contiene la herramienta llamada “Código” en la cual, se puede insertar código de Arduino para realizar simulaciones en tiempo real. Continuando con la programación, se genera un nuevo proyecto en el programa Arduino y se procede a realizar la programación pertinente al circuito a simular.

La programación del módulo circuito principal, se inicia con la creación de las variables que se emparejaran a los pines referenciados en la placa Arduino, en el segmento “Void setup” se describe la acción que contendrá cada pin, contemplando pines de entrada y salida, referenciando esto como OUTPUT a una señal de salida e INPUT a una señal de entrada. Para finalizar en el segmento “Void loop” se describen las acciones que se deben estar contemplando repetidamente, como lo puede ser un condicional, un cambio de estado o un pin de entrada, para generar una acción u otra.

Para realizar el código de programación se parte desde el algoritmo que describe el diagrama de flujo de la Figura 5, el cual se basa en el estado de tres variables booleanas: “Lock”, “M1” y “M2”. La primera indica si ambos pestillos ya han sido cerrados mediante la transmisión de los motores eléctricos y las siguientes denotan si se han accionado ya cada uno de los motores. El comportamiento esperado será consultar constantemente los sensores de final de carrera para así conocer el momento adecuado en el cual corresponda accionar cada uno los motores y asegurar así los candados flexibles una vez estén en una posición correcta para ser enganchados.

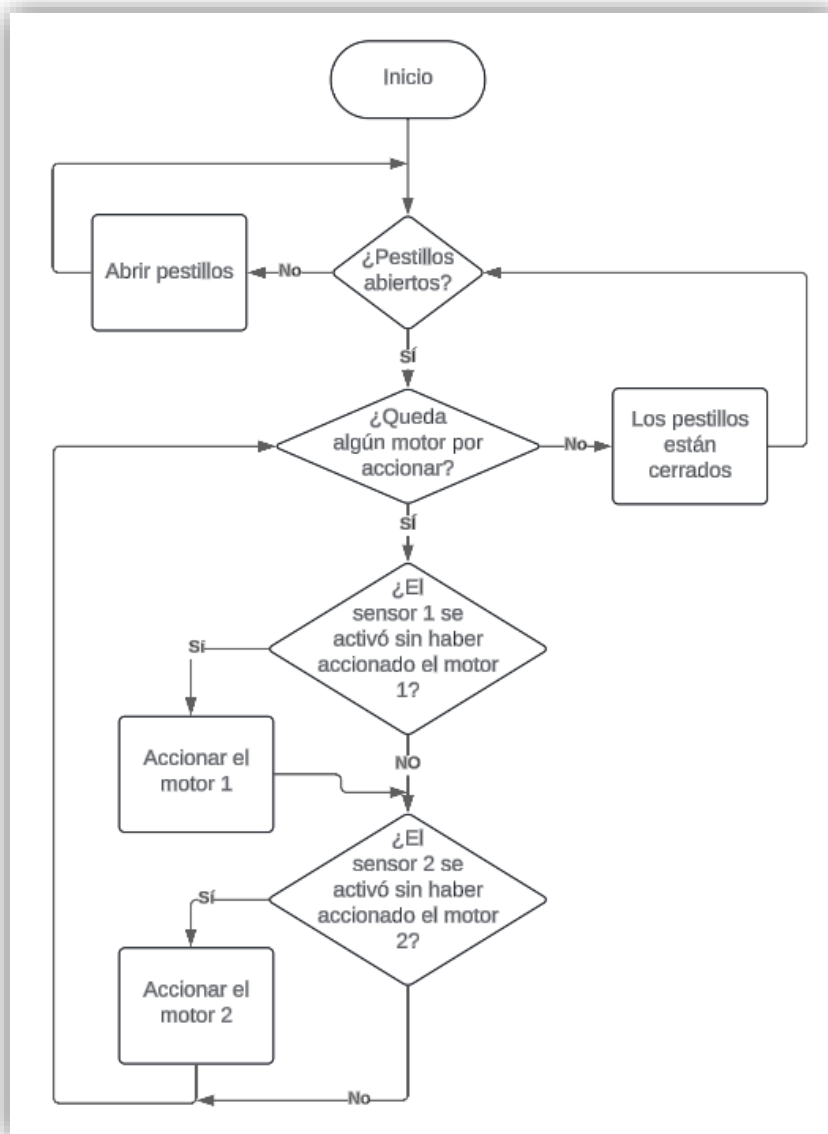


Figura 5. Diagrama de flujo del algoritmo de prueba del circuito principal - LockBikeRack

Fuente: Autor

Al finalizar con la programación del submódulo en cuestión, se continúa el proceso en la herramienta Tinkercad ingresando el código creado en la sección de “Código”. Como paso final se selecciona la opción “Iniciar simulación” y se generará una simulación (Figura 6) en tiempo real del circuito planeado, corroborando que la corriente que exigen los motores mientras actúan simultáneamente no supera a la suministrada por la fuente de alimentación.

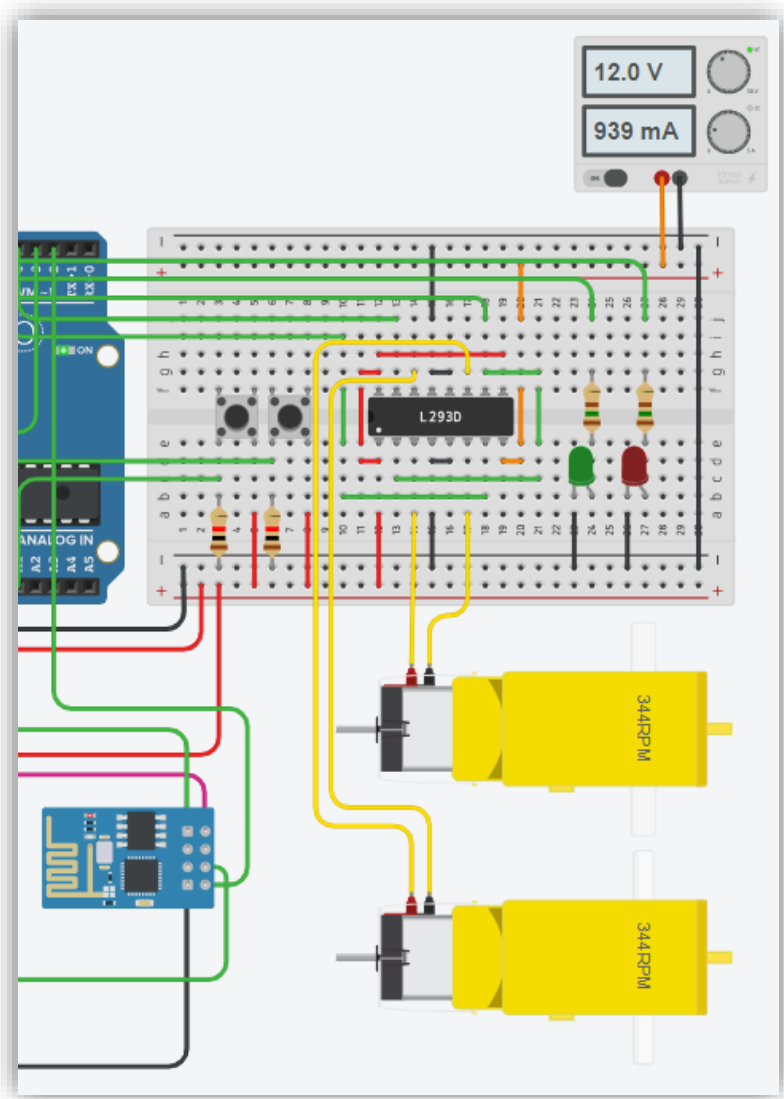


Figura 6. Simulación del algoritmo de prueba del circuito principal – LockBikeRack
Fuente: Autor

En la parte superior derecha del software Tinkercad, se contempla un icono de un integrado, se selecciona el segundo icono “Vista esquemática”, el cual otorga la vista de conexiones del circuito simulado en 2D. El esquemático completo se refleja en la Figura 7.

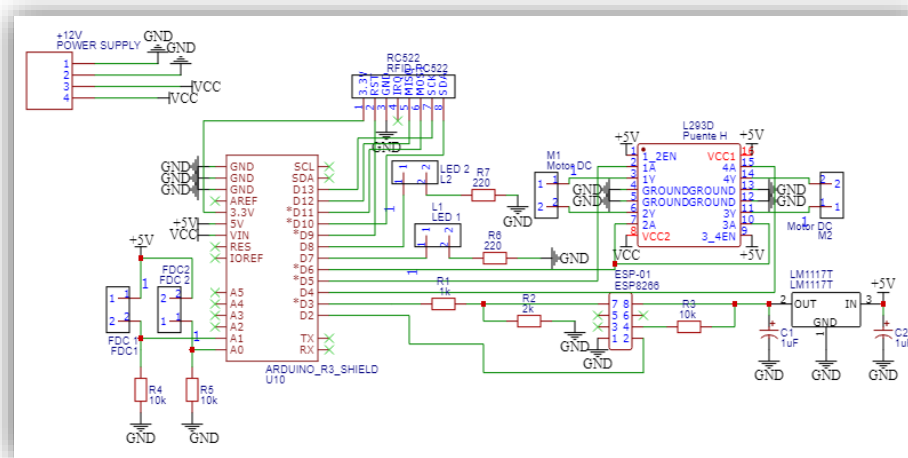


Figura 7. Esquema del sistema electrónico - LockBikeRack
Fuente: Autor

Gracias a la simulación se determinó los tiempos de recorrido de los motores, el tiempo de respuesta del controlador de motores y tiempo de actuación de los sensores.

8.1.3. *Diseño de la PCB*

Para el diseño de la PCB implementado en el sistema LockBikeRack, como circuito principal, se desarrolló utilizando el programa Proteus 8 profesional, utilizado a lo largo de la carrera. El programa cuenta con diversidad de librerías y herramientas para la creación de circuitos impresos; exportaciones de esquemáticos correspondientes a capas seleccionadas y ayudas para la construcción de una placa funcional.

La construcción del circuito impreso contiene parámetros de creación planteados a partir de la norma IPC-2221B[17], los cuales, gracias al programa son contemplados al iniciar con la implementación de este. Los parámetros tenidos en cuenta para el desarrollo fueron:

1. Partir de una simulación funcional.
2. Generación y estructuración de los elementos, para su mayor manejo al momento de crear pistas.
3. Holgura de las pistas, correspondientes a las señales que viajarán por estas. Se estableció un ancho de 30 milésimas de pulgada para las señales del circuito impreso y 15 milésimas de pulgada, para señales de corriente digitales bajas.
4. Generación de malla de cobre a tierra, para evasión de ruido y descargas electrostáticas sobre la placa.
5. Cantidad de recorrido y componentes del circuito, para el uso de capas. Por la cantidad de componentes y recorrido, se implementa para el diseño funcional del circuito impreso LockBikeRack, dos capas. La capa superior elegida para la visualización de los componentes, nombre del autor, imagen del logo de la universidad Santo Tomás y su estructura. La capa inferior principal, es empleada para el enrutamiento de las conexiones.

6. No crear pistas con ángulos de 90 grados.

Con los criterios establecidos se procedió a desarrollar la placa principal del sistema LockBikeRack en el programa Proteus 8 Professional. Para el inicio de la herramienta se debe seleccionar la opción “PCB layout”, ubicado en la parte superior izquierda, a continuación, se debe generar una figura correspondiente a la estructura del circuito a implementar, en el caso del sistema LockBikeRack, se selecciona la capa “Board Edge” ubicado en la parte inferior izquierda del programa, se genera una figura rectangular por el diseño de los elementos. Para la inserción de elementos, se selecciona el submódulo “Package Mode”, en el cual genera una ventana con las librerías instaladas con el producto, se debe realizar la búsqueda del elemento en cuestión y dar doble clic para ajustar su posición dentro de la capa “Top Copper” que referencia la capa superior del esquemático. Con los elementos posicionados, se elige la capa “Bottom Copper” para la creación de pistas que forjan la conectividad de elementos dentro del esquema.

Por los criterios de conectividad, se establece una holgura de 30 milésimas de pulgada para todas las señales del circuito, determinando que las señales que recorren las pistas son señales de corriente digitales bajas. Se enrutan los componentes del circuito para establecer la conectividad como se observa en la siguiente Figura 8.

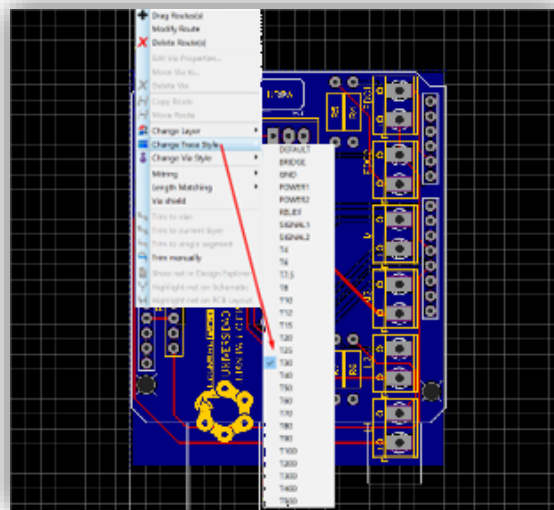


Figura 8. Holgura de las pistas, Proteus 8 Professional - LockBikeRack.

Fuente: Autor

Finalmente se aplica una malla de cobre a tierra seleccionando la herramienta “Power Plane Generator” y se diseña una placa de acrílico, con el propósito de evadir problemáticas con el ruido o descargas electroestáticas. Contemplando la estructura del sistema se realizan orificios en los esquemas de la placa para asegurar su posicionamiento interno y no interfiera el movimiento de la estructura. La estructura final del circuito se puede observar en la Figura 9 donde se muestra la capa superior del circuito principal.

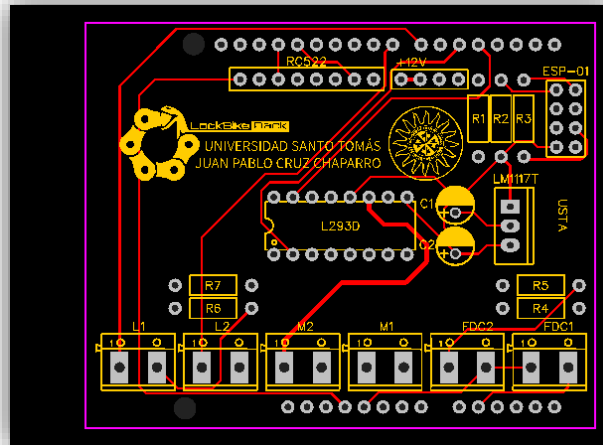


Figura 9. Diseño capa superior PCB, Proteus 8 Profesional – LockBikeRack
Fuente: Autor

En la Figura 10 se muestra la capa inferior de la PCB, la cual aparte de contener algunas conexiones entre los componentes electrónicos, también resalta una gran área correspondiente al plano de tierra del circuito, contribuyendo a la reducción de interferencias y del ruido eléctrico que puedan generar los motores DC.

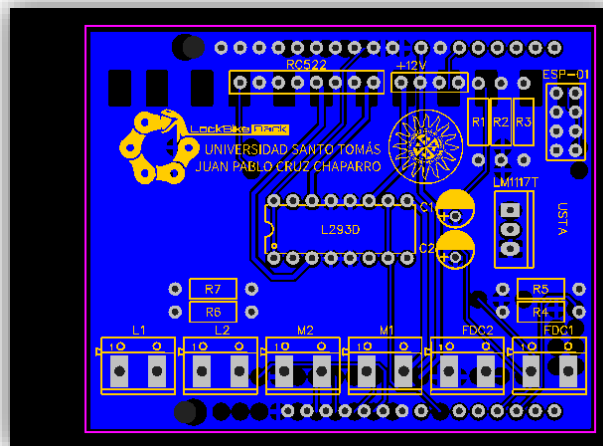


Figura 10. Diseño conexiones capa inferior PCB, impresión – LockBikeRack
Fuente: Autor

En el módulo “3D Visualizer” se forja un bosquejo del circuito en 3D para la visualización final del diseño como se puede observar en la Figura 11, la cual enseña la capa inferior de la PCB.

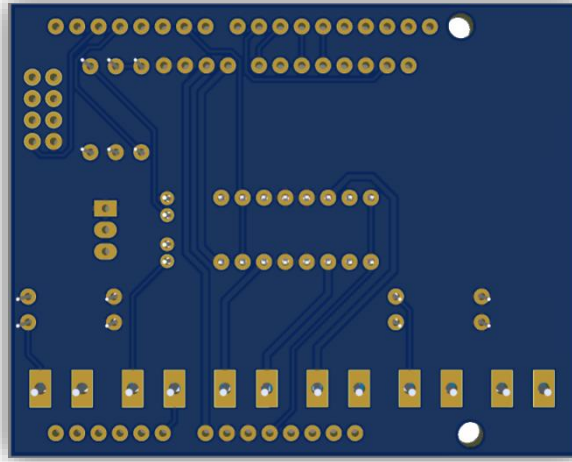


Figura 11. Diseño capa inferior PCB, Proteus 8 Profesional - LockBikeRack
Fuente: Autor

Del mismo modo, en la Figura 12 se puede observar la capa superior de la PCB, incluyendo marcas que indican la disposición de los componentes electrónicos sobre la placa.

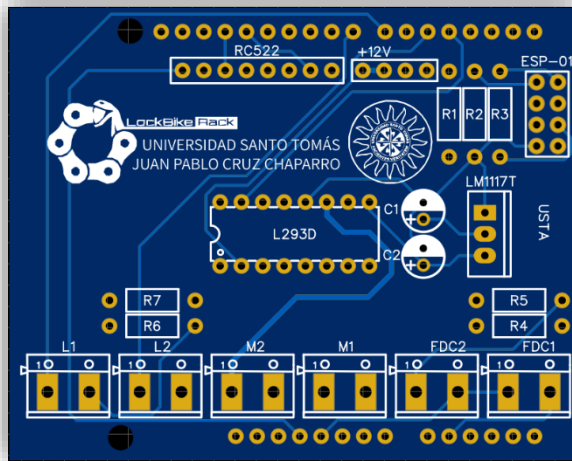


Figura 12. Diseño capa superior esquemático PCB, Proteus 8 Profesional – LockBikeRack
Fuente: Autor

Completando los pasos anteriores, se verifica el estado del circuito creado, verificando que el circuito no contemple errores de conectividad e indicadores de corriente por holgura de las pistas, dichas alertas se encuentran alojadas en la parte inferior derecha de la herramienta. Terminando las

valoraciones del diseño se procede a enviar los planos de impresión a la empresa PCBWay los cuales se encargan de la creación física del circuito impreso. En la

Figura 13 se visualiza uno de los planos requeridos.

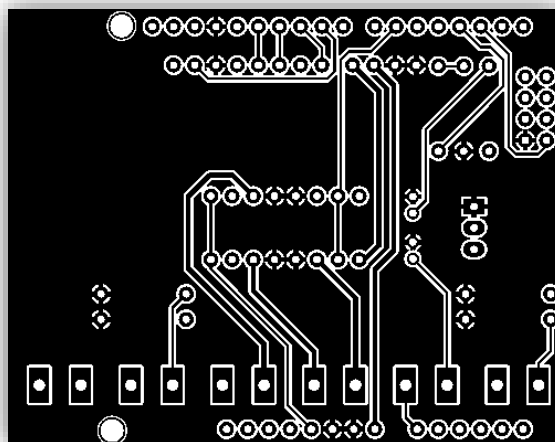


Figura 13. Diseño para impresión física capa inferior PCB, impresión – LockBikeRack
Fuente: Autor

La Figura 14 permite ver con mayor claridad las marcas que serán impresas sobre el circuito, las cuales además de rodear cada componente e incluir un código identificativo para cada uno, también permiten cierto nivel de personalización.

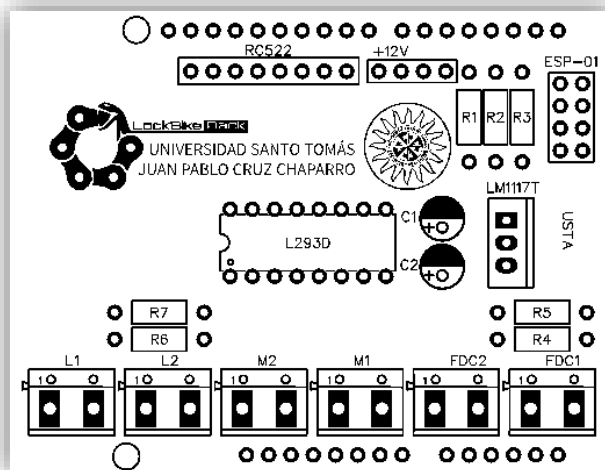


Figura 14. Diseño impresión física capa superior PCB, impresión – LockBikeRack
Fuente: Autor

El diseño es enviado a la empresa de impresión de circuitos, quienes se encargan de realizar una validación previa al circuito antes de crear su estructura física, de esta manera, en caso de

contener algún error en la simulación será retroalimentado para su debida corrección. Para su paso a producción física, se debe tener en cuenta la separación de la malla de cobre de tierra con las pistas, ya que estas pueden quedar unidas y generar un corto en la placa.

8.2. Submódulo conector y adaptador de corriente continua

El módulo Jack de corriente continua como se representa en la Figura 15, es usado para la alimentación del circuito, desde un Jack hembra incorporado en el módulo, de fácil conexión con una fuente de voltaje ya regulada. Para el sistema LockBikeRack se incorpora un adaptador de corriente continua directamente conectado a un tomacorriente de 100 - 240V AC ~ 50-60Hz regulando su voltaje a 12V DC/DC a 1 A. El voltaje que emite el submódulo de corriente continua es el ideal para la alimentación que requiere el módulo principal.



Figura 15. Módulo jack de corriente continua – LockBikeRack
Fuente: Autor

Se opta por el módulo conector Jack de corriente continua por la facilidad de su uso y de su incorporación a una PCB, ya que cuenta con 4 conectores de salida a protoboard, segmentados en 2, como muestra el esquema de la Figura 16. El primer segmento contiene la alimentación positiva de la entrada (En el caso del sistema LockBikeRack, alimentación de 12V), el segundo segmento de la regleta hace referencia a la conexión a negativo otorgado por la fuente.

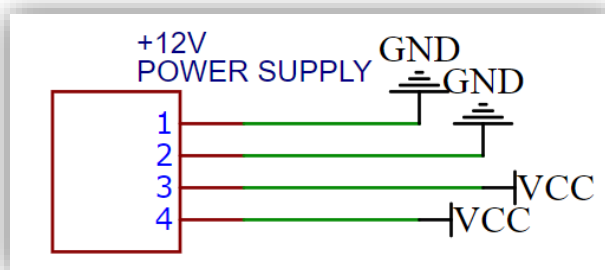


Figura 16. Esquemático del submódulo conector jack DC – LockBikeRack
Fuente: Autor

8.3. Submódulo controlador motores (Driver)

Con el submódulo de alimentación incorporado a la placa, el controlador L293D (Driver para motores eléctricos en configuración puente H), tiene como funcionalidad, dirigir dicha alimentación otorgada al respectivo motorreductor según el pulso que se genere en el submódulo de sensores.

El movimiento de los pestillos está integrado directamente con el funcionamiento del submódulo controlador motores, ya que, al ser accionado un sensor, este genera una señal positiva al Arduino y dependiendo de la programación incorporada, tomará una acción inmediata para enviar información al controlador y que este accione los motores.

El controlador de los motores eléctricos (L293D) consta de 16 pines, de los cuales, los pines 4, 5, 13 y 12, son la referencia a negativo del circuito, las salidas 3, 6, 14 y 11 son los puntos de anclaje de los motores, dependiendo de donde se envíe el pulso, estos rotarán en cierta dirección como se puede visualizar en la Tabla 1. Los segmentos 2, 7, 15 y 10 son las entradas del controlador, las cuales van a indicar por cual canal direccionar el voltaje de alimentación de los motorreductores. Finalmente se tienen los pines 16 y 8, correspondientes a la alimentación del integrado, en el pin 16, se contempla el voltaje normativo del integrado, de igual manera entrega el funcionamiento a la parte digital del integrado, el pin 8 contiene el voltaje nominal de los motores, siendo este la fuente principal del movimiento de los motores.

En la Tabla 1, se establecen las posibles combinaciones lógicas de entrada al sistema LockBikeRack, en los cuales se tienen únicamente contemplados los estados de apertura del pestillo y cierre de pestillo, lo cual referencia a la liberación de la guaya plegable y el anclaje de esta.

Posibles combinaciones lógicas submódulo Controlador motores				
Motor 1		Motor 2		Movimiento
M1 - Derecha	M1 - Izquierda	M2 - Derecha	M2 - Izquierda	
1	0	0	0	Cierre pestillo 1
0	1	0	0	Apertura pestillo 1
0	0	1	0	Cierre pestillo 2
0	0	0	1	Apertura pestillo 2

Tabla 1. Posibles combinaciones lógicas controlador motores - LockBikeRack
Fuente: Autor

Se determina la posibilidad de reducir una salida digital dado que el sistema LockBikeRack permite liberar los candados flexibles simultáneamente, compartiendo un mismo nodo para la

apertura de los dos pestillos como muestra la Figura 17 percatándose de mantener siempre polaridad en las salidas, evitando enviar una señal digital alta para dos terminales de un mismo motor.

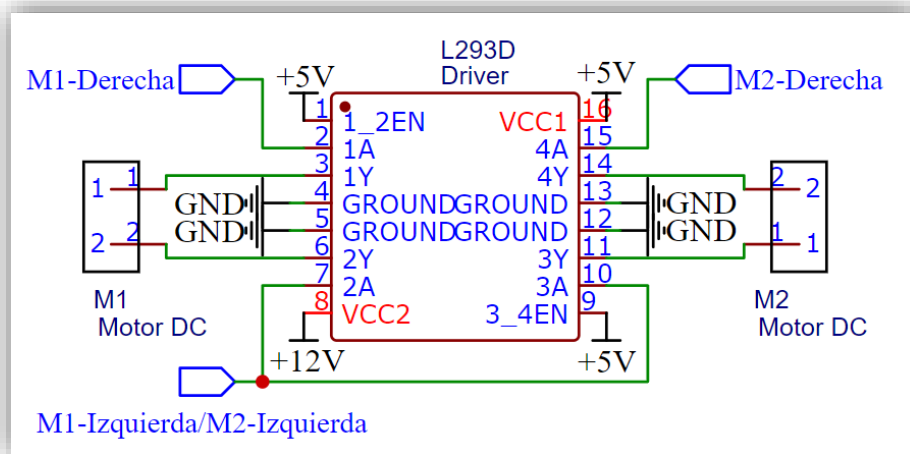


Figura 17. Esquemático submódulo controlador motores – LockBikeRack
Fuente: Autor

Se opta por el integrado L293D (Figura 18) ya que contiene las características ideales para el sistema LockBikeRack, permitiendo el control de 2 motores, entradas digitales para la activación del integrado y funcionamiento, voltajes nominales de 5V para la parte digital y 9V a 12V para la activación de motores. El integrado cumple con las funciones necesarias para el cumplimiento de los objetivos del sistema a implementar.



Figura 18. Modelo 3D submódulo controlador motores (driver) y terminales para los motores M1 y M2 – LockBikeRack
Fuente: Autor

8.3.1. Simulación submódulo controlador de motores (Driver)

La simulación del submódulo controlador de motores se realizó mediante Tinkercad de la misma manera que la simulación del circuito principal, sin embargo, se optó por un diseño reducido que permita destacar las conexiones y funciones del submódulo en cuestión. La simulación del submódulo controlador de motores incluye el integrado principal L293D (Driver), generando las conexiones tanto con el Arduino como las conexiones a los motores como se observa en la Figura 19. Se añaden pulsadores para generar la activación de cada motor y evidenciar el cambio de dirección de los motores posterior al haber activado ambos pulsadores.

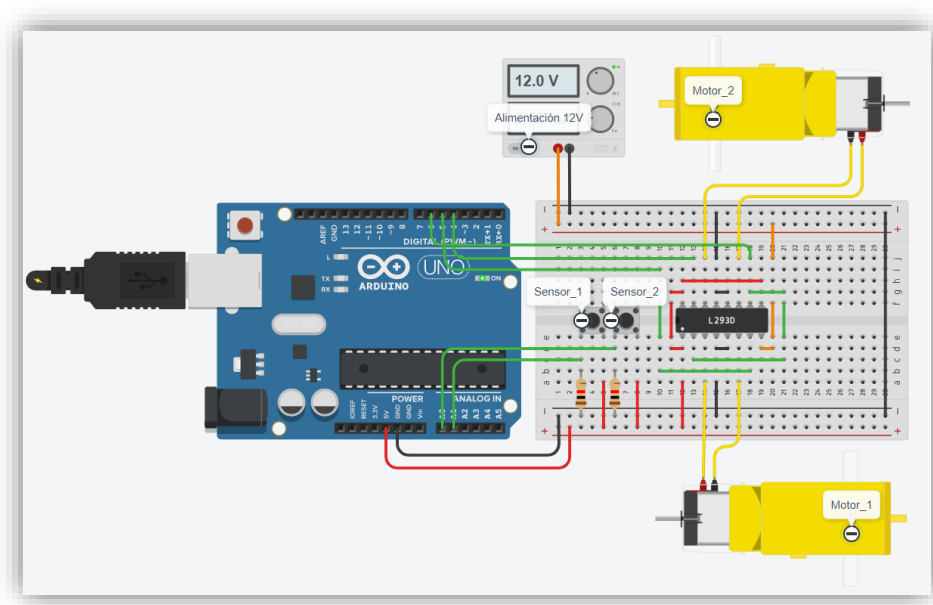


Figura 19. Simulación 2D submódulo controlador motores (driver) – LockBikeRack
Fuente: Autor

Comprobando el comportamiento deseado durante las pruebas de los motores se procede a implementar el diseño del submódulo, para el cual es posible interconectar los elementos mediante pistas sobre el circuito impreso resaltando únicamente puntos de interés como las terminales para conectar los motores eléctricos y el zócalo para el circuito integrado del controlador L293D, mostrados en la Figura 20.

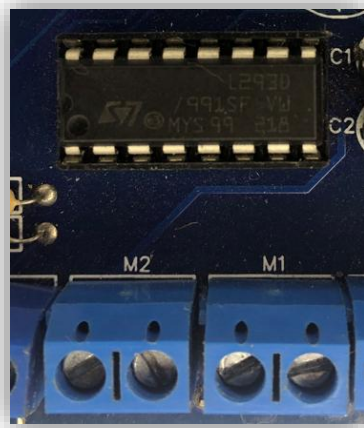


Figura 20. Submódulo controlador de motores (driver) – LockBikeRack
Fuente: Autor

De igual manera se puede contemplar la implementación de los motores eléctricos en la Figura 21 transmitiendo su torque desde un piñón, hacia una cremallera realizando un movimiento lineal bidireccional que mueve el pestillo.

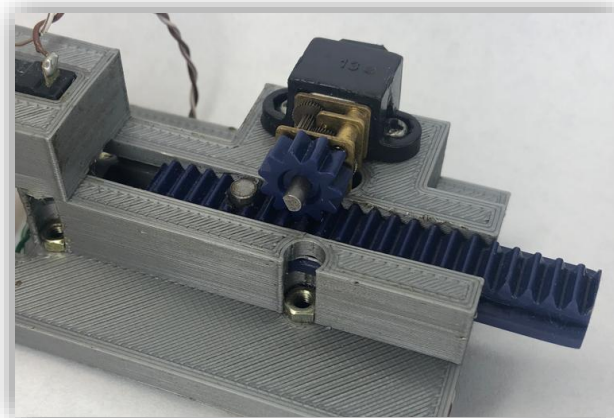


Figura 21. Montaje del motor con mecanismo de piñón a cremallera – LockBikeRack
Fuente: Autor

8.4. Submódulo Wi-Fi ESP8266

Contemplando la finalidad del funcionamiento del sistema LockBikeRack, se plantea un submódulo capaz de comunicarse con la base de datos mediante Wi-Fi y realizar las peticiones al servidor web, consultando información relevante tal como la disponibilidad del puesto al que corresponde el prototipo en la barricada y la existencia del usuario registrado por el lector NFC en la

base de datos. Dado que la autenticación de identidad implica un factor primordial para promover la seguridad del biciusuario, es esencial el uso de un microcontrolador que cuente con una antena Wi-Fi y con terminales de comunicación Rx y Tx para transmitir información hacia el Arduino UNO.

El módulo Wi-Fi ESP8266 es un chip con conexión a internet con el protocolo TCP/IP. Dicho modulo tiene como principal objetivo dar acceso a la red a cualquier circuito o estructura. Una de las características más destacadas del módulo es la conectividad de circuitos electrónicos a aplicativos webs, por su tamaño, su bajo consumo y su facilidad de uso. El chip utiliza un voltaje de operación entre 3V a 3.6V, los cuales son extraídos del módulo Arduino, una corriente de operación de 80mA y una temperatura de operación entre -40°C y 125°C.

Debido a que los pines de propósito general del Arduino UNO configurados como salidas digitales suministran un voltaje de 5V en su nivel lógico alto, es requerido un divisor de tensión para la terminal de comunicación Rx del Módulo ESP8266, además del regulador de voltaje LM1117T a 3.3V que mantenga los parámetros de operación para energizarlo. Se añaden también condensadores de 1uF a la entrada y la salida del regulador de voltaje para mantener niveles de voltaje estables, tal como sugiere la configuración compartida por el fabricante. La configuración completa del submódulo Wi-Fi ESP8266 se representa en el esquemático de la Figura 22 a continuación.

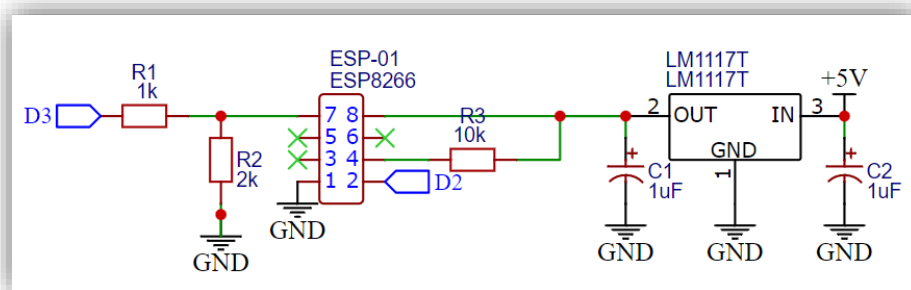


Figura 22. Esquemático submódulo Wi-Fi ESP8266 – LockBikeRack
Fuente: Autor

8.4.1. Simulación submódulo Wi-Fi ESP8266

La simulación del submódulo Wi-Fi ESP8266 a través del software Tinkercad permite verificar la regulación de voltaje adecuada, evitando daños en el módulo que puedan comprometer su funcionamiento. La Figura 23 muestra un desempeño adecuado de los componentes, asegurando 3.3V para alimentar el ESP8266, sin embargo, para comprobar la capacidad de conexión mediante la red Wi-Fi es necesario programar el dispositivo y ponerlo a prueba ante un escenario real.

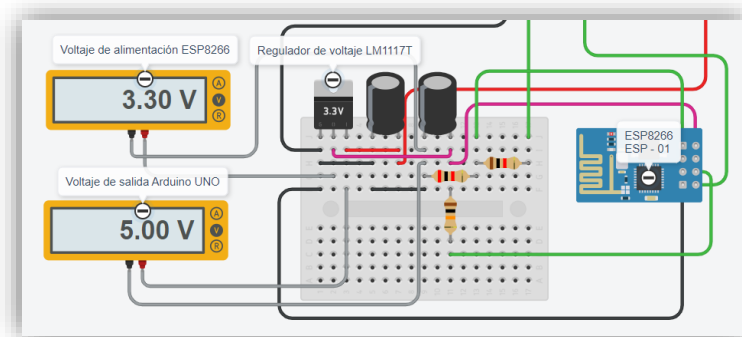


Figura 23. Simulación submódulo Wi-Fi ESP8266 con regulador de voltaje – LockBikeRack
Fuente: Autor

La programación del submódulo Wi-Fi precisa de un adaptador serial que comunique vía USB con el software de Arduino, el adaptador USB CH340 cuenta con los terminales correspondientes a los pines del módulo Wi-Fi ESP- 01. Para configurar el módulo Wi-Fi en modo de programación debe existir una conexión entre el pin GPIO0 a GND, por lo que se realiza una modificación al adaptador USB, incluyendo un interruptor entre los dos puntos como se observa en la Figura 24.

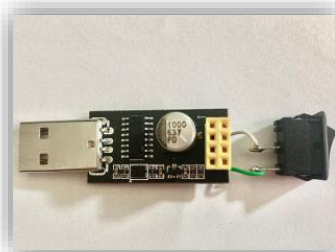


Figura 24. Adaptador USB CH340 – LockBikeRack
Fuente: Autor

La Figura 25 muestra el módulo Wi-Fi ESP8622 en su primera versión (ESP-01) compatible con el adaptador USB CH340, es posible programarlo incrustándolo en los pines del adaptador.

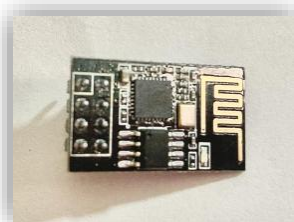


Figura 25. Módulo Wi-Fi ESP8266 – LockBikeRack
Fuente: Autor

El algoritmo de programación que ejecuta el submódulo Wi-Fi inicialmente se tiene que conectar a una red Wi-Fi mediante identificador de red SSID y contraseña, luego consulta la existencia de datos en el buffer del puerto serial, de ser así, lee los datos e interpreta su contenido, si en el puerto serial ha recibido el mensaje "EXIT", el módulo realizará una petición de tipo POST con el número del puesto de cicloparqueadero a la dirección "registro_salida", de lo contrario tomará el mensaje recibido en el puerto serial como el identificador único de la tarjeta "UID" y lo enviará junto con el puesto mediante una petición POST a la dirección "registro_entrada"; en ambos casos esperará la respuesta del servidor y la enviará al Arduino UNO. El diagrama de flujo de la Figura 26 sintetiza el algoritmo descrito.

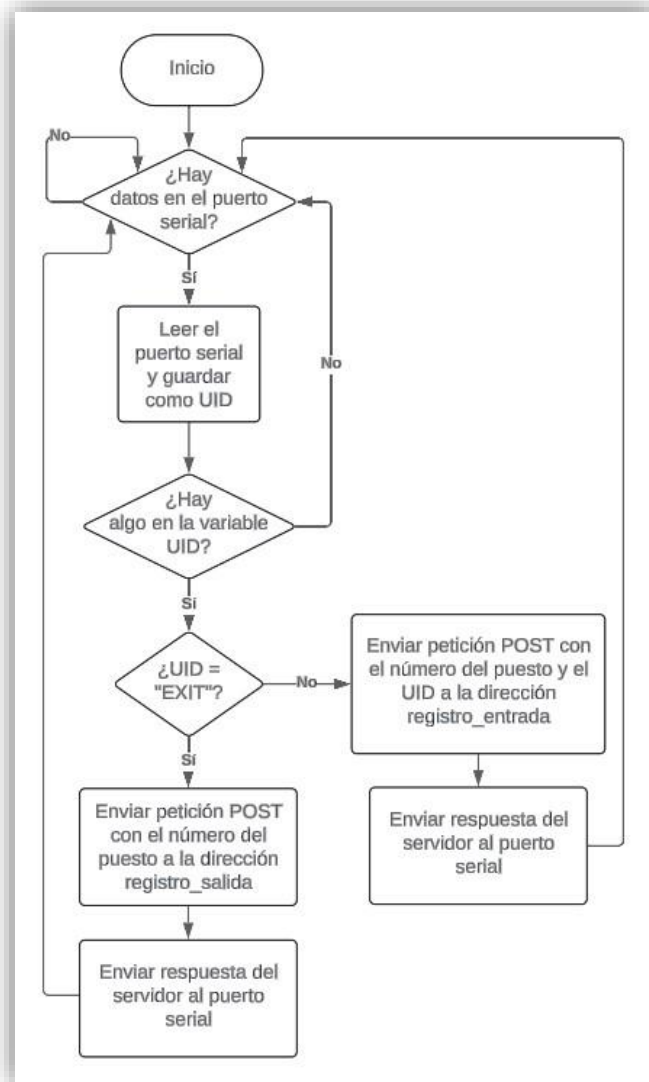
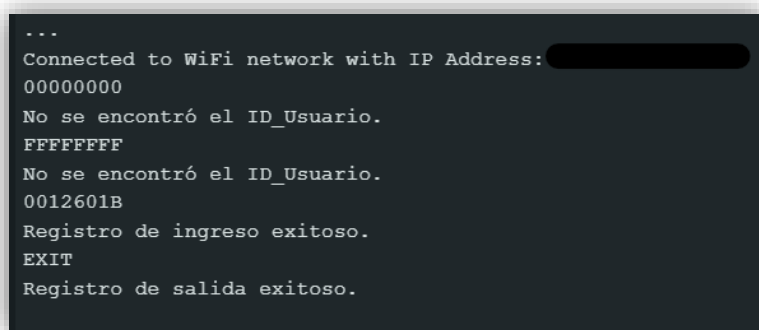


Figura 26. Diagrama de flujo del algoritmo de prueba del módulo ESP8266 - LockBikeRack
Fuente: Autor

Para la programar el submódulo Wi-Fi se debe seleccionar como placa principal en el programa Arduino, por lo cual se debe ingresar a la sección de “Herramientas”, dentro de la sección de placas se selecciona la placa “Generic ESP8266 Module” debido a que se está indicando que la programación del componente se realizará por protocolos. Para la simulación y prueba del funcionamiento del dispositivo, se generará una programación que realiza una petición a la página web LockBikeRack compartida mediante el host local.

El servidor cuenta con scripts de PHP que se encargan de realizar la conexión con la base de datos y realizar las consultas pertinentes, tales como verificar la existencia del usuario dentro de la base de datos y modificar el estado del puesto correspondiente del cicloparqueadero.

En el monitor serial, como se observa en la Figura 27, contiene el resultado de la programación del módulo ESP8266, como información relevante de la conexión, códigos digitados manualmente simulando la interacción con el Arduino UNO y respuestas del servidor en distintos casos.



```
...  
Connected to WiFi network with IP Address: [redacted]  
00000000  
No se encontró el ID_Usuario.  
FFFFFFF  
No se encontró el ID_Usuario.  
0012601B  
Registro de ingreso exitoso.  
EXIT  
Registro de salida exitoso.
```

Figura 27. Salida monitor serie Arduino, submódulo Wi-Fi ESP8266 – LockBikeRack

Fuente: Autor

Para el caso de la prueba, el monitor muestra el mensaje “Connected to WiFi network with IP Address: ...” cuando se consigue una conexión exitosa con la red, luego se digitan tres códigos en el monitor serial, siendo los dos primeros códigos UID erróneos que no pertenecen a la base de datos, obteniendo como respuesta del servidor el mensaje “No se encontró el ID_Usuario.”, consecutivamente se ingresa un código UID perteneciente a la base de datos, correspondido con el mensaje “Registro de ingreso exitoso.”.

Finalmente se ingresa el mensaje “EXIT”, el cual indica la salida del biciusuario, habilitando las operaciones respectivas en base de datos y respondiendo con “Registro de salida exitoso.”.

8.5. Submódulo lector de tarjetas RFID RC522 NFC

El submódulo lector de tarjetas RFID RC522 NFC representa un componente fundamental para el sistema LockBikeRack, pues permite registrar el ingreso y la salida del usuario, brindando una mayor seguridad mediante la autenticación de identidad comparando la tarjeta leída con las tarjetas registradas en la base de datos.

Los sistemas de radio frecuencia en la actualidad están generando un aumento de uso en los sistemas de identificación, sistemas de seguridad, acceso personal y logística de productos como lo pueden ser llaves de puertas eléctricas. Ya que trabajan por señales de radio frecuencias, generando un identificador único y generando acceso a este, por su identificador único.

El funcionamiento del submódulo se basa en etiquetas RFID o tarjetas pasivas. Cuando se acerca una tarjeta al módulo, este detecta el campo magnético y, a través de la antena del módulo, alimenta el chip de la tarjeta y simultáneamente lee la información almacenada en la etiqueta del RFID pasivo. A esta forma de conectividad se le denomina retrodispersión.

El diseño del submódulo lector NFC consta de alimentación mediante 3.3V y su referencia a tierra, además de conexiones entre pines de propósito general del Arduino UNO y los pines del módulo RFID RC522 como se muestra en la Figura 28, asignando la salida D9 como pin RST el cual permite reiniciar el módulo, la salida D10 como pin SDA para la comunicación serial, la D11 y D12 como MOSI (Master Out Slave In) y MISO (Master In Slave Out) respectivamente para el modelo de comunicación Maestro-Esclavo y la salida D13 como SCK que actúa como reloj para sincronizar la comunicación serial.

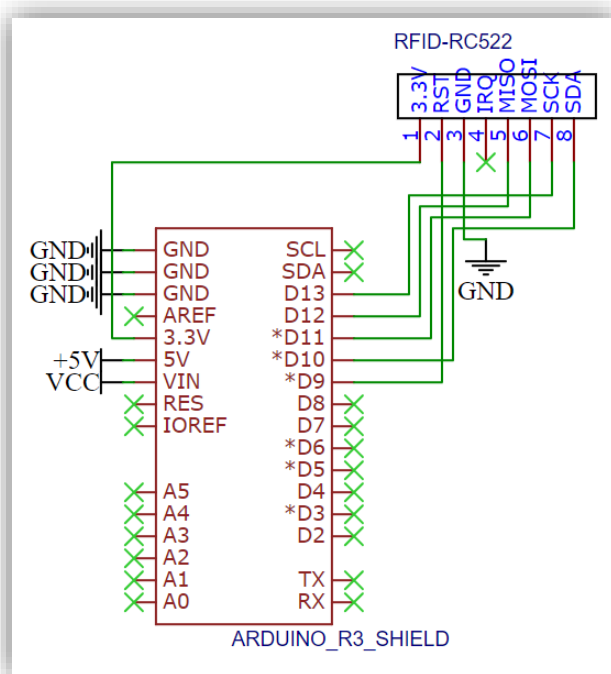


Figura 28. Esquemático submódulo lector de tarjetas RFID RC522 – LockBikeRack
Fuente: Autor

8.5.1. Simulación submódulo lector tarjetas RFID RC 522 NFC

La simulación del submódulo lector de tarjetas RFID RC522, es modelada mediante Tinkercad indicando las conexiones entre el módulo y el Arduino UNO, mostrado por la Figura 29, sin embargo, el software no cuenta con una herramienta que permita simular la interacción con una tarjeta con

chip NFC, en su defecto se plantea programar el montaje experimental para realizar una prueba de comunicación semejante a la función esperada a desempeñar por el submódulo en conjunto del sistema LockBikeRack.

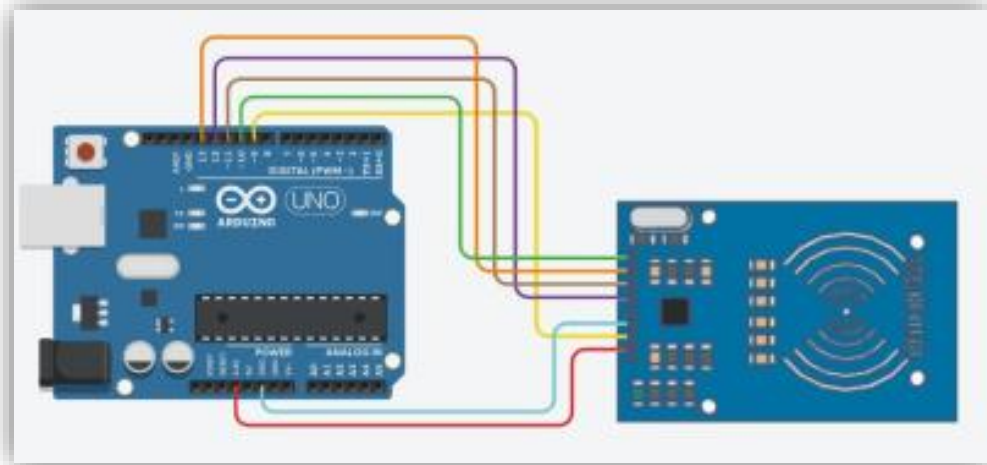


Figura 29. Simulación 2D lector de tarjetas RFID RC522 – LockBikeRack
Fuente: Autor

En cuanto a la programación del Arduino UNO representado por el diagrama de flujo de la Figura 30 son descritas una serie de condiciones que en conjunto garantizan el panorama adecuado para la lectura exitosa del código UID de una tarjeta NFC, las cuales de ser satisfactorias indican la posibilidad de guardar en memoria el código UID de la tarjeta, permitiendo comparar y validar que si el sistema está en uso, únicamente la persona que ingresó su bicicleta al puesto del cicloparqueadero, tenga acceso a retirarla mediante la liberación de los candados flexibles.

Para el diseño definitivo del sistema se ha determinado como medida de seguridad que el dispositivo en cuanto es encendido, primero consulte si el puesto al que corresponde se encuentra ocupado por una bicicleta, esto mediante una petición hacia el servidor, con el objetivo de que el prototipo determine su estado en caso de un corte energético.

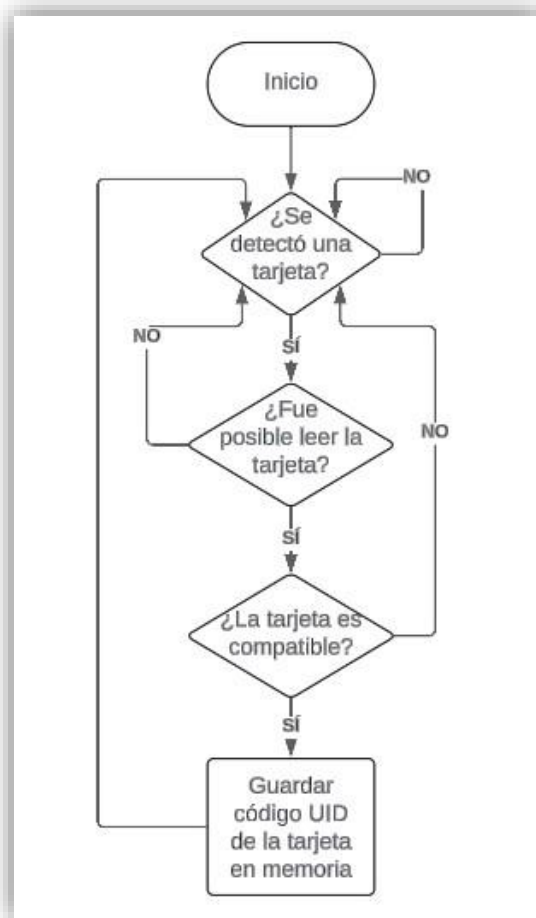


Figura 30. Diagrama de flujo del algoritmo de prueba del submódulo RFID-RC522 – LockBikeRack
Fuente: Autor

El diseño del proyecto contempla añadir un segundo módulo lector de tarjetas NFC externo, cuya finalidad sea leer el código UID para completar el registro de nuevos usuarios, éste módulo externo implica que el proceso de registro será asistido por un administrador con privilegios de gestión en la base de datos, sin embargo, la página web cuenta con un formulario de registro que permite crear un usuario nuevo con la particularidad de que no existirá un código UID vinculado a su perfil hasta completar el proceso.

El lector NFC del prototipo de sistema LockBikeRack se ubica en la esquina superior frontal de la cubierta estructural como se ve en la Figura 31, permitiendo al bicisuario colocar su tarjeta sobre el lector para guardar o retirar su bicicleta.

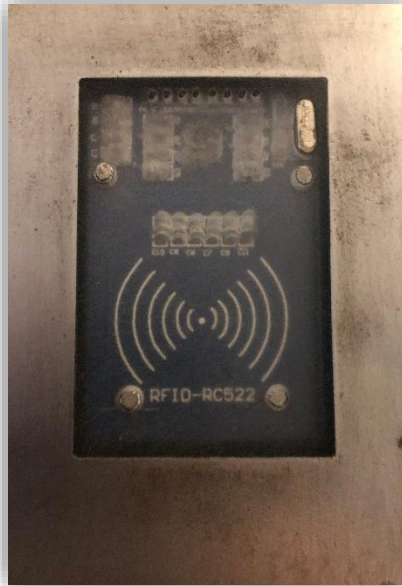


Figura 31. Submódulo lector de tarjetas RFID RC522 – LockBikeRack
Fuente: Autor

Para la interacción con el submódulo lector de tarjetas RFID RC522, el usuario se debe contar con una tarjeta con chip NFC. El UID será el código identificador para cada tarjeta, el cual permite distinguir la tarjeta de cualquier otra, ya que este código es único para cada una, en la Figura 32 se muestra una tarjeta a la cual se incluyó un rótulo con su respectivo número de UID. El proceso que debe realizarse es, acercar aproximadamente 3cm de la tarjeta con el chipo NFC al módulo RFID RC522, ya que el módulo RC522 tiene una lectura de 10cm de lejanía especificado en las características. Sin embargo, por pruebas realizadas en el sistema, se obtiene que este debe encontrarse a una menor distancia, ya que no actúa al estar tan lejos la tarjeta del módulo.

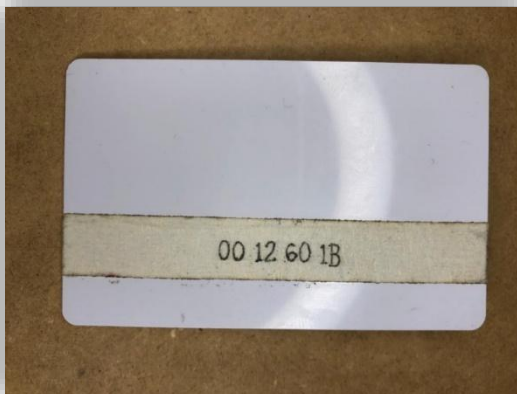


Figura 32. Tarjeta para lector RFID RC522 – LockBikeRack
Fuente: Autor

En el monitor serial, como se observa en la Figura 33, contiene la salida de la programación, como parámetros visibles o escrituras en el panel. Para el caso de la simulación y prueba del RFID RC522 se define un condicional que muestre si obtuvo una conexión exitosa con la tarjeta ingresada y de igual manera, que esta le otorgara la respuesta positiva “Abrir puerta”.

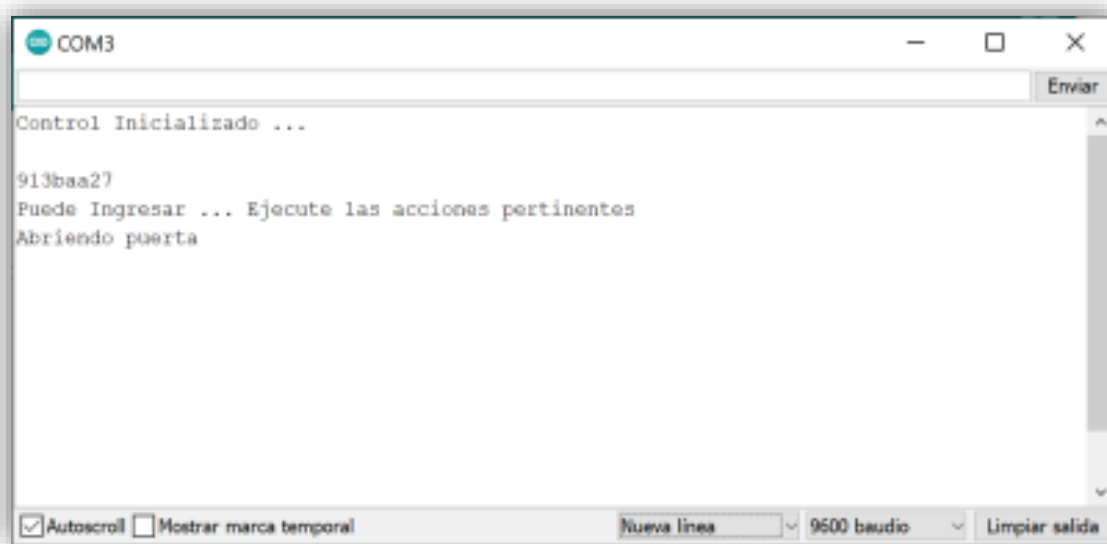


Figura 33. Salida monitor serie Arduino, lector de tarjetas RFID RC522 – LockBikeRack
Fuente: Autor

El uso de un lector NFC es determinado mediante el objetivo de brindar al bicisusuario una experiencia similar a la que podría experimentar en su día a día, realizando un procedimiento semejante al que haría para ingresar a un medio de transporte público masivo, a una institución académica o a sus oficinas de trabajo.

8.6. Submódulo de indicadores

La estética del sistema LockBikeRack es una parte fundamental, que le brindará al usuario información y una perspectiva de su funcionamiento. Se opta por capas de colores verde y rojo; El led verde indicará la posición libre, ya que el ojo humano asimila el color verde a la naturaleza espacios libres en los que se puede posicionar, el indicador led rojo se seleccionó de manera que a primera vista y por la segmentación de colores que recrea el ser humano, el rojo es un color de advertencia, de peligro o de no implicación con ese espacio [18]. Para brindar información visual al usuario se seleccionaron los indicadores leds de 10mm de 3V a 20mA.

Los pines digitales del Arduino son capaces de brindar 20mA de corriente continua cuando se encuentran en nivel alto (5V), por lo tanto, es necesario incluir una resistencia en serie para garantizar que la corriente no supere estos valores. Para calcular el valor de la resistencia en serie a cada led, se utiliza la siguiente expresión:

$$R = \frac{5v - 3v}{20mA} = 100\Omega$$

El esquema de conexiones del submódulo de indicadores que se visualiza en la Figura 34 consta de terminales conectadas entre los pines de las salidas digitales y resistencias cableadas a tierra, esto con la intención de conectar los LED indicadores con cables de 30cm para facilitar su colocación sobre las caras de la cubierta estructural más lejanas.

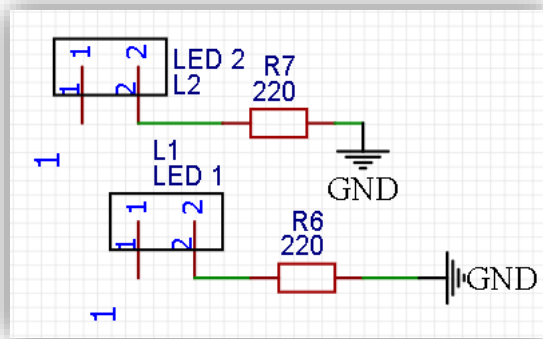


Figura 34. Esquemático submódulo indicadores – LockBikeRack
Fuente: Autor

8.6.1. Simulación submódulo de indicadores

Para la simulación del submódulo Indicadores, se utilizó el componente Arduino UNO cuyos pines de salida digitales permiten alimentar a los leds, se realiza una conexión a una resistencia en serie entre el led y el conector a tierra del Arduino para cada led, tal como se aprecia en la Figura 35.

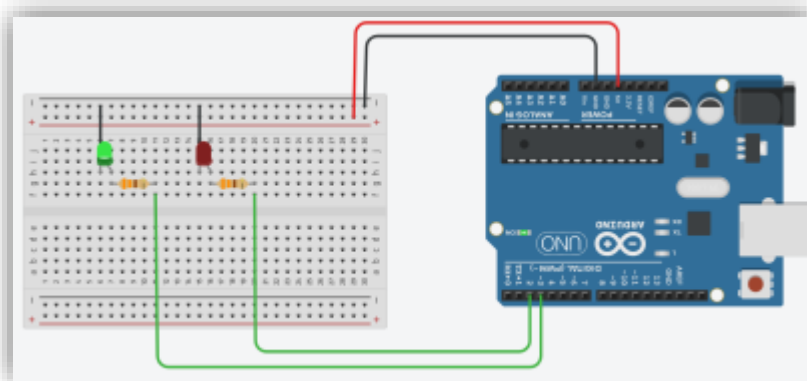


Figura 35. Simulación 2D submódulo indicadores – LockBikeRack
Fuente: Autor

La programación del submódulo de indicadores está relacionada con el comportamiento del submódulo lector NFC como se visualiza en el diagrama de la Figura 36, debido a que su estado depende directamente de la etapa a la que corresponda la lectura de la tarjeta, es decir, si es la primera vez que es leída dicha tarjeta, los indicadores responden encendiendo el LED de color rojo para representar que ahora el sistema está ocupado por un usuario. Por el contrario, si ya había registrado una tarjeta previamente, existirán dos alternativas, el usuario regresa a retirar su bicicleta por tanto el LED rojo se apaga y el LED verde se enciende indicando que el puesto está ahora disponible, o en su lugar, alguien intenta utilizar una tarjeta distinta, por lo cual el sistema deberá señalar el error mediante el parpadeo del LED rojo.

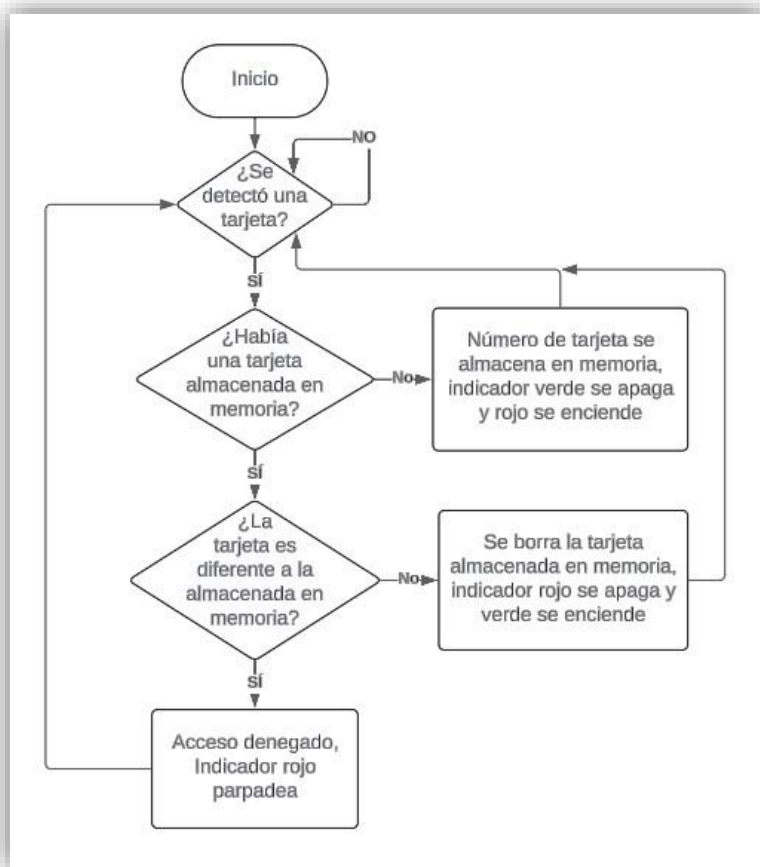


Figura 36. Diagrama de flujo del algoritmo de prueba del submódulo de indicadores – LockBikeRack
Fuente: Autor

8.7. Submódulo de sensores

Para activar el movimiento de los motores, es necesario el uso de sensores de final de carrera que permitan conocer el estado del candado flexible e interpretar si se encuentra o no en la posición adecuada para la activación de los pestillos y mantener una posición de cierre. Los sensores del sistema generan la señal digital para el procesamiento de la información en el Arduino UNO. Los

sensores de final de carrera son usualmente empleados para detectar la posición final de un objeto, indicando que el movimiento ha llegado a la posición final o inicial del recorrido.

Los fines de carrera están compuestos por un resorte, el cual actúa como indicador de posicionamiento del indicador, un soporte externo, donde se aloja el mecanismo, una leva de accionamiento que se activará cuando el resorte se encuentra completamente comprimido y un contacto móvil, que se encuentra directamente relacionado al contacto normalmente cerrado.

El submódulo de sensores que describe la Figura 37 indica la conexión entre el pin de entrada digital y una resistencia en configuración pull down garantizando la lectura de un valor lógico digital bajo hasta que se active el sensor de final de carrera, cerrando el circuito e indicando ahora un valor digital alto.

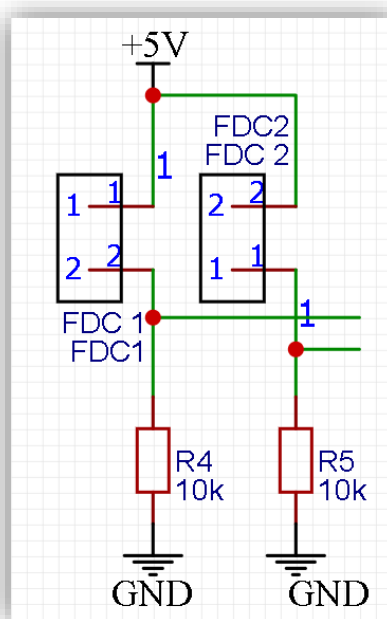


Figura 37. Esquemático submódulo sensores – LockBikeRack
Fuente: Autor

En la Figura 38 se visualiza un modelo en 3D del sensor de final de carrera empleado para el diseño del sistema LockBikeRack y los soportes de pestillos.

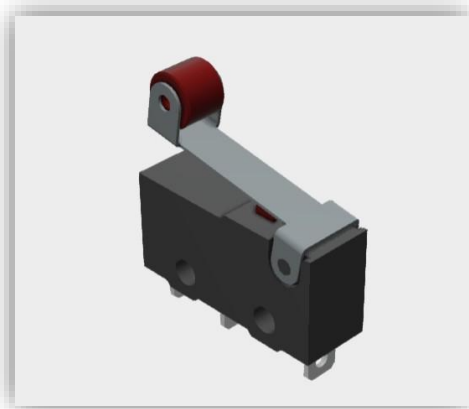


Figura 38. Modelo 3D del sensor de final de carrera – LockBikeRack
Fuente: Autor

8.7.1. Simulación submódulo de sensores

Para la simulación del submódulo de sensores se modela el circuito en 2D mediante el software Tinkercad tal como se aprecia en la Figura 39, los pulsadores representan el comportamiento de los sensores de final de carrera en una configuración normalmente abierto, esto permite reconocer el momento en el que los candados flexibles se encuentran en la posición adecuada para cerrar los pestillos y asegurar la bicicleta del usuario

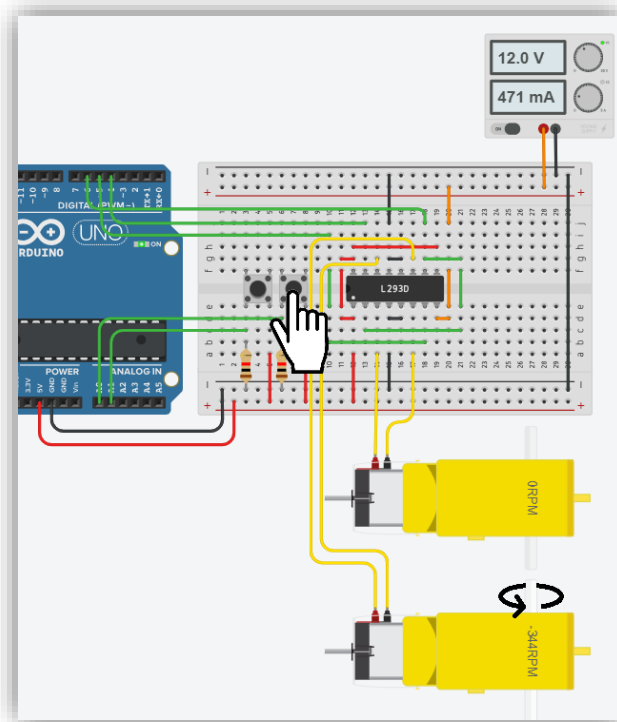


Figura 39. Simulación 2D submódulo sensores – LockBikeRack
Fuente: Autor

La programación del submódulo sensores que se muestra por medio del diagrama de flujo de la Figura 40, consta de varios lazos cuyas condiciones dependen de la activación de ambos motores, dicha activación se realiza únicamente si cada uno de los sensores ha sido presionado por más de un segundo, tiempo suficiente para que el movimiento del motor desplace el pestillo asociado hasta su posición final, impidiendo que los candados plegables sean retirados. Para la simulación planteada ambos motores girarán en la dirección opuesta, tan pronto como ambos pestillos se encuentren cerrados, retornándolos a su posición original.

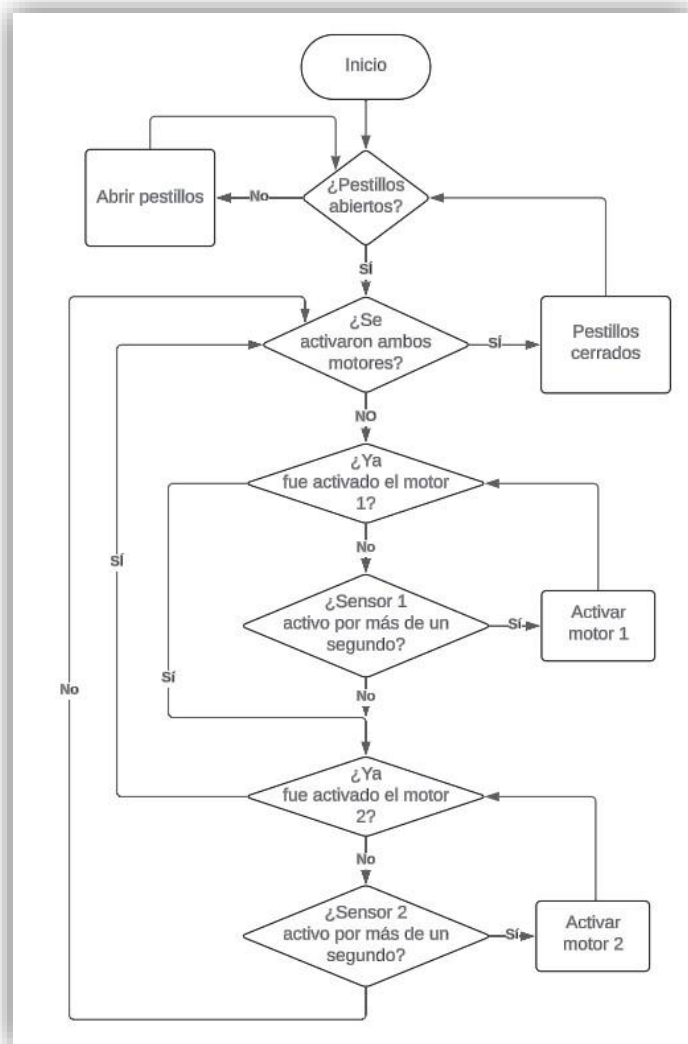


Figura 40. Diagrama de flujo del algoritmo de prueba del submódulo de sensores – LockBikeRack
Fuente: Autor

Los sensores de final de carrera son integrados a los soportes de pestillos como se muestra en la Figura 41 donde serán activados cada vez que el usuario introduzca los candados plegables en sus respectivas ranuras.

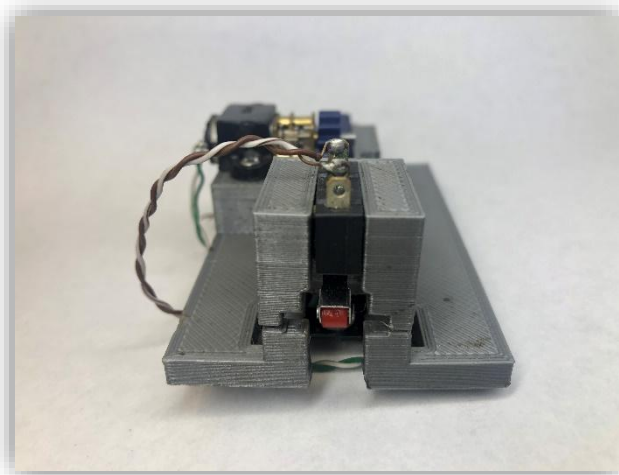


Figura 41. Sensor de final de carrera fijado al soporte de pestillo – LockBikeRack
Fuente: Autor

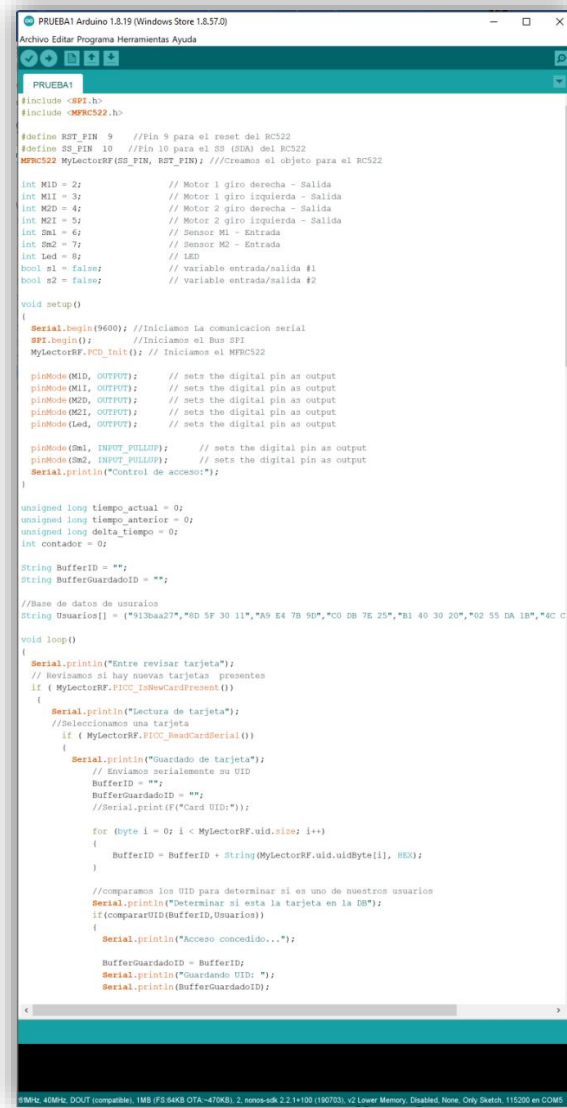
8.8. Integración de componentes del sistema LockBikeRack

Una vez terminado el diseño de los componentes electrónicos que conforman el sistema LockBikeRack, se procede a realizar la corroboración de la unión de los dispositivos, el comportamiento de cada elemento al estar incorporados en el mismo segmento. Para ello se inicia con la implementación de una simulación y la programación final del sistema, el cual abarca la unión de los elementos y sus programaciones.

8.8.1. *Circuito principal*

Para la corroboración del sistema principal, se realiza un algoritmo en el IDE de Arduino (Figura 42, Figura 43, Figura 44 y Figura 45). La prueba consiste en detectar una tarjeta pasiva de chip NFC, realizar una confirmación de identidad. Al ingresar al sistema, el submódulo Wi-Fi ESP01, generará una señal por medio de internet, a la página web, indicándole que el parqueadero correspondiente, se encuentra ocupado hasta nuevo aviso; al mismo tiempo se genera un cambio visual en los indicadores del sistema LockBikeRack, en los cuales se apaga el led verde y se activa el led rojo. A continuación, el submódulo de sensores se encuentra esperando el pulso de activación para realizar el cierre de los pestillos. Al ser accionados ambos finales de carrera, el sistema se queda en un estado de espera en el submódulo lector de tarjetas RFID. El estado de espera únicamente será cambiado, cuando el submódulo RFID RC522 obtengan la misma información de la tarjeta pasiva con la que se generó el guardado.

Cuando el usuario correspondiente al guardado principal, realice el acercamiento al submódulo lector de tarjetas NFC, este cambiará su estado interno como en la página web, ya que este puesto quedará desocupado, el submódulo ESP8266 genera una petición de cambio de estado en la base de datos del aplicativo, los indicadores cambian de estado, al quedar libre el puesto en cuestión, nuevamente el led verde se activará y para finalizar el submódulo controladores de motor generará la apertura de los pestillos, para el desacople de la bicicleta con el sistema LockBikeRack.



```

PRUEBA! Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editor Programa Herramientas Ayuda

PRUEBA!
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN 9 //Pin 9 para el reset del RC522
#define SS_PIN 10 //Pin 10 para el SS (SDA) del RC522
MFRC522 MyLectorRF(SS_PIN, RST_PIN); //Creamos el objeto para el RC522

int MID = 2; // Motor 1 giro derecha - Salida
int MII = 3; // Motor 1 giro izquierda - Salida
int M2D = 4; // Motor 2 giro derecha - Salida
int M2I = 5; // Motor 2 giro izquierda - Salida
int SMI = 6; // Sensor M1 - Entrada
int SM2 = 7; // Sensor M2 - Entrada
int Led = 8; // LED
bool s1 = false; // variable entrada/salida #1
bool s2 = false; // variable entrada/salida #2

void setup()
{
  Serial.begin(9600); //Iniciamos La comunicacion serial
  SPI.begin(); //Iniciamos el Bus SPI
  MyLectorRF.PCD_Init(); // Iniciamos el MFRC522

  pinMode(MID, OUTPUT); // sets the digital pin as output
  pinMode(MII, OUTPUT); // sets the digital pin as output
  pinMode(M2D, OUTPUT); // sets the digital pin as output
  pinMode(M2I, OUTPUT); // sets the digital pin as output
  pinMode(Led, OUTPUT); // sets the digital pin as output

  pinMode(SMI, INPUT_PULLUP); // sets the digital pin as output
  pinMode(SM2, INPUT_PULLUP); // sets the digital pin as output
  Serial.println("Control de acceso!");
}

unsigned long tiempo_actual = 0;
unsigned long tiempo_anterior = 0;
unsigned long delta_tiempo = 0;
int contador = 0;

String BufferID = "";
String BufferGuardadoID = "";

//Base de datos de usuarios
String Usuarios[] = {"913baa27","6D 5F 30 11","A9 E4 7B 9E","C0 D8 7E 25","B1 40 30 20","02 55 DA 1B","AC C

void loop()
{
  Serial.println("Entre revisar tarjeta");
  // Revisamos si hay nuevas tarjetas presentes
  if ( MyLectorRF.PICC_IsNewCardPresent() )
  {
    Serial.println("Lectura de tarjeta");
    //Seleccionamos una tarjeta
    if ( MyLectorRF.PICC_ReadCardSerial() )
    {
      Serial.println("Guardado de tarjeta");
      // Enviamos serialmente su UID
      BufferID = "";
      BufferGuardadoID = "";
      //Serial.print(F("Card UID:"));

      for (byte i = 0; i < MyLectorRF.uid.size; i++)
      {
        BufferID = BufferID + String(MyLectorRF.uid.uidByte[i], HEX);
      }

      //comparamos los UID para determinar si es uno de nuestros usuarios
      Serial.println("Determinar si esta la tarjeta en la DB");
      if (compararUID(BufferID,Usuarios))
      {
        Serial.println("Acceso concedido...");
        BufferGuardadoID = BufferID;
        Serial.println("Guardando UID: ");
        Serial.println(BufferGuardadoID);
      }
    }
  }
}

```

Figura 42. Programación final, variables del sistema – LockBikeRack
Fuente: Autor



```
PRUEBA1 Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda

PRUEBA1

//EnvioAplicativoOcupado();
//Mover motores-----
MoverMotorEntrada(s1,s2);
delay(5000);
Serial.println("Corroboracion UID salida");
CompararTarjetaEntrada(BufferID,BufferGuardadoID);

Serial.println("Mover motores salida");
delay(1000000);
MoverMotorSalida(s1,s2);
EnvioAplicativoLibre();
}
else
Serial.println("Acceso denegado...");

// Terminamos la lectura de la tarjeta tarjeta actual
MyLectorRF.PICC_HaltA();

}

}
else if (true)
{
}
}

}

//-----FUNCIONES-----
//Función para comparar dos vectores
boolean compararUID(String UID,String Usuarios[])
{
    Serial.println("UID registrado");
    Serial.println(UID);

    for (byte i = 0; i < MyLectorRF.uid.size; i++)
    {
        Serial.println("Usuario posicion: ");
        Serial.println(Usuarios[i]);

        if (Usuarios[i] == UID)
        {
            Serial.println("UID Encontrado");
            return(true);
        }
    }
    return(false);
}

//Función para comparar vector de entrada
boolean CompararTarjetaEntrada(String UID,String UIDGuardado)
{
    Serial.println("Entre comparar tarjeta");
    bool status1 = true;
    while(status1 == true)
    {
        if ( MyLectorRF.PICC_IsNewCardPresent() )
        {
            Serial.println("Leyendo tarjeta");
            //Seleccionamos una tarjeta
            if ( MyLectorRF.PICC_ReadCardSerial() )
            {
                Serial.println("guardando tarjeta...");
                // Enviamos serialmente su UID
                UID = "";
                //Serial.print(F("Card UID:"));

                for (byte i = 0; i < MyLectorRF.uid.size; i++)
                {
                    UID = UID + String(MyLectorRF.uid.uidByte[i], HEX);
                }
                Serial.println("BufferID: ");
                Serial.println(UID);
                Serial.println("comparando tarjeta...");
                Serial.println(UIDGuardado);
            }
        }
    }
}
```

9MHz, 40MHz, DOUT (compatible), 1MB (FS 64KB OTA~470KB), 2 nonop-sd-2.2.1+100 (190703), V2 Lower Memory, Disabled, None, Only Sketch, 115200 en COM5

Figura 43. Programación final, ajustes de variables – LockBikeRack
Fuente: Autor

```

PRUEBA1 Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda

PRUEBA1

Serial.println(UIDGuardado);

Serial.println("comparando tarjeta...");
//comparamos los UID para determinar si es uno de nuestros usuarios
if(UIDGuardado == UID)
{
    Serial.println("Acceso concedido -> UID Correcto");
    status1 = false;
}
else
{
    Serial.println("Acceso denegado -> UID Incorrecto");
}
// Terminamos la lectura de la tarjeta tarjeta actual
MyLectorRF.PICC_Ralta();
}

}

return(true);
}

//Función para enviar a aplicativo ocupado
boolean EnvioAplicativoOcupado()
{
    return(true);
}

//Función para enviar a aplicativo libre
boolean EnvioAplicativoLibre()
{
    return(true);
}

//Función mover motores entrada
boolean MoverMotorEntrada(bool sensor1,bool sensor2)
{
    while(sensor1==false || sensor2==false) //Control motores de entrada
    {
        Serial.println("Entre WHILE Entrada\n");
        if (digitalRead(Sm1) == HIGH && digitalRead(Sm2) == LOW && sensor1 == false)
        {
            Serial.println("Mover motor 1 - Entrada");
            digitalWrite(M1I, HIGH); // sets the M1 Right on
            digitalWrite(M1D, LOW); // sets the M1 left off
            delay(260); // waits for a second
            digitalWrite(M1I, LOW); // sets the M1 Right off
            sensor1 = true;
        }
        else if (digitalRead(Sm2) == HIGH && digitalRead(Sm1) == LOW && sensor2 == false)
        {
            Serial.println("Mover motor 2 - Entrada");
            digitalWrite(M2I, HIGH); // sets the M2 left on
            digitalWrite(M2D, LOW); // sets the M2 Right off
            delay(260); // waits for a second
            digitalWrite(M2I, LOW); // sets the M2 left off
            sensor2 = true;
        }
        else
        {
            digitalWrite(M1I, LOW); // sets the M1 off
            digitalWrite(M1D, LOW); // sets the M1 off
            digitalWrite(M2I, LOW); // sets the M2 off
            digitalWrite(M2D, LOW); // sets the M2 off
        }
    }
    return (sensor1,sensor2);
}

//Función mover motores salida
boolean MoverMotorSalida(bool sensor1,bool sensor2)
{
    Serial.println("Entre WHILE Salida\n");
    if (sensor1 == true && sensor2 == true)
    {

```

8MHz, 40MHz, DOUT (compatible) 1MB (FS:54KB OTA-470KB), 2 nonos-usb 2.2.1+100 (190703), v2 Lower Memory, Disabled, None, Only Sketch, 115200 en COM5

Figura 44. Programación final - LockBikeRack
Fuente: Autor



Figura 45. Programación final, funciones – LockBikeRack
Fuente: Autor

8.9. Diseño de piezas del prototipo

Para el diseño estructural del prototipo LockBikeRack, se concibió un material que soportara los diferentes climas actuales. Por lo pensado, se implementa el diseño de los siguientes prototipos ajustados a la posición de los circuitos electrónicos usados con anterioridad. Se inicia un plan de medidas a el cicloparqueadero de la universidad Santo Tomás, para realizar un modelo adaptable a la estructura metálica ya establecida.

Para obtener un diseño concreto y profesional, se procede a buscar softwares de diseño estructural o piezas sólidas. Se emplean los softwares (CAD), que significa Diseño Asistido por

Computador, en los cuales hay varias ramas de este tipo de herramientas. Para el caso del sistema LockBikeRack, se inicia el proceso de realización en el programa Autodesk inventor, en el cual la generación de figuras solidas. Los programas de la familia Autodesk, son principalmente usados en el diseño industrial en escala productiva. Con las referencias obtenidas, se procedió a crear nuestras propias piezas del prototipo generando los planos correspondientes para envío y fabricación profesional.

Con los parámetros medidos, se continúa con la proyección de la estructura general, la cual, se encontrará en la parte externa, recibiendo así, todas las impurezas o maltrato que se le pueda generar con el paso de los diferentes climas actuales.

8.9.1. *Diseño estructura general*

El diseño estructural de la pieza general o la capa que recubrirá los elementos electrónicos se inició generando un cubo simétrico de medidas 30cm de ancho por 30cm de largo (Figura 46), el cual permita la inserción de los elementos de manera práctica. Para la creación del cubo exterior, se concibió en el programa Autodesk inventor 2023, generar un nuevo programa y continuando con la selección de un nuevo boceto. Se inicia seleccionando el plano horizontal, correspondiente a los vectores (x, y). Se opta por generar pista por pista de 30 mm dándole valores en milímetros ya que es la medida estándar del programa; se continúa generando un boceto en 3D sólido y realizar un vaciado de este, ya que dentro de la estructura debe contemplarse los circuitos electrónicos.

Para los recortes que contiene la estructura general demostrados en la Figura 46, se inicia el boceto en la parte superior y se selecciona el lado en el cual se quiere aplicar la edición; para el caso de la estructura del sistema LockBikeRack debe generarse, los recortes de las entradas al pestillo, que constan de una medida de 5 mm de ancho a 250 mm de alto referido a las salidas de las guayas plegables. Los recortes de la parte inferior del sistema son incluidos como los sujetadores o el punto de anclaje de la caja a las barricadas del cicloparqueadero, con unas medidas de 5 mm de alto y 125 mm de ancho. Por último, se genera el recuadro de 40mm x 60 mm del submódulo RFID RC522 en la parte inferior derecha de la tapa principal de la estructura. Finalmente se crean recortes redondos para los indicadores de los estados del prototipo, los cuales contienen un diámetro de 5cm, posicionados en la parte superior de la caja uno a la parte centro izquierda y el siguiente recorte en la sección centro derecha de la capa superior.



Figura 46. Programación final, funciones – LockBikeRack
Fuente: Autor

Para la estructura exterior se solicitó lamina de acero inoxidable, en el cual, no va a permitir el ingreso de ambientes dañinos para el circuito. Teniendo así, la certeza de una estructura confiable, que protegerá los componentes electrónicos situados dentro de esta. La estructura metálica en lamina recrea la idea de un sistema seguro.

8.9.2. *Diseño divisiones*

Las Divisiones que generan una estructuración de pisos o secciones, dentro de la estructura metálica, se aplicaron primordialmente, para la extracción de los circuitos electrónicos y su mantenimiento preventivo o cualquier tipo de correctivo. El diseño de la placa de división, contiene unas medidas de 96 mm de ancho y 196 de largo, con 2 recortes con funcionalidad práctica, el primer corte se establece en la parte inferior de la división, para realizar el paso del cableado de los circuitos electrónicos, en el segundo recorte cuenta con unas medidas de 5 mm de ancho y 28 mm de alto, se encuentra ubicado a 97 mm del tope inferior al comienzo del recorte, con la finalidad de generar un tope en el momento de ingreso de la guaya plegable como se puede observar en la Figura 47.

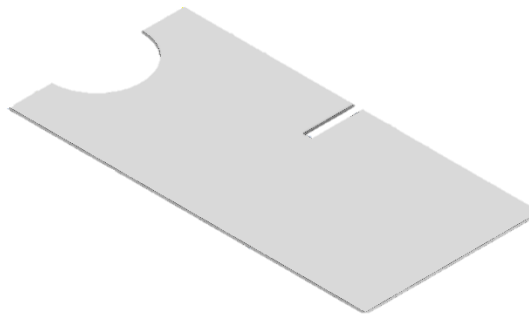


Figura 47. Simulación división pisos – LockBikeRack
Fuente: Autor

Como se puede ver en la siguiente Figura 48, se visualiza el posicionamiento de las divisiones dentro de la estructura general y su organización.



Figura 48. Simulación división pisos estructura general – LockBikeRack
Fuente: Autor

Para el implemento de los circuitos electrónicos, se realizó un esquema a tamaño real, de una posible organización de los circuitos electrónicos (Figura 49). Ya que este cuenta con los componentes electrónicos seleccionados en el sistema LockBikeRack.



Figura 49. Simulación división pisos circuitos electrónicos – LockBikeRack
Fuente: Autor

8.9.3. *Diseño de la estructura soporte del pestillo*

El soporte de pestillo trata de una pieza modelada en 3D mediante el software Fusion 360. La pieza se encarga de soportar tanto el motorreductor como el pestillo, además de hacer de carril para la cremallera de dientes que recibe el movimiento angular del piñón y lo convierte en movimiento rectilíneo.

Su diseño implicó la toma de medidas de aquellas partes que debía soportar, por lo cual el diseño cuenta con paredes inclinadas que rodean y sujetan indirectamente a la base del pestillo y vías que rodean el recorrido que hace la cremallera; el soporte de motorreductor cuenta con un par de agujeros, en donde se atornilla para asegurar el motor.

Finalmente, el diseño consta de una base de 95 mm x 50 mm, la cual es crucial para aportar resistencia estructural, y a la par, enganchar la pieza a los niveles de división de pisos internos del chasis estructural del dispositivo.

En la Figura 50 es posible visualizar el modelo diseñado para su posterior impresión en filamento PLA, contemplando espacios de 0.5 mm con respecto a los puntos de contacto con el pestillo y cremallera, buscando que no existan una fricción para el movimiento del mecanismo.

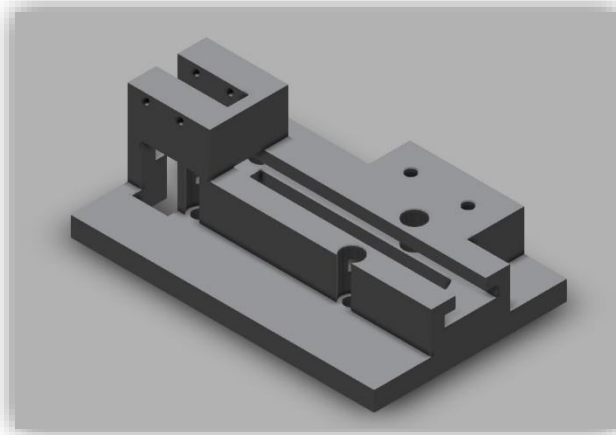


Figura 50. Simulación de soporte pestillo – LockBikeRack
Fuente: Autor

Gracias a los softwares de Autodesk, los cuales permitieron la creación de estos prototipos estructurales, los cuales, generan una consistencia solida a la finalización del sistema LockBikeRack. La organización de los planos y el entendimiento estructuras se llevó a cabo con los programas de Autodesk, para el envío a producción del prototipo. Obteniendo los valores de superficie y de anchura de el cicloparqueadero se pudo disponer de varias mejoras iimplementadas a la estructura principal.

8.10. Diseño de aplicación y servicios web

El funcionamiento del sistema LockBikeRack implica mantener una comunicación vía internet entre el dispositivo y el servidor web, en donde se realicen diversas consultas que verifiquen la identidad del usuario y brinden acceso a los candados flexibles para asegurar su bicicleta a los soportes del cicloparqueadero. También se plantea la posibilidad de que cualquier usuario conozca el estado del cicloparqueadero visualizando los puestos disponibles al ingresar en la plataforma desde un navegador web.

Para la ejecución de las tareas descritas anteriormente se modela el esquema de la Figura 51, en donde se identifican dos secciones, una correspondiente a la comunicación entre el dispositivo de seguridad y la base de datos, por medio de una capa de servicios; y otra sección destinada a mostrar información del cicloparqueadero a través de una capa de aplicación web.

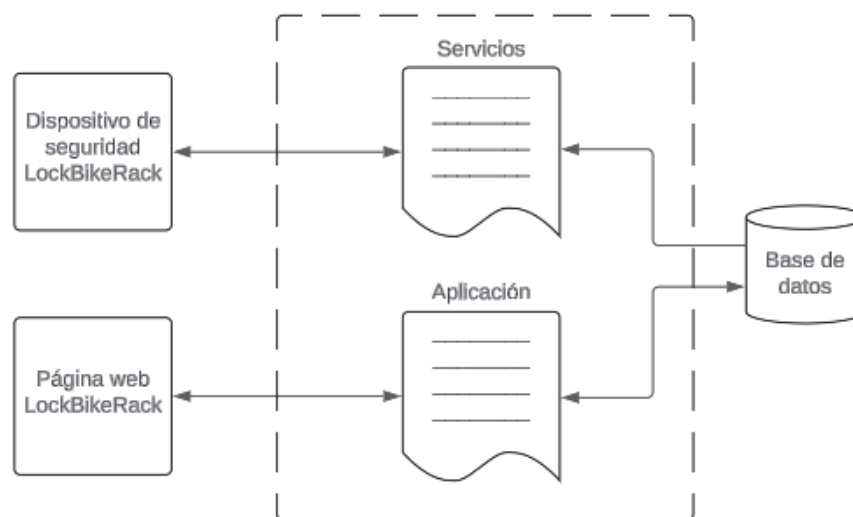


Figura 51. Esquema de servicios y aplicación web - LockBikeRack
Fuente: Autor

8.10.1. Servicios

La capa de servicios consta de algoritmos de programación que permiten recibir la información obtenida por el dispositivo y contrastarla con registros de la base de datos, de tal manera que permitan tomar decisiones con respecto al comportamiento del dispositivo de seguridad.

La Figura 52 describe el algoritmo que lleva a cabo la verificación del usuario para su ingreso al sistema, ésta comprende la búsqueda del usuario en la base de datos mediante el código de identificación UID de su tarjeta, de ser correctos los datos cambiará el estado del puesto de cicloparqueadero a "Ocupado" y registrará la fecha y hora de ingreso en la tabla de estación y guardado.

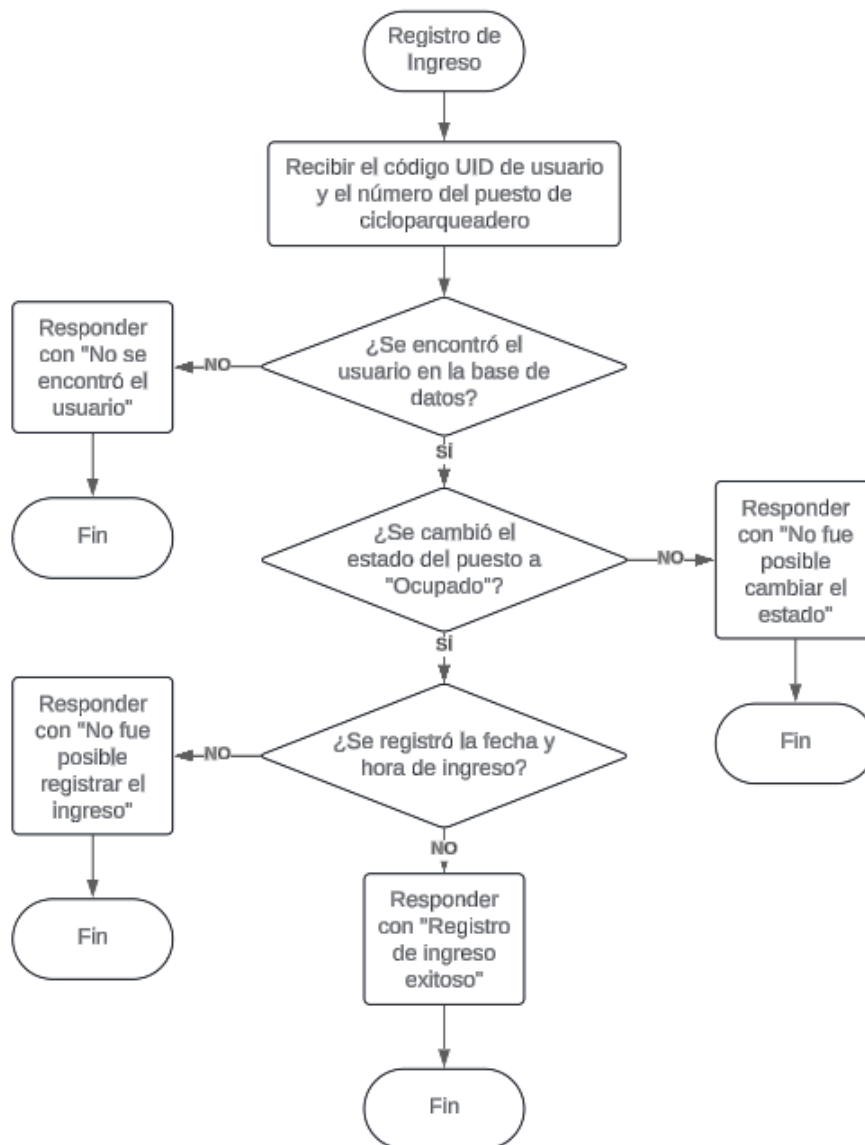


Figura 52. Diagrama de flujo del algoritmo de registro de ingreso - LockBikeRack
Fuente: Autor

A continuación, se muestra el algoritmo del registro de salida mediante el diagrama de flujo de la Figura 53, en donde el programa recibe el número del puesto al que corresponde y se encarga de cambiar su estado a “Disponible” además de completar el registro de salida, incluyendo su fecha y hora en la tabla estación y guardado.

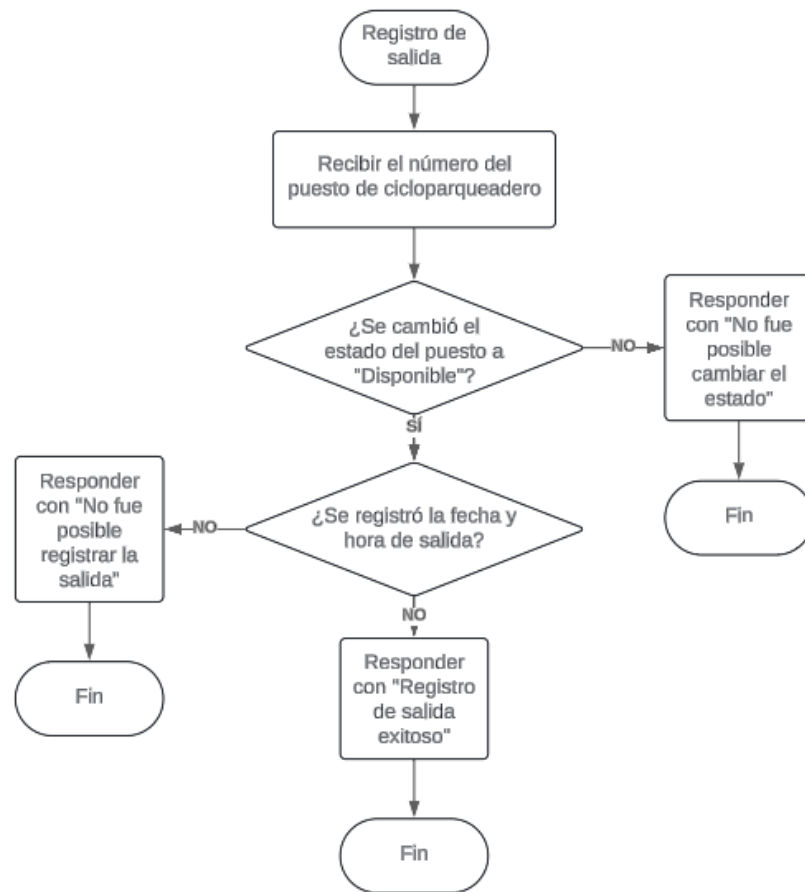


Figura 53. Diagrama de flujo del algoritmo de registro de salida - LockBikeRack
Fuente: Autor

Finalmente es incluido el algoritmo de la Figura 54, al contemplar la posibilidad de que se reinicie el dispositivo aun estando en uso y exista una bicicleta ocupando el lugar en la estación. Para ello el sistema siempre realiza una consulta al encenderse, determinando si el número del puesto asociado al dispositivo se encuentra ocupado y de ser así, toma directamente el código de identificación encontrado para que la próxima lectura de tarjeta corresponda a la salida del usuario.

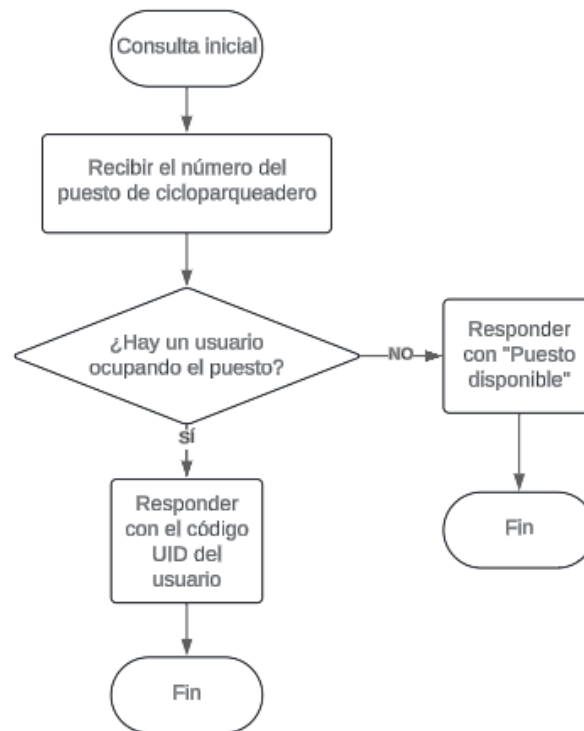


Figura 54. Diagrama de flujo del algoritmo de consulta inicial - LockBikeRack
Fuente: Autor

8.10.2. *Aplicación*

La capa de aplicación corresponde al conjunto de vistas al que el usuario accede a través de la página web y al componente funcional que dicta el comportamiento de la página en cada una de ellas. La página está compuesta por un formulario de inicio de sesión, un formulario de registro y una página principal.

El formulario de inicio de sesión que se muestra en la Figura 55 incluye los campos “Correo electrónico” y “Contraseña”, con los cuales el usuario podrá acceder a la página principal de contar con una cuenta registrada en la base de datos del sistema.

LOCKBIKERACK

Ingresa para consultar bici-parqueaderos.

Correo electrónico

Contraseña

Iniciar sesión

¿No tienes una cuenta? [Regístrate](#)

Figura 55. Formulario de inicio de sesión - LockBikeRack
Fuente: Autor

De no poseer una cuenta, el usuario tendrá la opción de registrarse, haciendo clic en el texto “Regístrate”, donde será redirigido al formulario de la Figura 56, el cual deberá completar con datos personales tales como su nombre y apellidos, su tipo y número de documento, código institucional, correo electrónico, programa académico y contraseña. Luego de completar este proceso, el usuario podrá ingresar a la página directamente.

REGÍSTRATE

Nombre

Apellido

Documento de identificación

☒ T.I. ☐ C.C. ☐ C.E. ☐ PAS.

Código

Correo electrónico

Programa

Contraseña

Repetir contraseña

Registrar

¿Ya tienes una cuenta? [Iniciar Sesión](#)

Figura 56. Formulario de registro - LockBikeRack
Fuente: Autor

Al acceder a la página principal por medio de una cuenta registrada, la página muestra un mensaje de bienvenida seguido de una tabla, la cual indica el estado de cada uno de los puestos del cicloparqueadero, tal como se observa en la Figura 57.

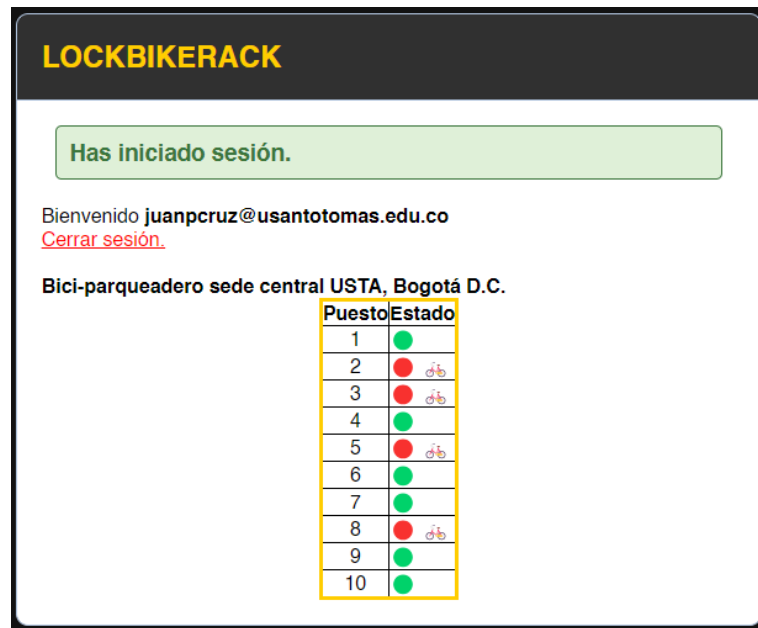


Figura 57. Página principal y estado del cicloparqueadero - LockBikeRack
Fuente: Autor

Para conseguir el comportamiento esperado de cada una de las vistas de la página web, se propone el algoritmo de la Figura 58, donde se indican las distintas acciones a realizar según la petición que se reciba desde los formularios. Siendo únicamente una comparación que valide los datos de usuario para el caso de inicio de sesión, una incorporación de un nuevo usuario en la base de datos para el caso de registro o una visualización de los estados de cada uno de los puestos del cicloparqueadero para la página principal.

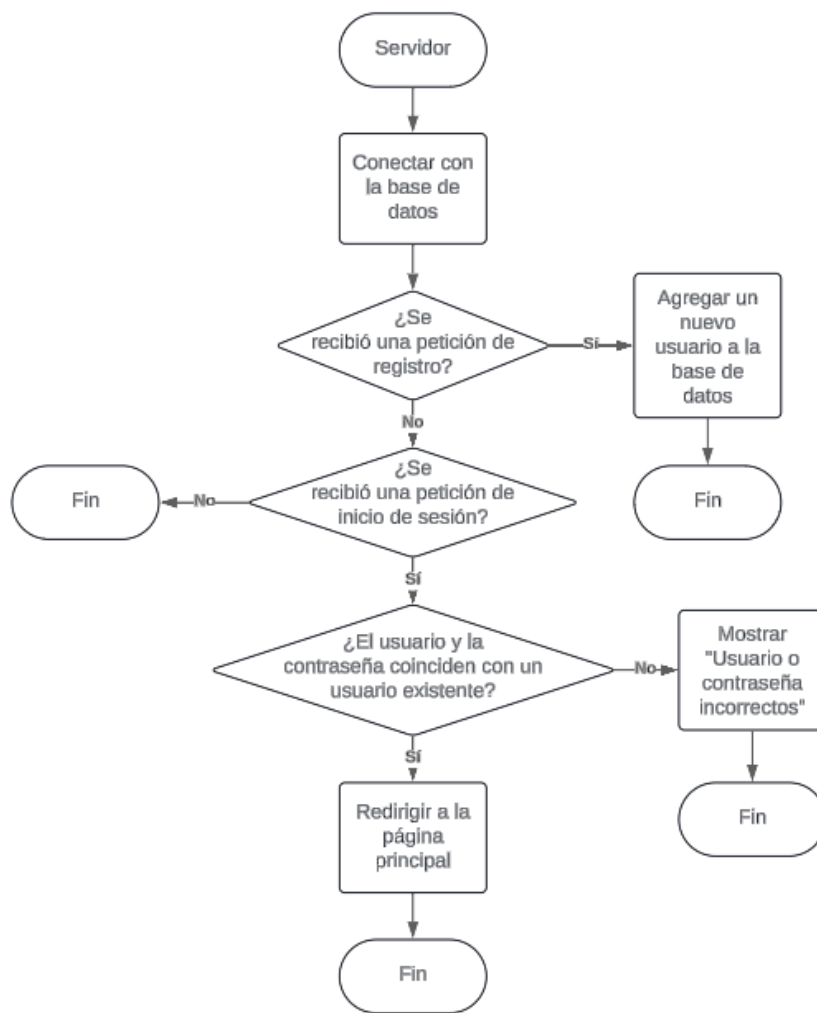


Figura 58. Diagrama de flujo del algoritmo del servidor - LockBikeRack
Fuente: Autor

Capítulo 9

Resultados

Mediante este trabajo se propone la implementación del sistema de seguridad LockBikeRack que permite automatizar el acceso y registro de usuarios en un cicloparqueadero, para ello, se construyó un prototipo con correas plegables de activación electrónica a través de la lectura de etiquetas NFC. El sistema cuenta con una página web, la cual muestra el estado de disponibilidad a tiempo real de los puestos de parqueo, al estar vinculada a una base de datos, permite recibir la información directa de cada dispositivo mediante la comunicación por conexión Wi-Fi.

A continuación, se enseñan los resultados divididos en los siguientes componentes:

- Circuitos electrónicos.
- Piezas estructurales.
- Base de datos.
- Página web.
- Montaje final del prototipo.

9.1. Circuitos electrónicos

Se implementó el diseño del circuito principal, el cual permite redireccionar las conexiones con la placa Arduino, el módulo RFID-RC522, el módulo Wi-Fi ESP8622 (ESP-01), los motorreductores del mecanismo de pestillos, los fines de carrera, leds indicadores y fuente de alimentación de 12 V. El circuito principal tiene unas dimensiones de 9 cm x 6 cm.

En la Figura 59 se visualiza la cara superior del circuito principal indicando los submódulos implementados y sus respectivos pines de conexión; la cara inferior del mismo se muestra en la Figura 60, en la cual se observan las pistas del circuito junto con el plano de tierra y las soldaduras para los elementos electrónicos.

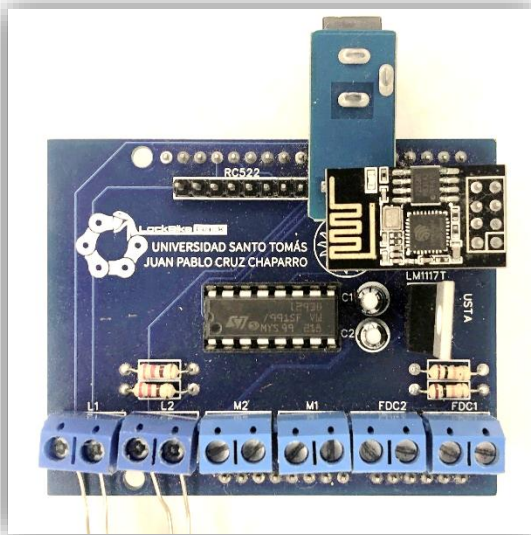


Figura 59. PCB del circuito principal superior –
LockBikeRack
Fuente: Autor

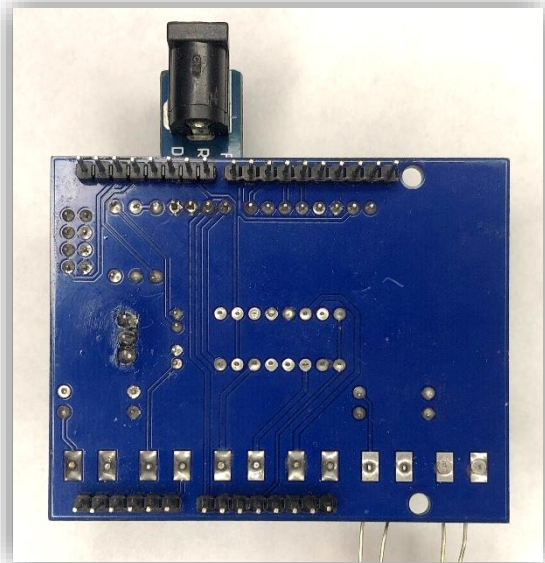


Figura 60. PCB del circuito principal inferior –
LockBikeRack
Fuente: Autor

9.2. Piezas estructurales

En cuanto a la construcción física del sistema LockBikeRack, se implementó una estructura compuesta por dos tipos de fabricación.

9.2.1. *Soportes del mecanismo de los pestillos*

Consta de piezas fabricadas mediante impresión 3D de filamento de ácido poli láctico (PLA), de dimensiones 95 mm x 50 mm x 18 mm, brindando soporte a los pestillos metálicos, a los engranajes de cremallera y al motorreductor encargado del movimiento del pestillo.

La Figura 61 muestra una imagen en perspectiva de uno de los soportes de mecanismo de pestillo posterior a su impresión 3D, mientras que en la Figura 62 se ve una imagen de este, esta vez en plano cenital; en ambas imágenes se perciben los agujeros correspondientes a los tornillos que sujetan la pieza tanto al motorreductor, como a la base de cada nivel de la estructura.



Figura 61. Impresión 3D soporte de pestillo
perspectiva – LockBikeRack
Fuente: Autor

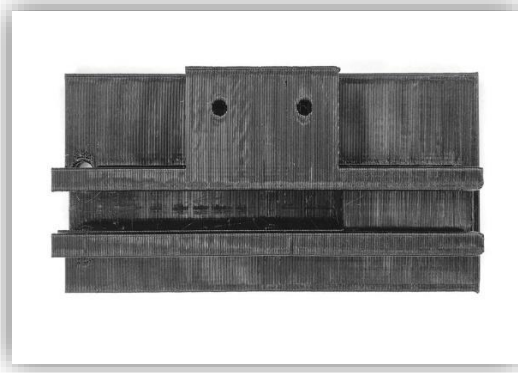


Figura 62. Impresión 3D soporte de pestillo superior
– LockBikeRack
Fuente: Autor

Ambas piezas se realizaron con una altura de capa de 0.24 mm, el cual equivale a las líneas horizontales para cada capa de impresión, además cuenta con una pared de 1.2 mm de ancho y un relleno al 50%; permitiendo que el diseño adquiriera la resistencia mecánica suficiente para soportar el movimiento del motor y la cremallera.

9.2.2. *Cubierta estructural*

El diseño implementado cuenta con una cubierta física con divisiones internas separadas en 3 niveles. Esta cubierta posee unas dimensiones de 33 cm x 30 cm x 30 cm, y cuenta con 2 divisiones internas separadas a 10 cm entre ellas y con respecto a la parte exterior.

En la Figura 63 se ve una imagen en perspectiva de la cubierta estructural; en la Figura 64 se ve la cubierta desde un plano frontal. Se perciben los ganchos pertenecientes a las correas de sujeción

al costado derecho, dos indicadores en la parte superior y algunas aperturas en la cara frontal correspondientes a los cierres de los pestillos y al espacio para el lector NFC.



Figura 63. Cubierta estructural perspectiva –
LockBikeRack
Fuente: Autor



Figura 64. Cubierta estructural frontal –
LockBikeRack
Fuente: Autor

9.3. Base de datos

Con la finalidad de brindar un soporte a los datos obtenidos en el registro de acceso a la plataforma de LockBikeRack, se implementó una base de datos sobre el software de gestión de bases de datos (DBMS) Microsoft SQL Server 2018. La estructura de la base de datos se ilustra en el diagrama de la Figura 65, cuyas tablas son visibles en los ficheros de la Figura 66.

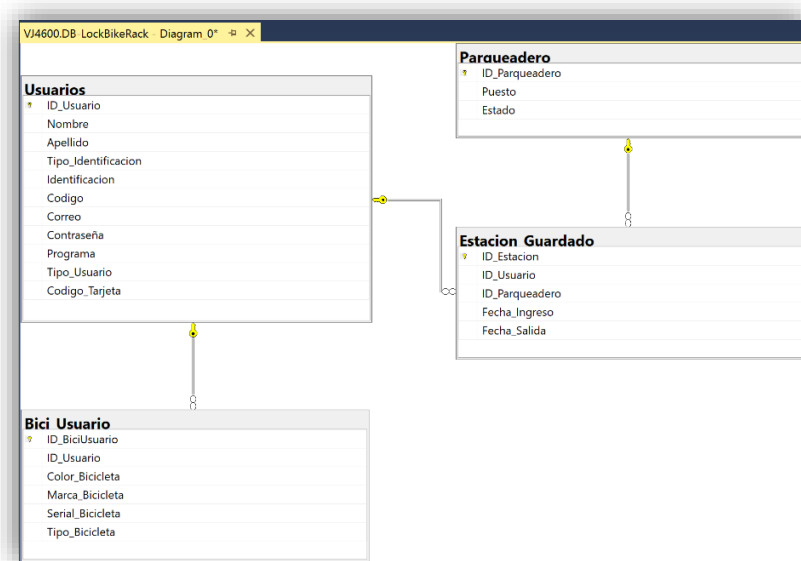


Figura 65. Diagrama de base de datos – LockBikeRack
Fuente: Autor

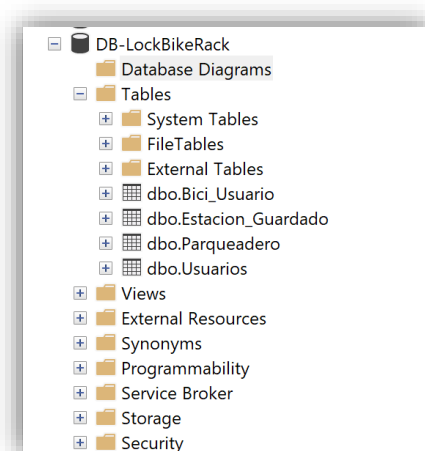


Figura 66. Ficheros de base de datos – LockBikeRack
Fuente: Autor

Las capturas que se enseñan en la Figura 67, la Figura 68, la Figura 69 y la Figura 70; permiten observar los campos y tipos de variables implicados en cada una de las tablas de la base de datos establecida. También se identifican las llaves primarias de cada tabla, las cuales incluyen el ícono de una llave a la izquierda del nombre del campo.

Column Name	Data Type
ID_BiciUsuario	int
ID_Usuario	int
Color_Bicicleta	varchar(50)
Marca_Bicicleta	varchar(50)
Serial_Bicicleta	varchar(50)
Tipo_Bicicleta	varchar(50)

Figura 67. Tabla Bici_Usuario base de datos – LockBikeRack
Fuente: Autor

Column Name	Data Type
ID_Estacion	int
ID_Usuario	int
ID_Parqueadero	int
Fecha_Ingreso	datetime
Fecha_Salida	datetime

Figura 68. Tabla Estacion_Guardado base de datos – LockBikeRack
Fuente: Autor

Column Name	Data Type
ID_Parqueadero	int
Puesto	int
Estado	int

Figura 69. Tabla Parqueadero base de datos – LockBikeRack
Fuente: Autor

Column Name	Data Type
ID_Usuario	int
Nombre	varchar(50)
Apellido	varchar(50)
Tipo_Identificacion	varchar(50)
Identificacion	int
Codigo	int
Correo	varchar(50)
Contraseña	varchar(50)
Programa	varchar(50)
Tipo_Usuario	varchar(50)
Codigo_Tarjeta	varchar(50)

Figura 70. Tabla Usuarios base de datos – LockBikeRack
Fuente: Autor

Para validar el funcionamiento esperado de la base de datos, se han ingresado algunos campos a modo de ejemplo. Al realizar una consulta por el contenido de las tablas, se obtiene como resultado la captura visible en la Figura 71.

```

SELECT * FROM Bici_usuario;
SELECT * FROM Parquedero;
SELECT * FROM Estacion_Guardado;
SELECT * FROM Usuarios;

```

ID_BiciUsuario	ID_Usuario	Color_Bicicleta	Marca_Bicicleta	Serial_Bicicleta	Tipo_Bicicleta
1	1	Blanco	GW	BYOCN203	Pista
2	2	Negro	GW	E8BKX7X1	Pista

ID_Parquedero	Puesto	Estado
1	1	0
2	2	0
3	3	0
4	4	0
5	5	0
6	6	0
7	7	0
8	8	0
9	9	0
10	10	0

ID_Estacion	ID_Usuario	ID_Parquedero	Fecha_Ingreso	Fecha_Salida
1	1	1	2022-11-16 06:04:05	2022-11-16 06:05:15
2	1	4	2022-11-16 09:01:40	2022-11-16 09:01:55
3	4	4	2022-11-16 09:03:58	2022-11-16 09:07:34

ID_Usuario	Nombre	Apellido	Tipo_Identificacion	Identificacion	Codigo	Correo	Contraseña	Programa	Tipo_Usuario	Codigo_Tarjeta
1	Juan Pablo	Cruz Chaparro	C.C.	1022430587	2200334	juanpacruz@usantotomas.edu.co	/rueUjPgY	Ingeniería Electrónica	Administrador	00128018
2	Nombre	Apellido	C.C.	1014304449	2200832	Usuario@usantotomas.edu.co	qwFuwx0r0i	Ingeniería Electrónica	Usuario	9138AA27
3	Nombre	Apellido	C.C.	1031430907	2200749	UsuarioA@usantotomas.edu.co	TfOo83Ac2C	Ingeniería Electrónica	Usuario	B1403020
4	Nombre	Apellido	C.C.	1027750472	2200604	UsuarioB@usantotomas.edu.co	PWwppwhr2	Ingeniería Electrónica	Usuario	A8E47B9D
5	Nombre	Apellido	C.C.	1032422286	2200791	UsuarioC@usantotomas.edu.co	Pigm1H GvL	Ingeniería Electrónica	Usuario	C0DB7E25

Figura 71. Consulta de contenido de tablas, base de datos – LockBikeRack
Fuente: Autor

9.4. Página web

En el programa de Microsoft Visual Studio .NET se implementa una página codificada en lenguaje HTML5, la cual cumple la función de aplicativo desde el cual mostrar a tiempo real el estado del cicloparqueadero.

La página web se encuentra incorporada en la dirección local, en donde se encuentran iniciados tanto el servidor web, como la base de datos SQL, en donde se realizan las consultas y el ingreso de nuevos registros al ingresar como un nuevo usuario.

La Figura 72 permite visualizar el aspecto de ingreso de la página principal, contando con un formulario de inicio de sesión y registro al sitio web del sistema LockBikeRack.

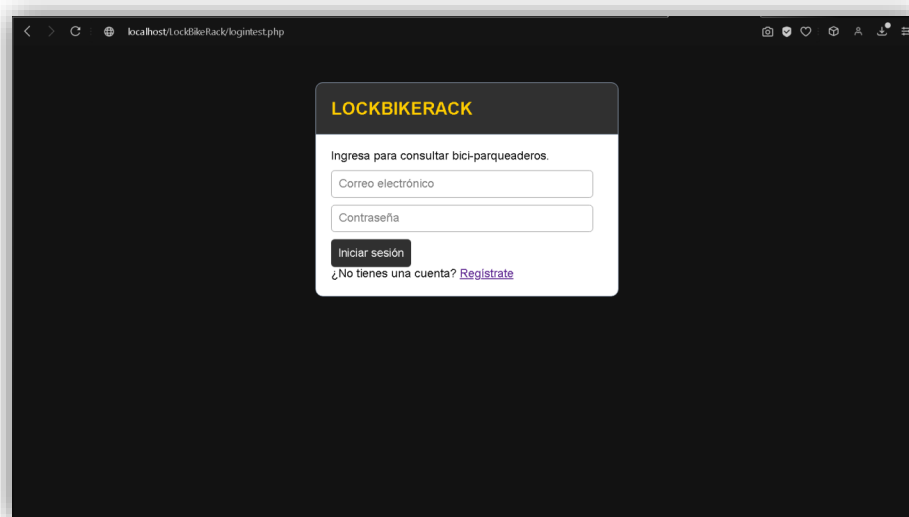


Figura 72. Ingreso, página web – LockBikeRack
Fuente: Autor

Al hacer clic en “Regístrate” el servidor redireccionará al usuario hacia el formulario que se observa en la Figura 73, esta vez permitiendo registrar un nuevo usuario.

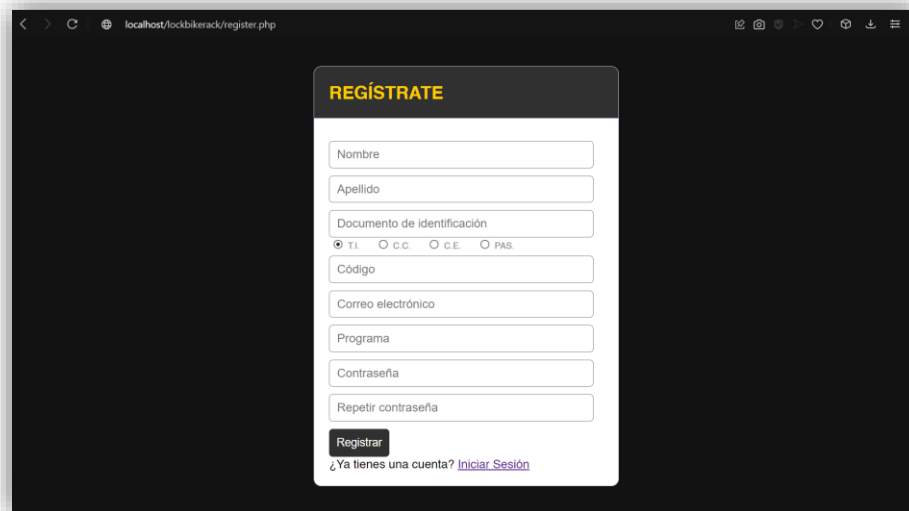


Figura 73. Registro, página web – LockBikeRack
Fuente: Autor

La página principal a la que se tendrá acceso iniciando sesión con una cuenta existente corresponde a un apartado enfocado en mostrar el estado de cada uno de los puestos del cicloparqueadero como se ve en la Figura 74, indicando un color verde para las posiciones disponibles y un color rojo para aquellas que se encuentren ocupadas.

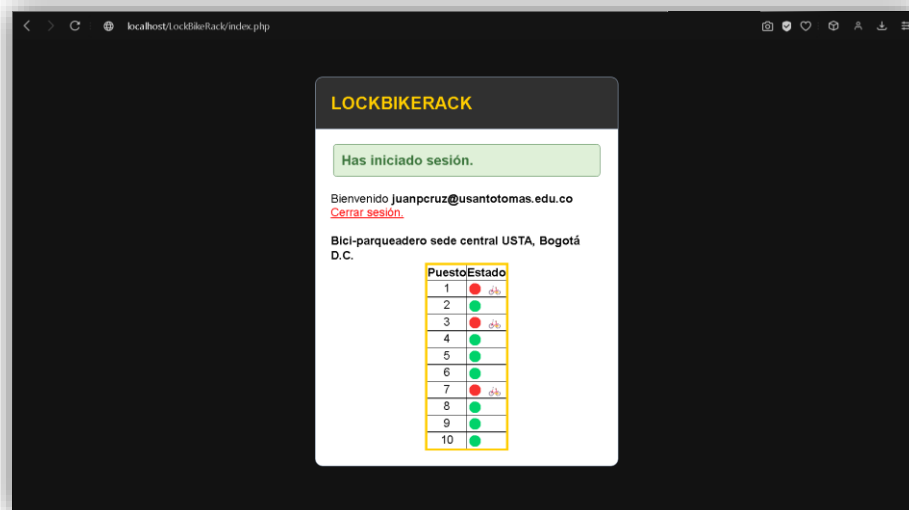


Figura 74. Estado del cicloparqueadero, página web – LockBikeRack
Fuente: Autor

9.5. Montaje final del prototipo

Al momento de finalizar la fabricación del prototipo, se montó el dispositivo en uno de los puestos del cicloparqueadero, en donde posteriormente se validó el funcionamiento del sistema tras asegurar una bicicleta mediante las correas del sistema, tal como se evidencia en la Figura 75.



Figura 75. Montaje final – LockBikeRack
Fuente: Autor

La Figura 76 corresponde a una captura de pantalla del cómo se mostraría el estado de algunos puestos disponibles mediante el sistema de seguridad de LockBikeRack, donde se evidencia la disponibilidad de algunos lugares representados con el color verde; mientras que la Figura 77, indica cómo se visualizaría el estado del cicloparqueadero con todos sus puestos ocupados representados mediante el color rojo y es símbolo de una bicicleta.

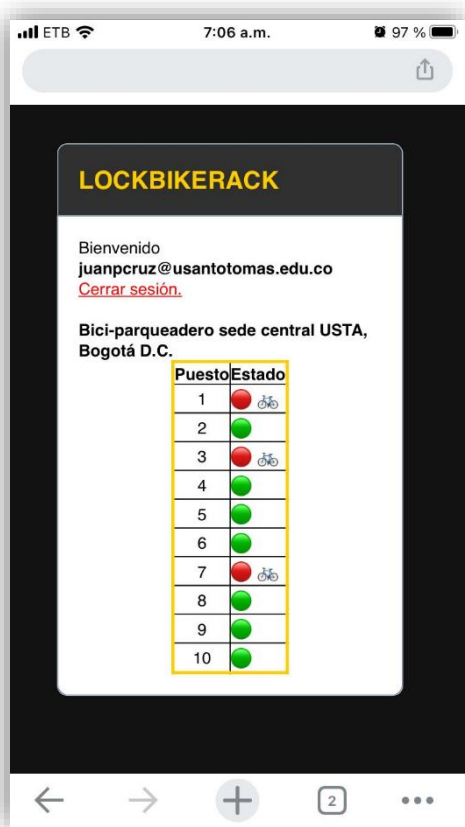


Figura 76. Estado disponible, página web – LockBikeRack
Fuente: Autor

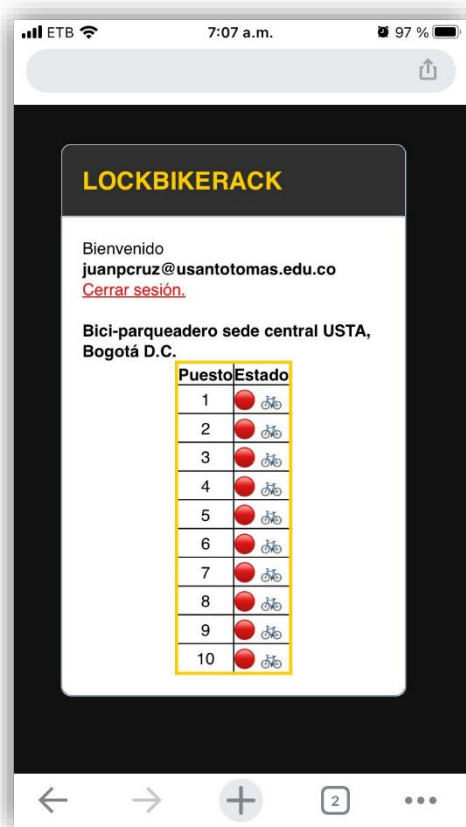


Figura 77. Estado ocupado, página web – LockBikeRack
Fuente: Autor

Capítulo 10

Conclusiones

Los objetivos planteados para este proyecto han sido cumplidos y han permitido determinar una serie de contribuciones y apreciaciones concluyentes.

A su vez, el análisis previo de compatibilidad entre los componentes y módulos electrónicos ha simplificado en gran medida el desarrollo práctico del proyecto, esto gracias a los procesos de investigación implicados en generar las bases teóricas sustanciales que han hallado las herramientas adecuadas para enfatizar el trabajo en avances productivos.

La implementación de una Base de datos como estructura de gestión de acceso al sistema LockBikeRack, en paralelo permitió garantizar el cumplimiento del aplicativo web propuesto, que, vinculado a ella, muestra de manera esperada la disponibilidad de cupos en el cicloparqueadero.

La información almacenada en los registros relaciona al usuario con su bicicleta y su información personal, brindándole una mayor confianza al existir un enlace directo entre los campos de ID de cada una de las tablas del esquema de base de datos diseñado.

La labor de aprendizaje acerca de distintos softwares enfocados en el diseño asistido por ordenador (CAD), específicamente en aquellos de la firma Autodesk, ha brindado un mayor entendimiento a nivel espacial, permitiendo llevar a cabo optimizaciones puntuales de diseño y aportando pulidez al producto, para hacer de este una solución más competente con las expectativas de calidad actuales.

Capítulo 11

Trabajos Futuros

El prototipo LockBikeRack se encuentra diseñado principalmente para parqueaderos que cuenten con una estructura base elevada, lo que quiere decir que, la llanta delantera se encuentre en la parte superior de la base, otorgando una posición vertical de la bicicleta. Partiendo del diseño realizado no se garantiza que el sistema LockBikeRack se adapte cómodamente a cualquier estructura ya implementada en cualquier cicloparqueadero.

Se realiza una proyección con el sistema LockBikeRack a garantizar la seguridad y comodidad en la acción de parquear. Por lo mencionado, se contemplan proyecciones a futuro, con el fin de garantizar todos los aspectos de seguridad y comodidad. Para este caso se proyectan varias mejoras al sistema como:

- 1. Cámaras de vigilancia:** Incluyendo cámaras al sistema, se puede garantizar una supervisión constante por parte de un personal de seguridad, el cual no debe mantenerse posicionado en el parqueadero para monitorear las acciones de los biciusuarios u posibles robos.
- 2. Doble autenticación:** Generar un bloque cerrado al cicloparqueadero. Incorporando una entrada con autenticación al cicloparqueadero, lo que garantiza que solo personal autorizado o con tarjetas de acceso, puedan alojar sus bicicletas dentro del sistema.
- 3. Sistema entrega de tarjetas:** Diseñar un sistema de entrega de tarjetas, asemejándose a un parqueadero de carros o motos, generando un tiquete de ingreso y que este sea aceptado en los lectores NFC incorporados ya en el sistema.
- 4. Alertas en el aplicativo:** Contando que el aplicativo web no sea únicamente un visualizador de cupos, sino también, contando con alertas a un usuario administrador que pueda detectar un abuso o posible hurto a la bicicleta incorporada en el sistema LockBikeRack.
- 5. Sensores de seguridad:** Incluir sensores de contacto, sensores infrarrojos o tipos de sensores que contribuyan al monitoreo de la bicicleta incorporada en el sistema. Detectando un posible hurto de la bicicleta y garantizando así la seguridad de esta.
- 6. Paneles solares:** Contando con las nuevas tecnologías, se puede adaptar paneles solares como alimentación del sistema LockBikeRack, ayudando así al medio ambiente, utilizando energías renovables. Mayormente los cicloparqueaderos actuales, cuentan con un posicionamiento al aire, lo que haría una facilidad a la hora de anexar energías renovables al sistema.
- 7. Indicadores de seguridad:** Actualmente el sistema LockBikeRack, cuenta con indicadores de cupo, lo que referencia a la disponibilidad del parqueadero; sin embargo, pueden adaptarse más indicadores al sistema o trabajar con los indicadores incorporados, para mejorar el

posicionamiento de una bicicleta de un posible hurto, dándole un mayor alcance a un personal de seguridad a la hora de dirigirse a un cicloparqueadero y localizar el puesto que contiene la alerta.

Cabe resaltar, que las mejoras pensadas para el sistema pueden variar a lo largo del tiempo, ya que las nuevas tecnologías siguen avanzando y adaptándose a el sistema con facilidad.

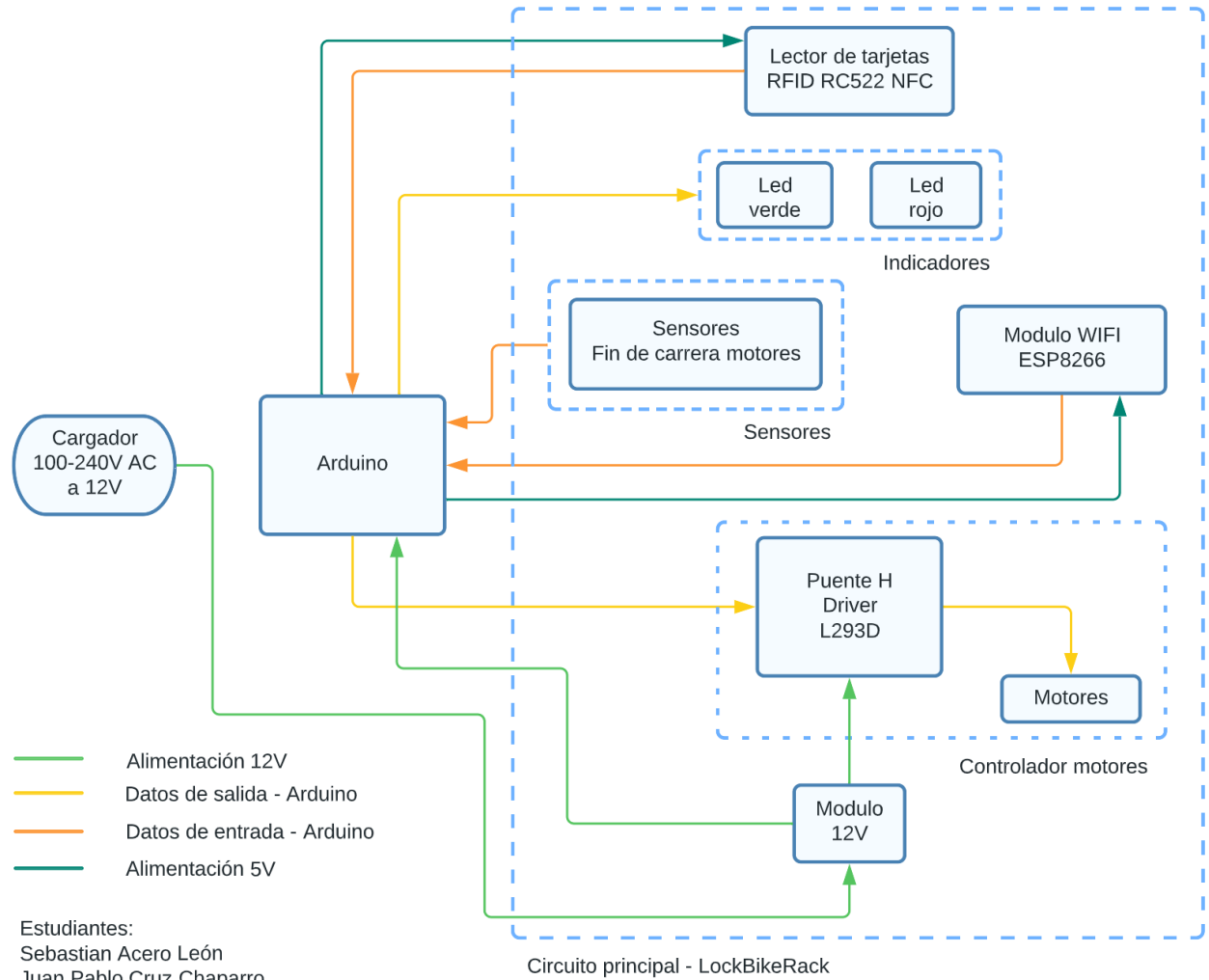
REFERENCIAS

- [1] Redacción Bogotá, «El espectador,» Grupo Valorem, 10 Noviembre 2021. [En línea]. Available: <https://www.elespectador.com/bogota/en-lo-corrido-del-ano-mas-de-8200-ciclas-han-sido-hurtadas-en-bogota/>. [Último acceso: 14 Junio 2022].
- [2] Secretaría distrital de movilidad, «Registro bici Bogotá,» s.f. [En línea]. Available: <https://registrobicibogota.movilidadbogota.gov.co/#!/>. [Último acceso: 14 Junio 2022].
- [3] Departamento Administrativo Nacional de Estadística, «DANE,» Gustavo Verbel, 28 Marzo 2023. [En línea]. Available: https://www.dane.gov.co/files/investigaciones/poblacion/convivencia/2021/Bol_ECSC_2021.pdf. [Último acceso: 11 Abril 2024].
- [4] Oficina de Análisis de Información y Estudios Estratégicos, «Secretaría Distrital de Seguridad, Convivencia y Justicia,» Grupo Valorem, Septiembre 2023. [En línea]. Available: https://scj.gov.co/sites/default/files/documentos_oaiee/Reporte_bogota_2023_09.pdf. [Último acceso: 11 Abril 2024].
- [5] Redacción Bogotá, «El Tiempo,» 7 Febrero 2024. [En línea]. Available: <https://www.eltiempo.com/bogota/las-localidades-y-zonas-de-la-ciudad-donde-mas-roban-bicicletas-852165>. [Último acceso: 11 Abril 2024].
- [6] P. Ponsa y A. Granollers, Diseño y automatización industrial, Cataluña: Universitat politècnica de catalunya, 2009.
- [7] R. Tocci y N. Widmer, Sistemas digitales: principios y aplicaciones, Pearson educación, 2003.
- [8] F. Bellas, R. Unanue y V. Fernández, Lenguajes de programación y procesadores, Editorial centro de estudios Ramon Areces SA, 2016.
- [9] N. Wirth, Algoritmos y estructuras de datos (No. 04; QA76. 6, W56.), Prentice-Hall Hispanoamericana, 1987.
- [10] Ó. Torrente, Arduino: curso práctico de formación, Alpha Editorial, 2013.
- [11] J. Carballar y J. Fálcon, WI-FI. Lo que se necesita conocer, RC libros, 2010.
- [12] A. Nambiar, «RFID technology: A review of its applications,» International Association of Engineers, Hong Kong, 2009.
- [13] L. Powers y M. Snell, «Microsoft visual studio 2008 Unleashed,» Pearson Education, 2008.
- [14] Autodesk, «Inventor: Powerful mechanical design software for your most ambitious ideas,» [En línea]. Available: <https://www.autodesk.com/products/inventor/overview>. [Último acceso: 14 Junio 2022].

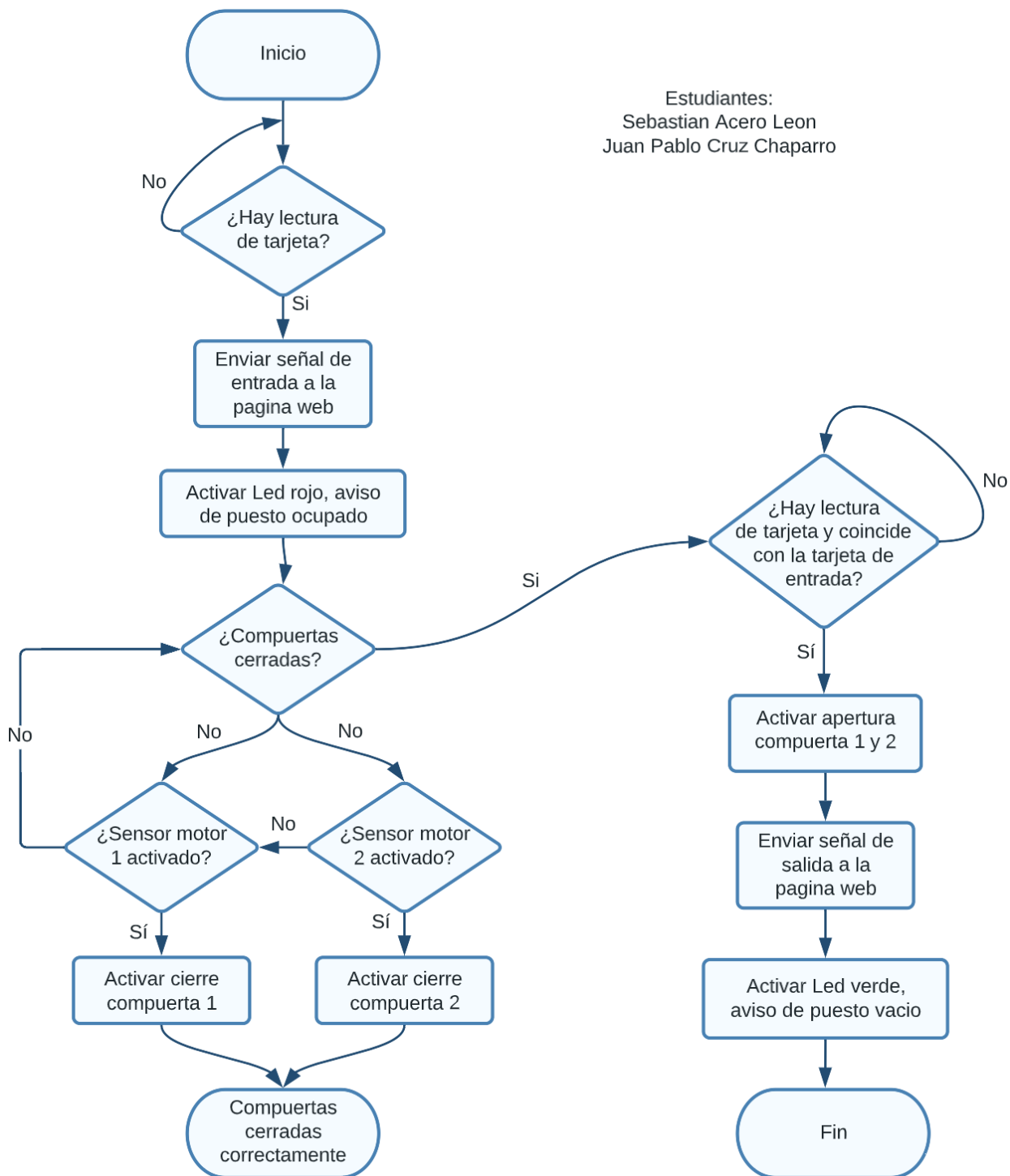
- [15] Autodesk, «Unify design, engineering, electronics, and manufacturing with Fusion 360,» [En línea]. Available: <https://www.autodesk.com/products/fusion-360/overview>. [Último acceso: 14 Junio 2022].
- [16] Autodesk, «Tinkercad,» [En línea]. Available: <https://www.tinkercad.com>. [Último acceso: 14 Junio 2022].
- [17] A. Laverde, «Aldelta Technologies, Norma IPC 2221B para diseño de PCB - actualización 2012,» 23 Febrero 2021. [En línea]. Available: <https://www.aldeitatec.com/blog-diseno-con-normas-y-certificaciones/norma-ipc-2221b-para-diseno-de-pcb-actualizacion/>. [Último acceso: 11 Abril 2024].
- [18] d'Art, Escola and de Vic, Superior de Disseny, «eartvic,» 2018. [En línea]. Available: <https://perio.unlp.edu.ar/catedras/ecal/wp-content/uploads/sites/125/2020/04/Psicologia-del-color.pdf>. [Último acceso: 11 Abril 2024].

ANEXOS

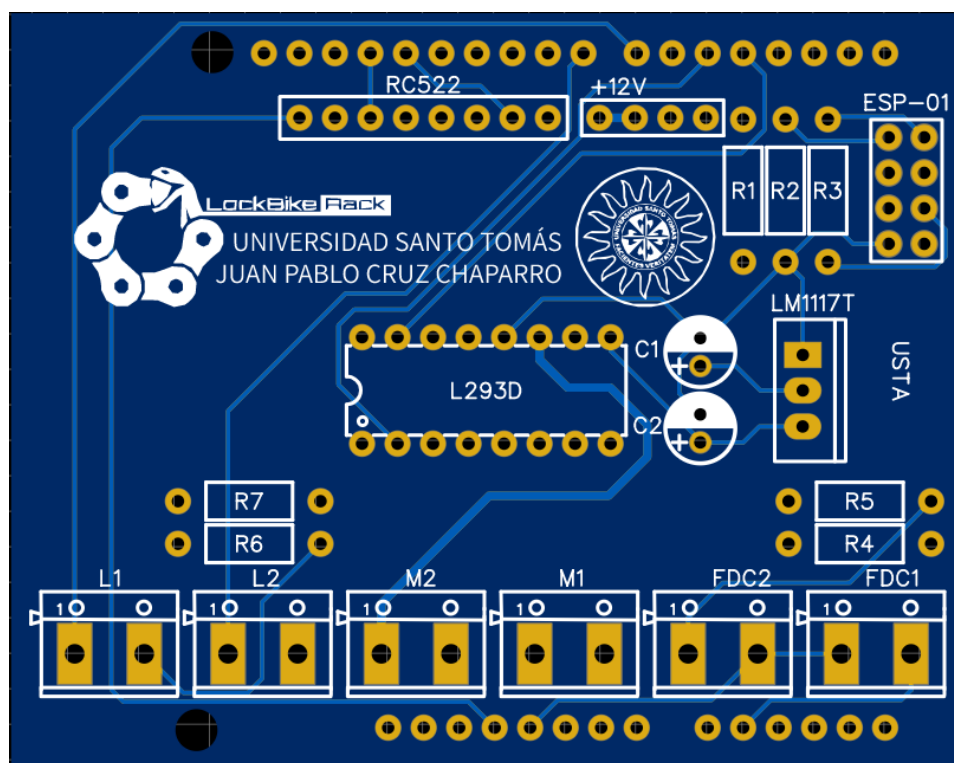
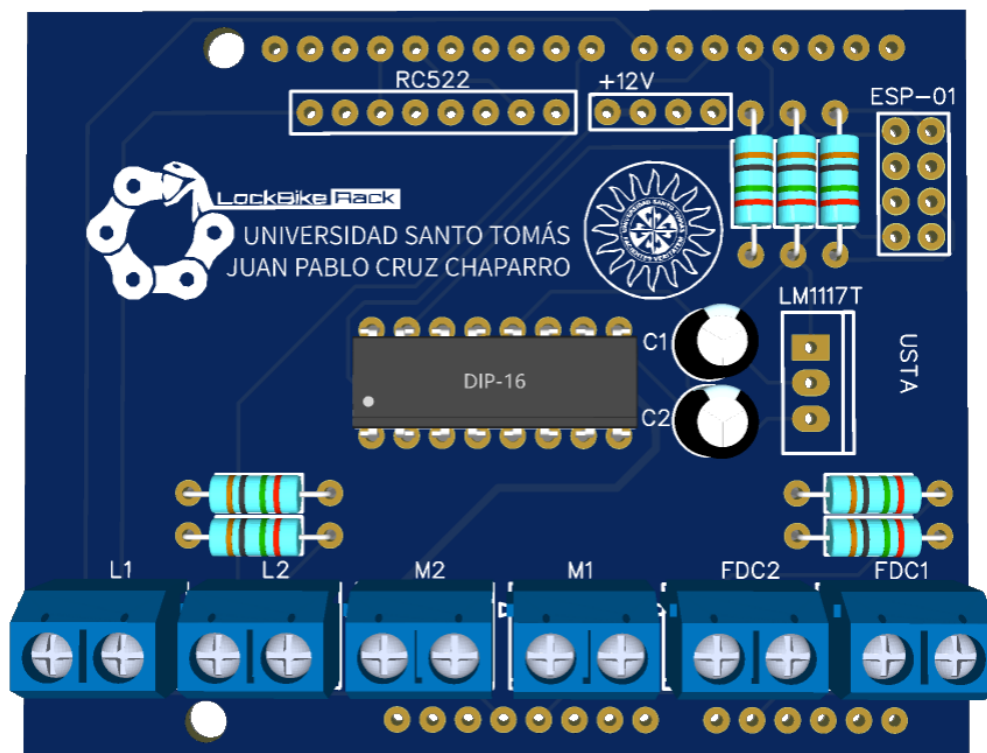
Anexo 1. Diagrama de sistemas electrónicos

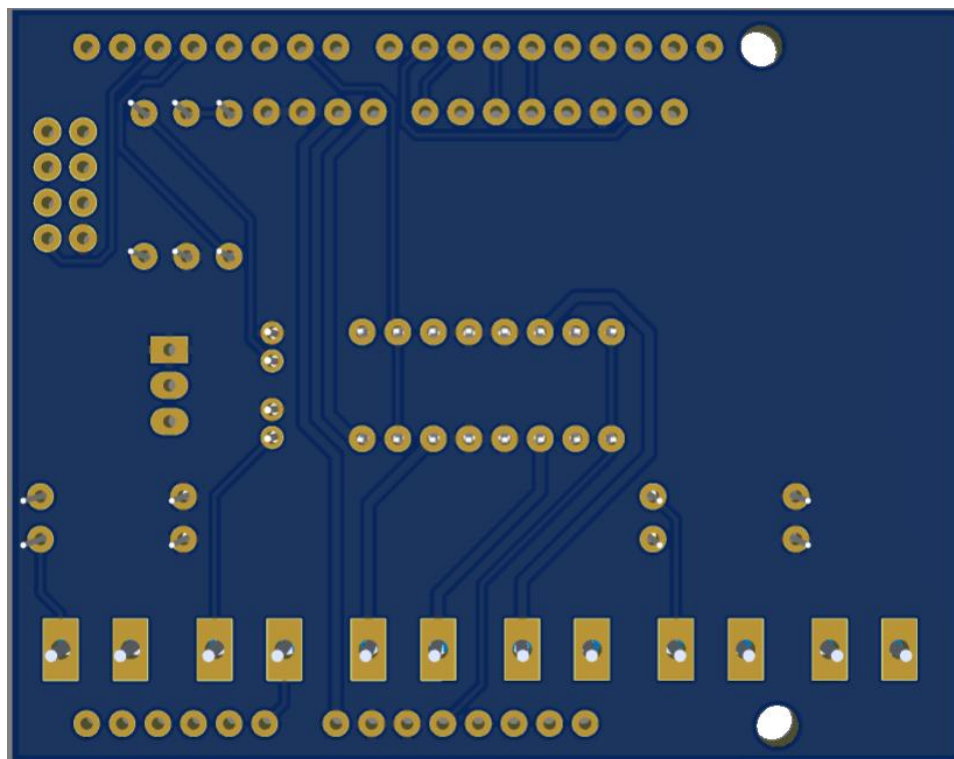
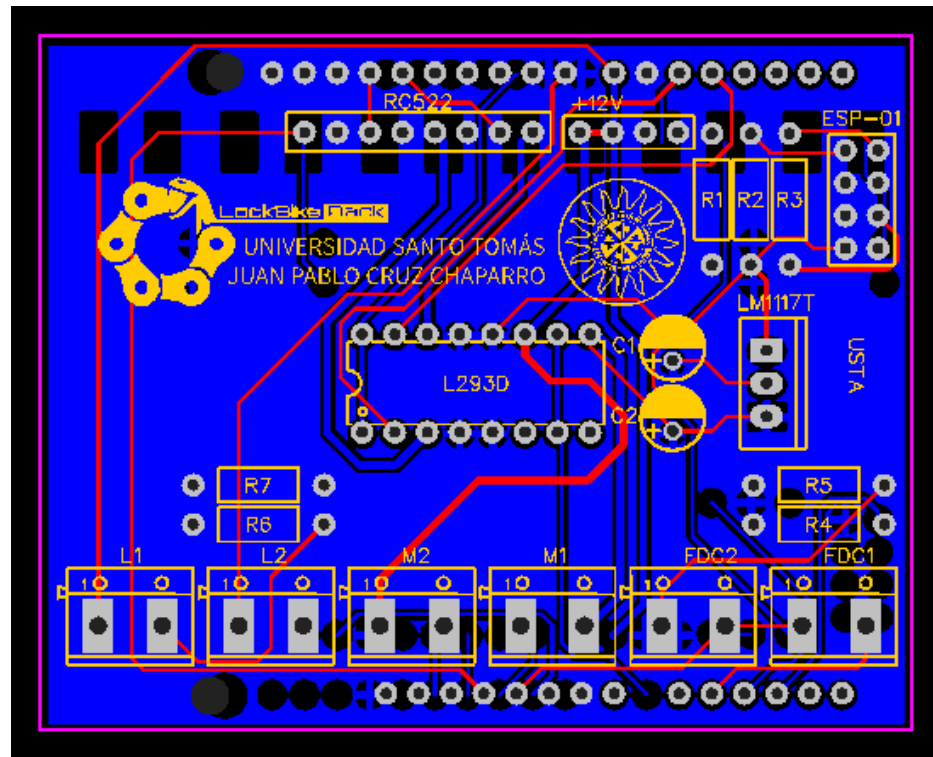


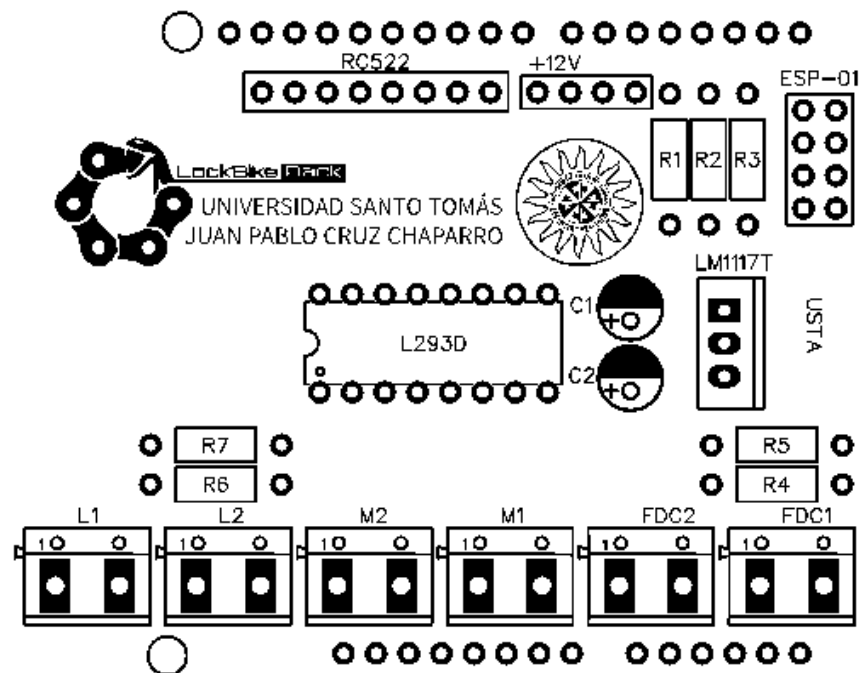
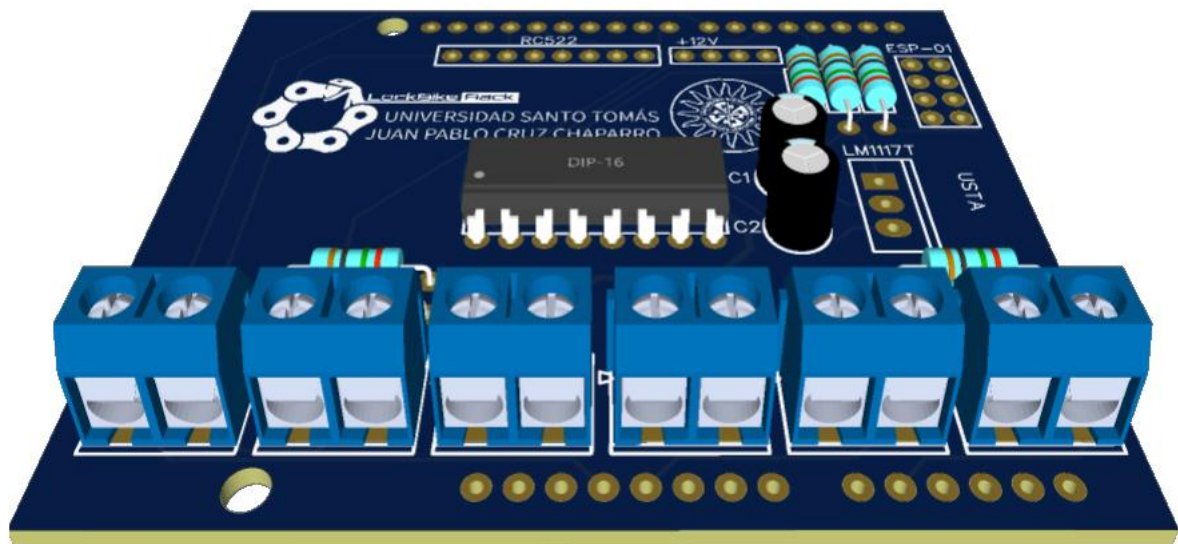
Anexo 2. Diagrama de flujo del sistema

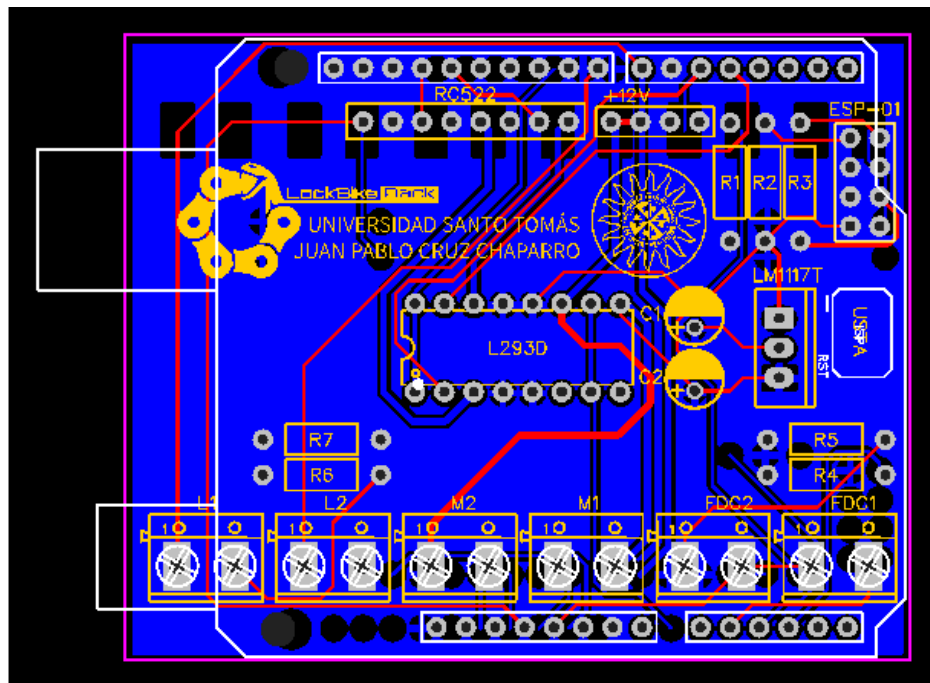
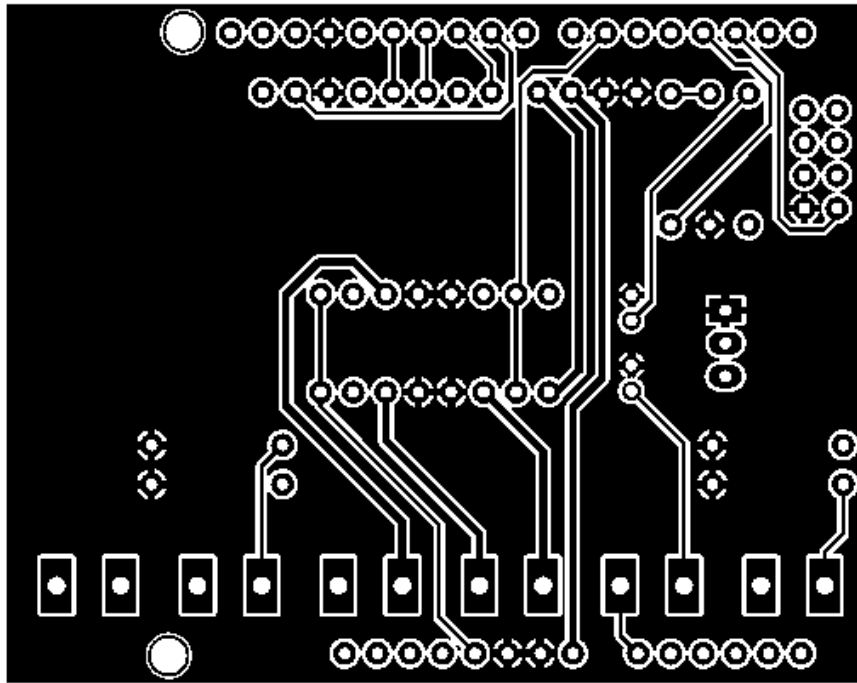


Anexo 3. Planos del circuito principal

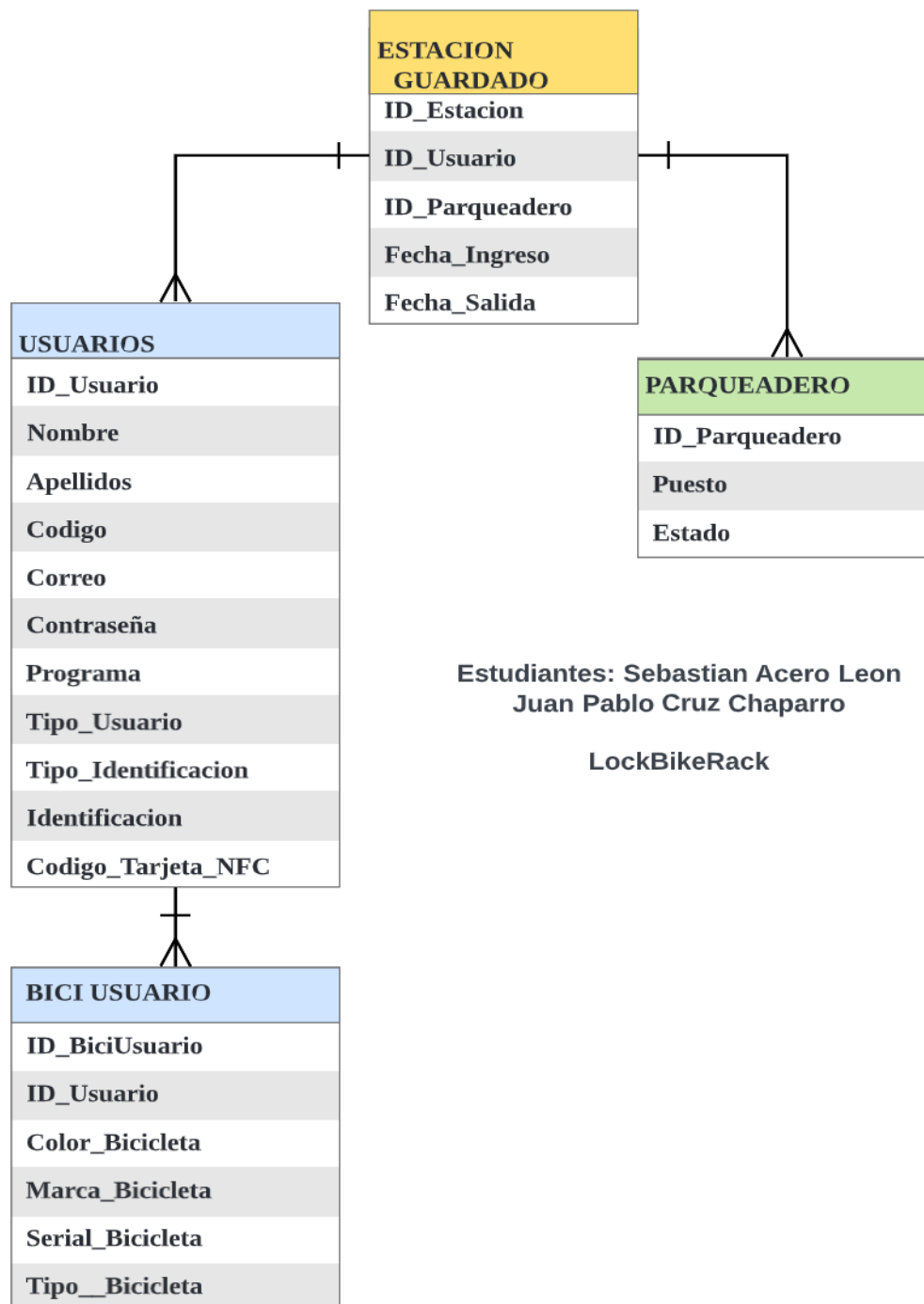




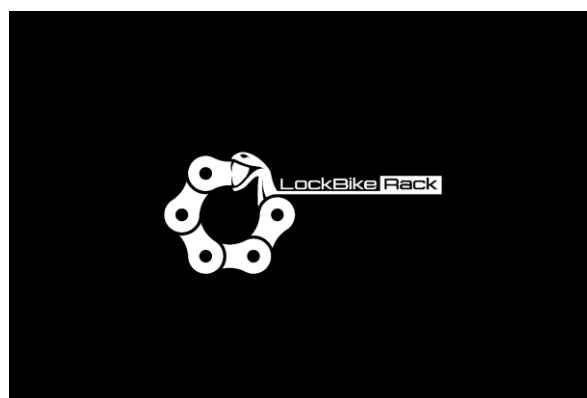
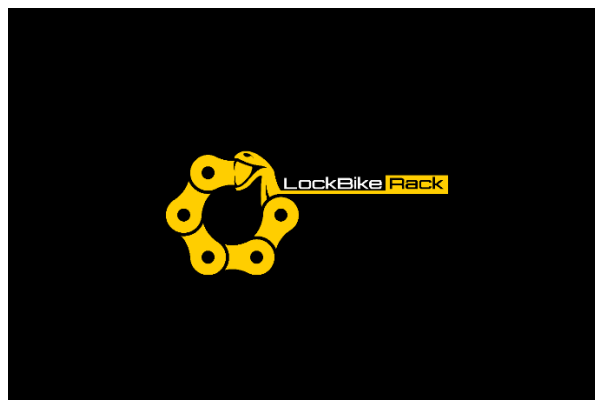




Anexo 4. Esquema de la base de datos

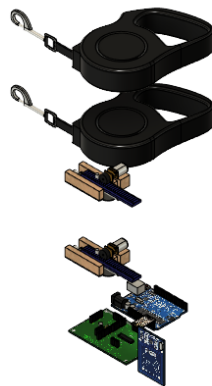
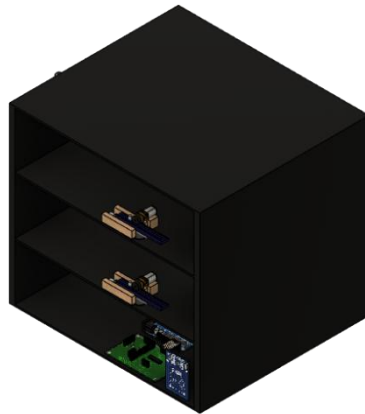
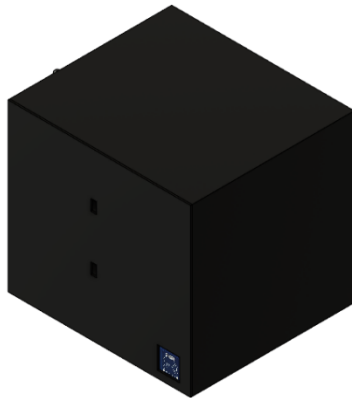


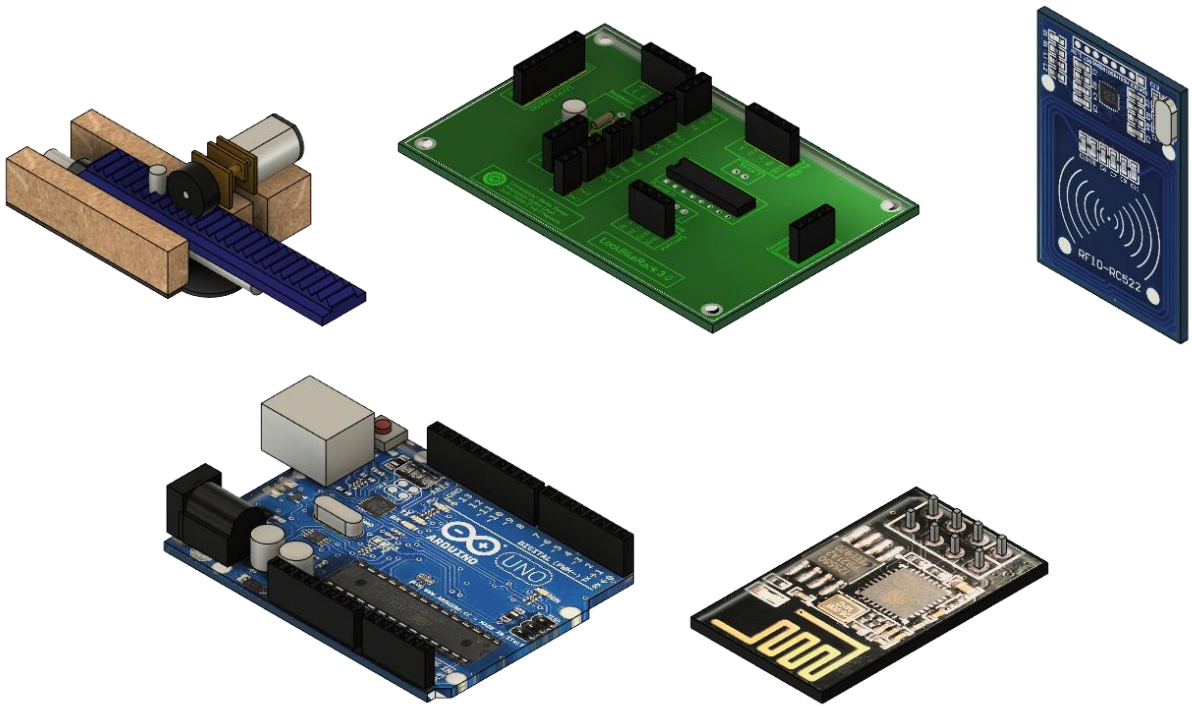
Anexo 5. Diseño del logotipo

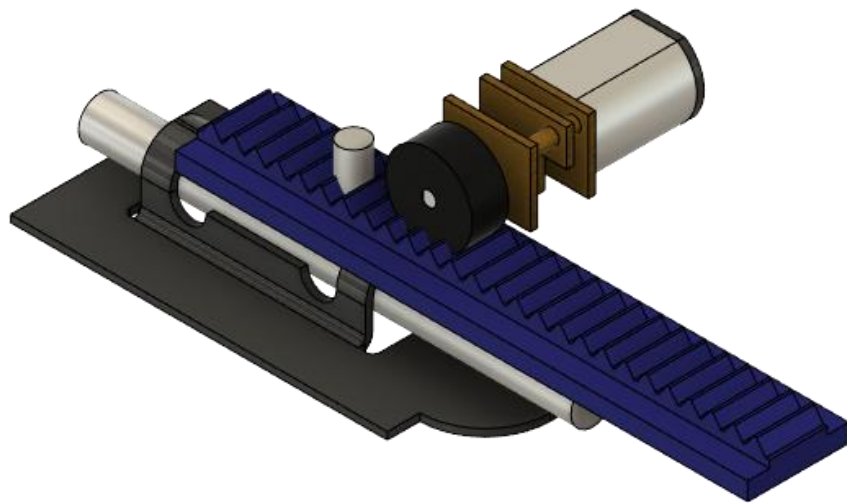
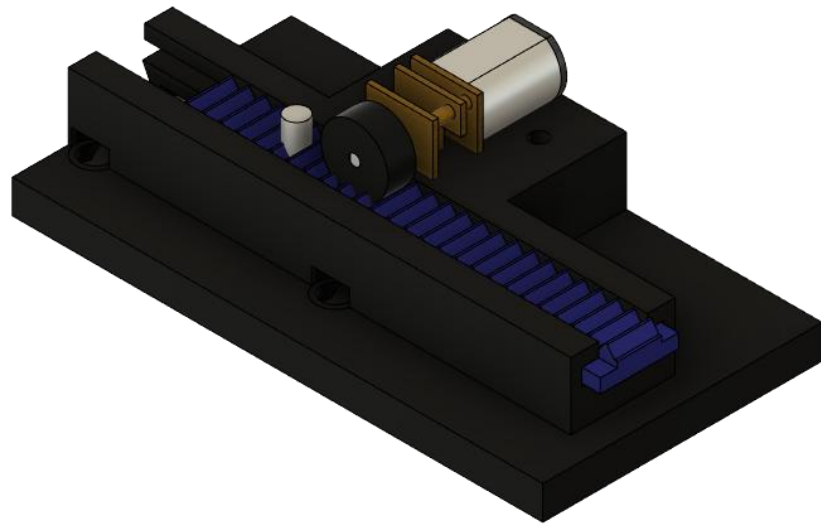


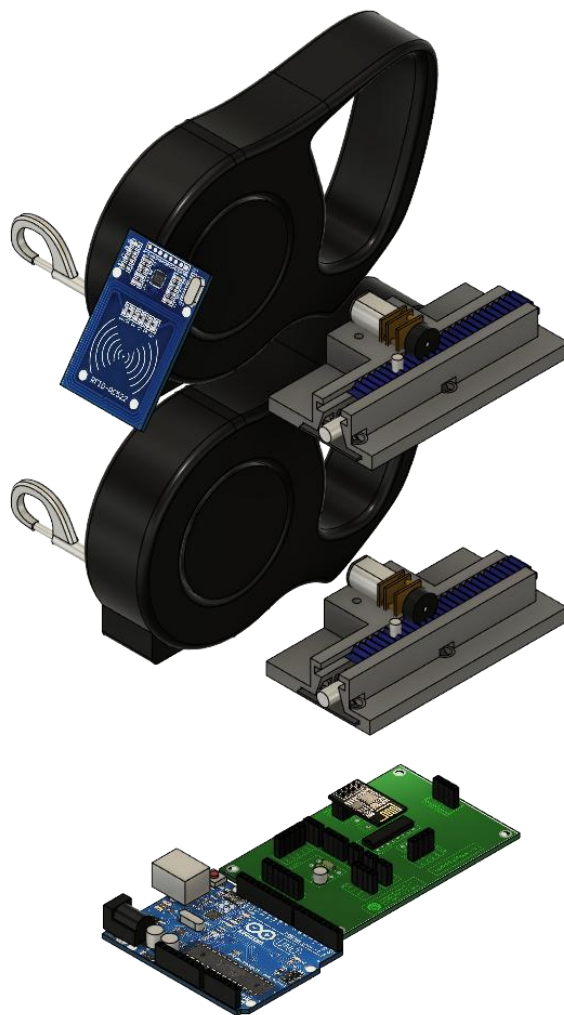
Anexo 6. Modelos 3D

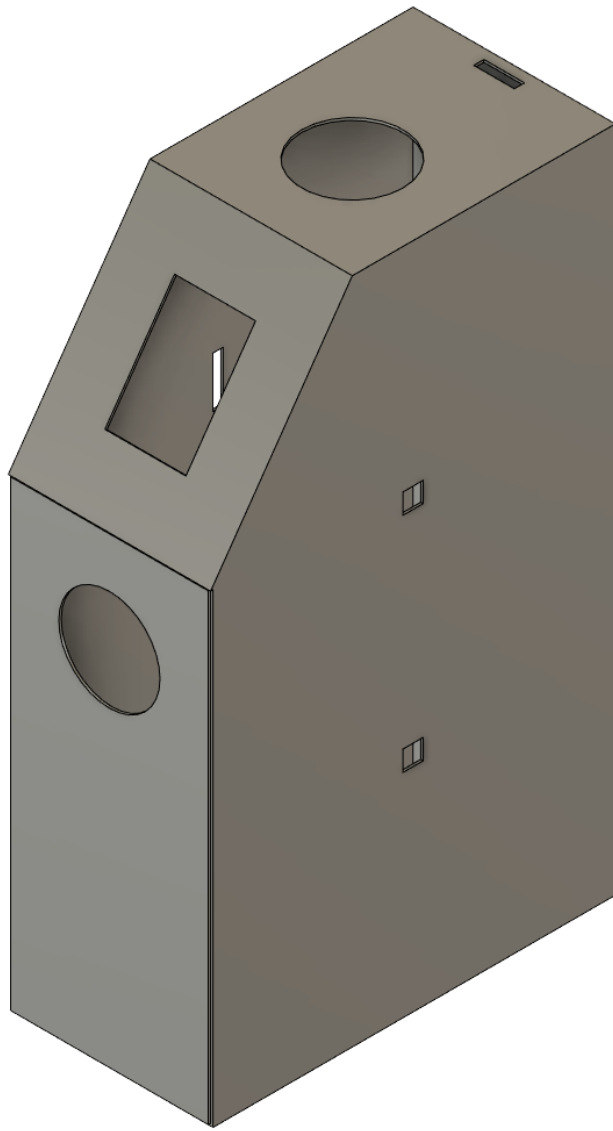


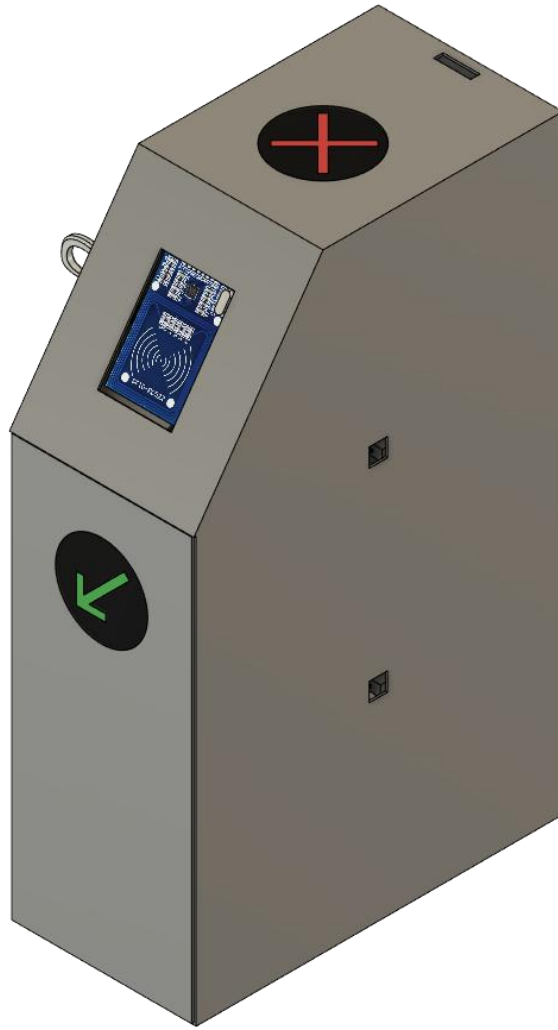


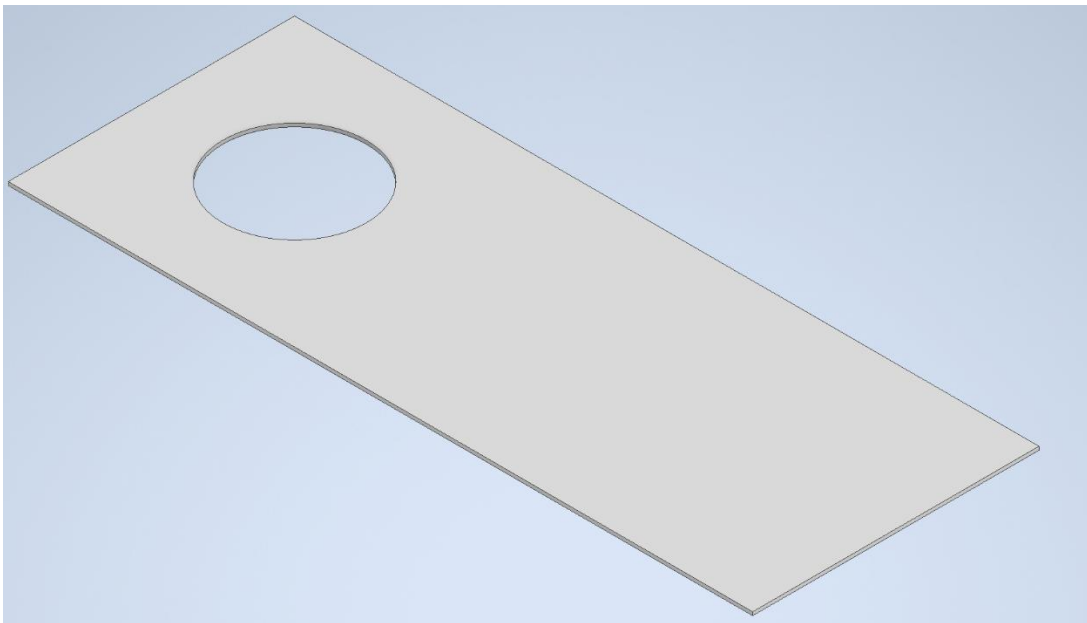
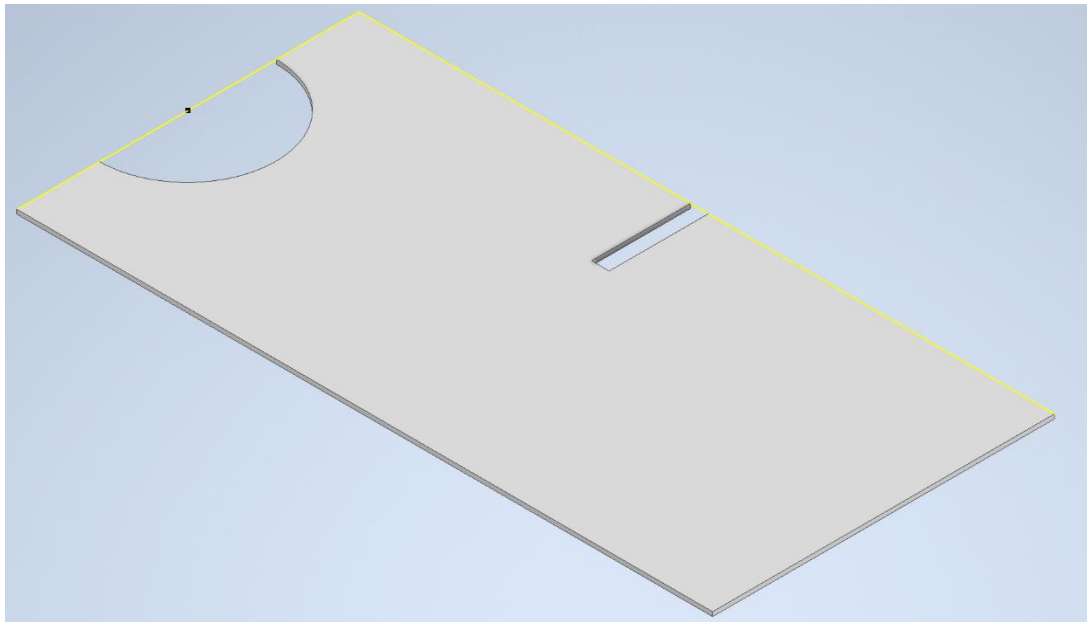


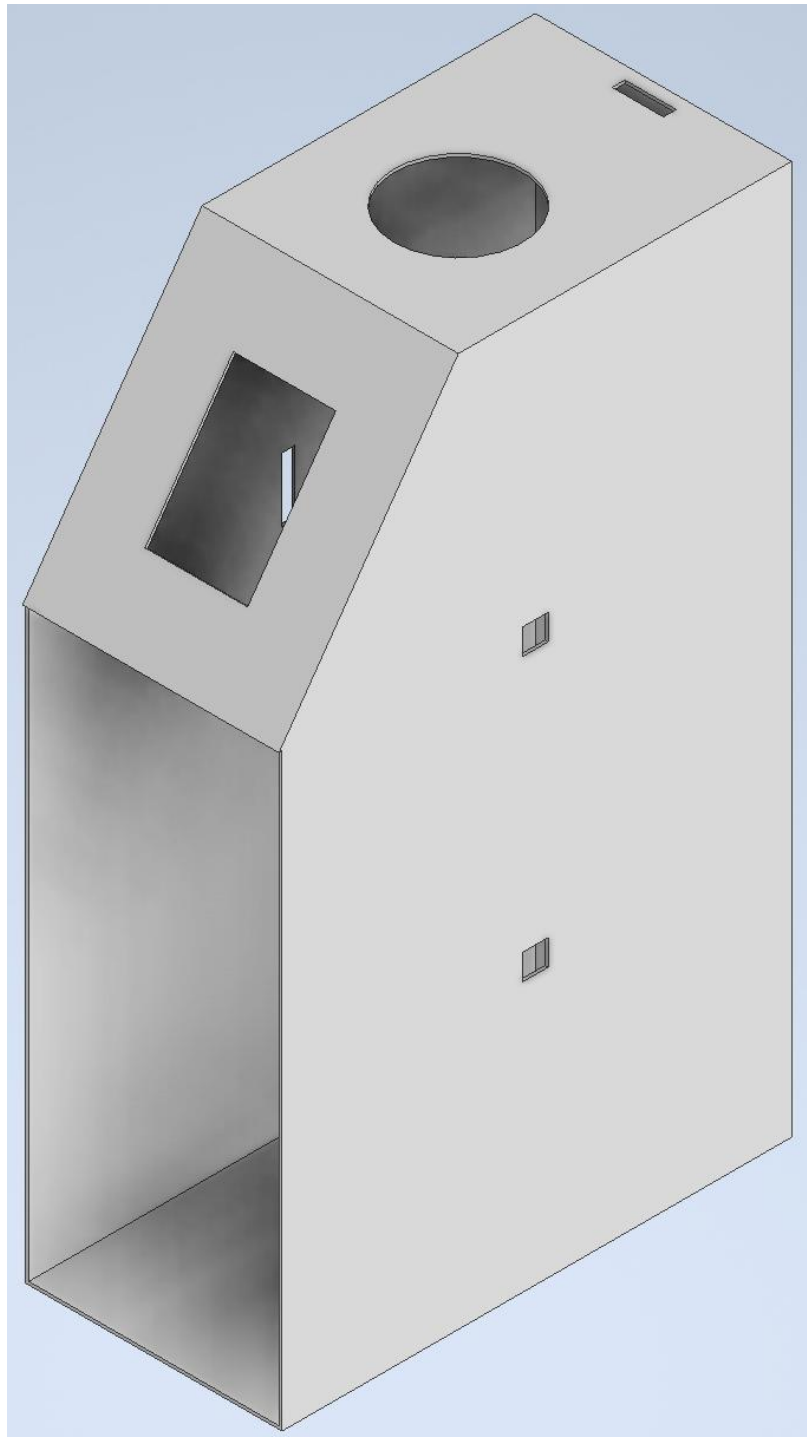


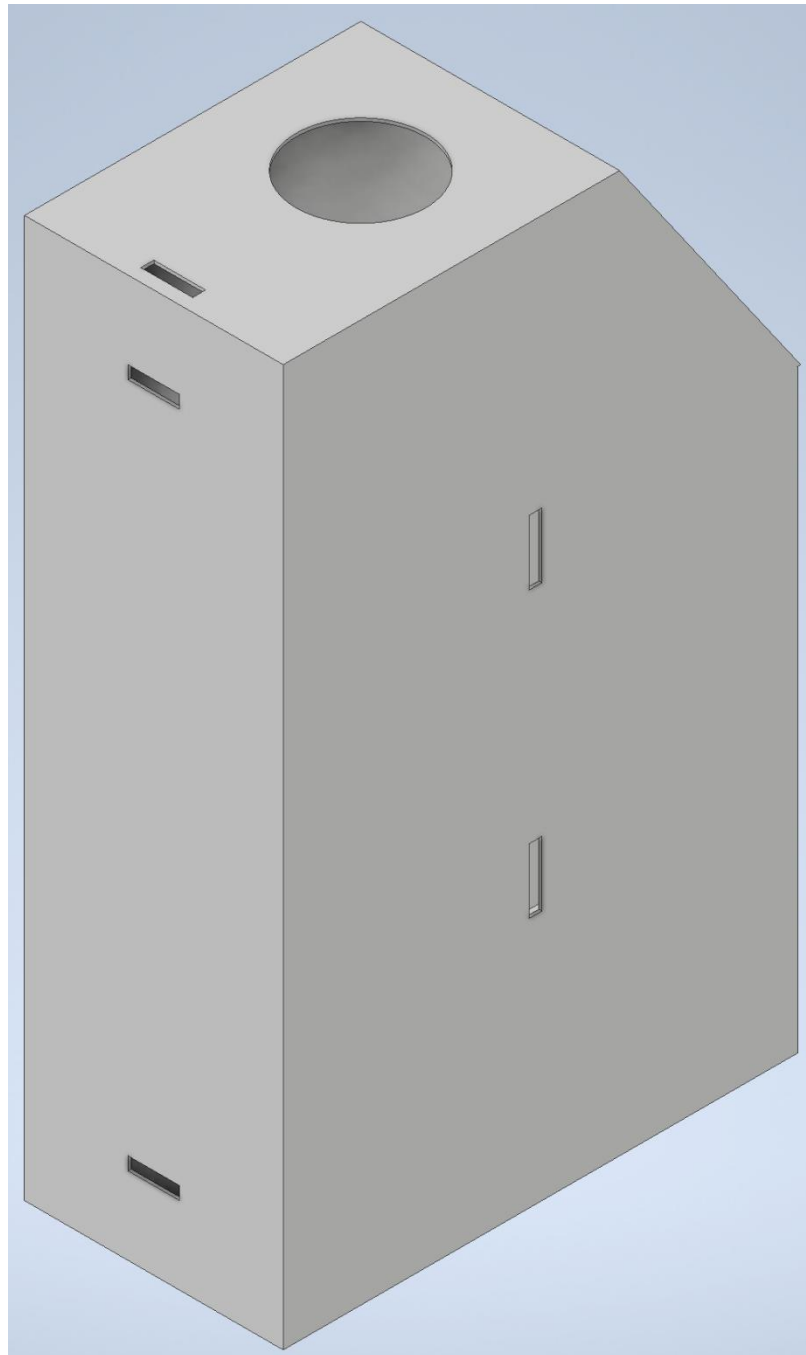


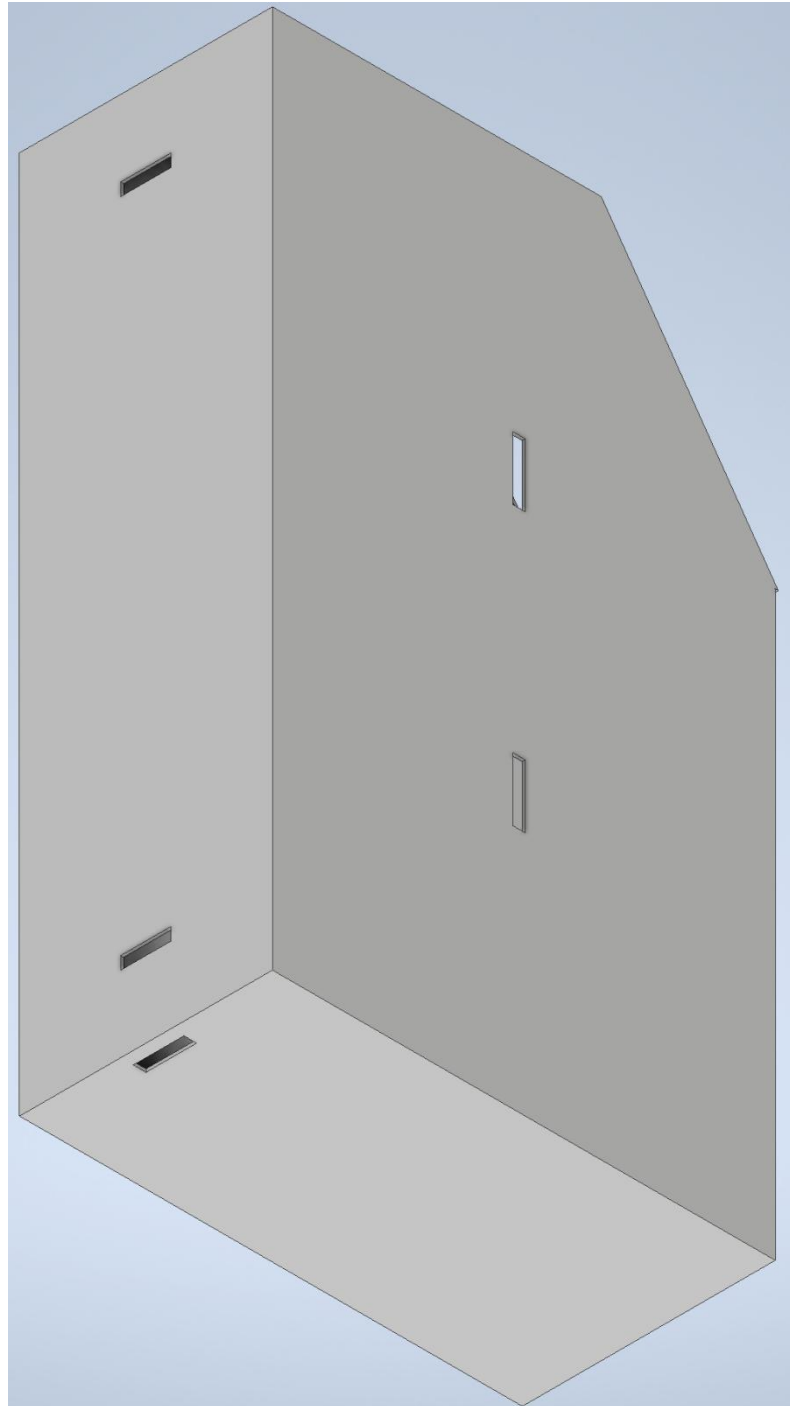




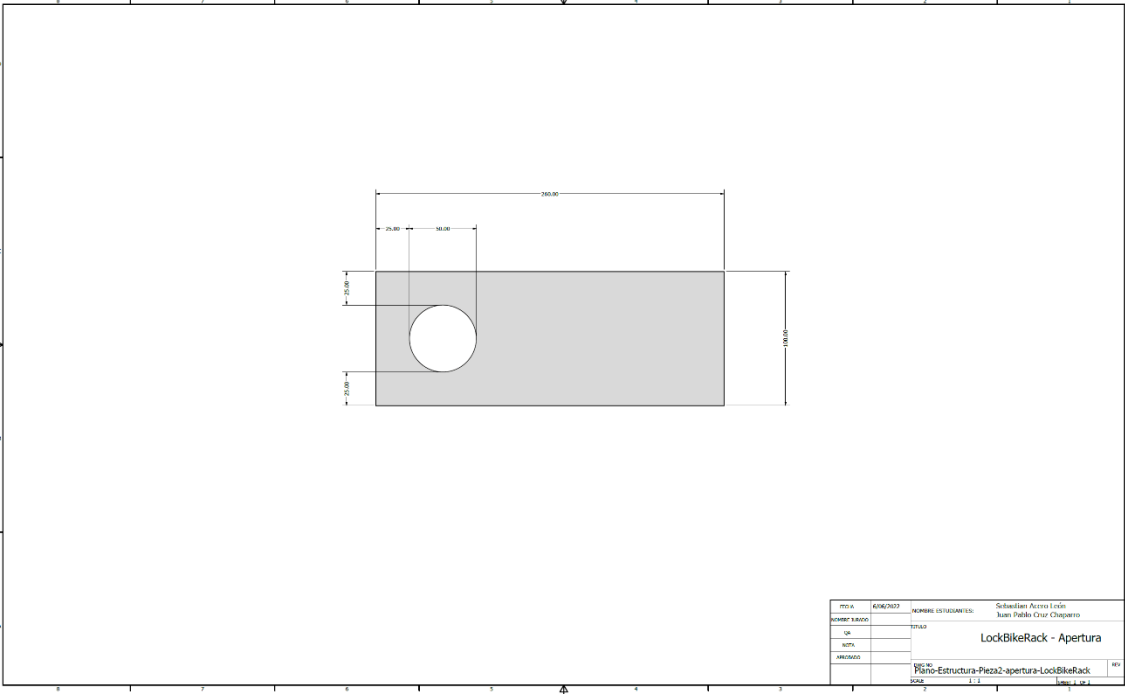
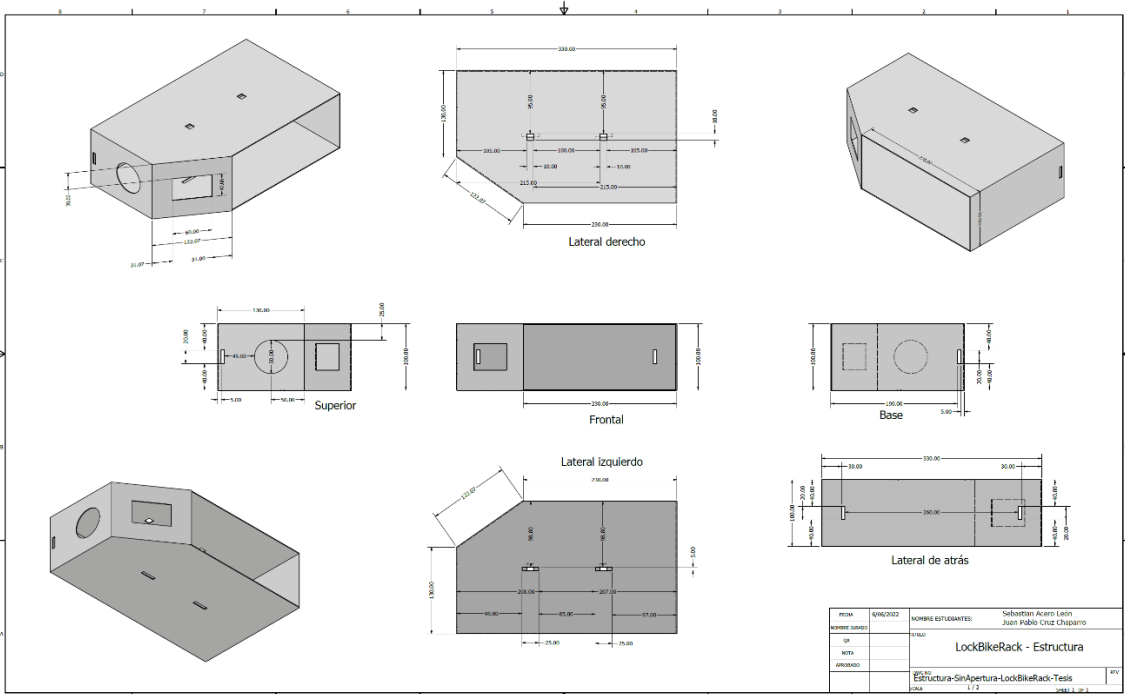




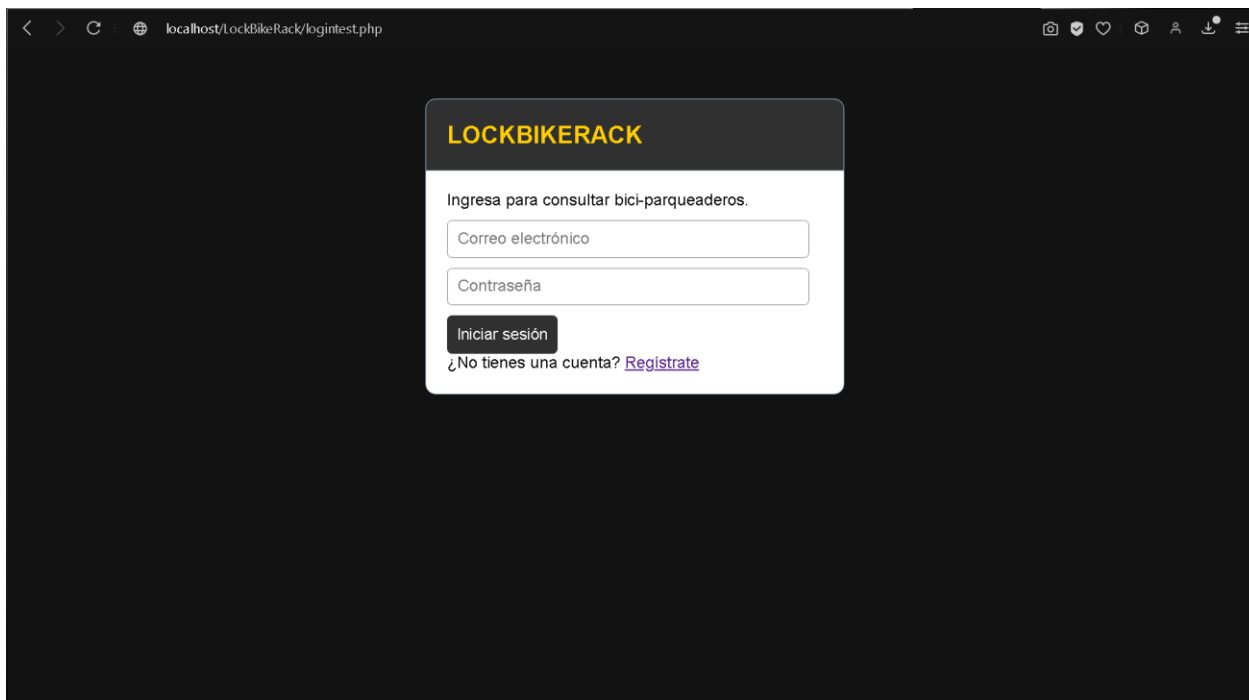




Anexo 7. Planos de la cubierta y soporte de pestillo



Anexo 8. Capturas de la página web



A screenshot of a web browser displaying the login page for LOCKBIKERACK. The browser's address bar shows 'localhost/LockBikeRack/logintest.php'. The page has a dark background with a central white login form. The form is titled 'LOCKBIKERACK' in yellow. Below the title, it says 'Ingresa para consultar bici-parqueaderos.' followed by input fields for 'Correo electrónico' and 'Contraseña'. There is a dark 'Iniciar sesión' button and a link to '¿No tienes una cuenta? [Regístrate](#)'.

LOCKBIKERACK

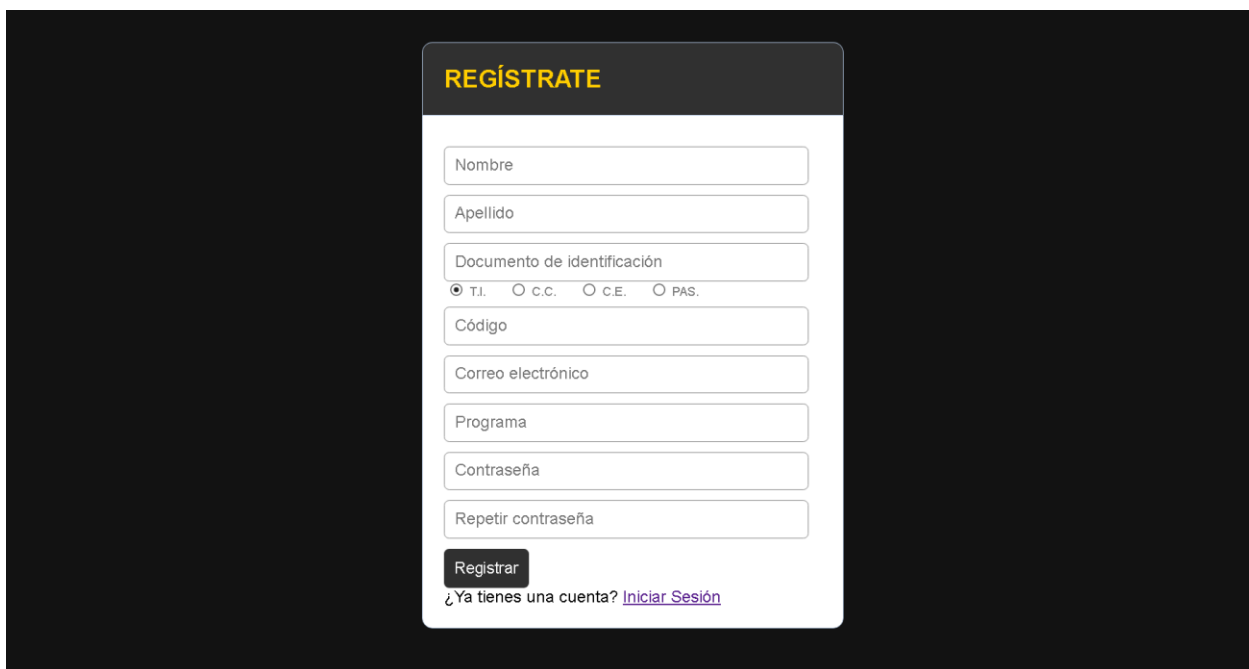
Ingresa para consultar bici-parqueaderos.

Correo electrónico

Contraseña

Iniciar sesión

¿No tienes una cuenta? [Regístrate](#)



A screenshot of a web browser displaying the registration page for LOCKBIKERACK. The page has a dark background with a central white registration form. The form is titled 'REGÍSTRATE' in yellow. It contains several input fields: 'Nombre', 'Apellido', 'Documento de identificación', 'Código', 'Correo electrónico', 'Programa', 'Contraseña', and 'Repetir contraseña'. Below the 'Documento de identificación' field, there are radio buttons for 'T.I.', 'C.C.', 'C.E.', and 'PAS.', with 'T.I.' selected. There is a dark 'Registrar' button and a link to '¿Ya tienes una cuenta? [Iniciar Sesión](#)'.

REGÍSTRATE

Nombre

Apellido

Documento de identificación

☒ T.I. ☐ C.C. ☐ C.E. ☐ PAS.

Código

Correo electrónico

Programa

Contraseña

Repetir contraseña

Registrar

¿Ya tienes una cuenta? [Iniciar Sesión](#)

Anexo 9. Códigos de programación

```
#include <SoftwareSerial.h>
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9

SoftwareSerial mySerial(2, 3);

MFRC522 rfid(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

byte nuidPICC[4];
String payload;
bool Lock, M1, M2;
static unsigned long Timer1, Timer2;
const int FDC1 = A1; //Fin de carrera 1
const int FDC2 = A0; //Fin de carrera 2
const int M2E = 4; //Habilitador del Motor 2
const int M1E = 5; //Habilitador del Motor 1
const int MCom = 6; // Habilitador común de los motores
const int LED1 = 7; //Led verde
const int LED2 = 8; //Led rojo

void setup()
{
  mySerial.begin(9600);
  SPI.begin(); // Init SPI bus
  rfid.PCD_Init(); // Init MFRC522
  for (byte i = 0; i < 6; i++)
  {
    key.keyByte[i] = 0xFF;
  }
  payload = "";
  Lock = false;
  M1 = false;
  M2 = false;

  pinMode(FDC1, INPUT);
  pinMode(FDC2, INPUT);
  pinMode(MCom, OUTPUT);
```

```

pinMode(M1E, OUTPUT);
pinMode(M2E, OUTPUT);
pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);

digitalWrite(MCom, LOW);
digitalWrite(M1E, LOW);
digitalWrite(M2E, LOW);
digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
}

void loop()
{
    if ( ! rfid.PICC_IsNewCardPresent())
        return;

    if ( ! rfid.PICC_ReadCardSerial())
        return;

    MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);

    if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
        piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
        piccType != MFRC522::PICC_TYPE_MIFARE_4K)
        return;

    if (nuidPICC[0] != 0x00 || nuidPICC[1] != 0x00 || nuidPICC[2] != 0x00 ||
nuidPICC[3] != 0x00)
    {
        if (rfid.uid.uidByte[0] != nuidPICC[0] || rfid.uid.uidByte[1] != nuidPICC[1]
|| rfid.uid.uidByte[2] != nuidPICC[2] || rfid.uid.uidByte[3] != nuidPICC[3] )
        {

            //No es la misma tarjeta, led rojo parpadea y resetea el loop. (ERROR:
PUESTO OCUPADO Y USUARIO DESCONOCIDO)

            //Serial.println("¡ACCESO DENEGADO!");
            digitalWrite(LED2, LOW);
            delay(50);
            digitalWrite(LED2, HIGH);

```

```

    delay(100);
    digitalWrite(LED2, LOW);
    delay(50);
    digitalWrite(LED2, HIGH);
    delay(100);
    digitalWrite(LED2, LOW);
    delay(50);
    digitalWrite(LED2, HIGH);
    delay(150);
}
else
{

    //Es la misma tarjeta, led rojo se apaga y led verde se enciende,
    desbloquea los seguros. (USUARIO RETIRA LA BICICLETA)
    //Reset a la variable nuidPICC y envía "OK" al ESP8266.

    //Serial.println("¡Vuelva pronto!");
    digitalWrite(LED2, LOW);
    delay(50);
    digitalWrite(LED1, HIGH);
    delay(100);
    digitalWrite(LED1, LOW);
    delay(50);
    digitalWrite(LED1, HIGH);
    delay(100);
    digitalWrite(LED1, LOW);
    delay(50);

    mySerial.print("EXIT");

    while (payload != "Registro de salida exitoso.")
    {
        if (mySerial.available() > 0) payload = mySerial.readString();
        payload = payload.substring(0, payload.length() - 2);

        digitalWrite(LED2, HIGH);
    }

    //mySerial.print("OK");
    digitalWrite(LED2, LOW);
    //Serial.println("¡Vuelva pronto!");

```

```

    digitalWrite(MCom, HIGH);
    delay(218);
    digitalWrite(MCom, LOW);
    M1 = false;
    M2 = false;
    Lock = false;
    digitalWrite(LED1, HIGH);
    delay(150);

    payload = "";
    memset(nuidPICC, 0, sizeof(nuidPICC));
    delay(500);
}
}

else
{

    //Tarjeta nueva, led verde se apaga y led rojo se enciende, permite el cierre
de los seguros. (USUARIO INGRESA LA BICICLETA)
    //Guarda el UID leído en la variable nuidPICC, lo convierte en String y lo
envía al ESP8266.

    //Serial.println("¡Bienvenido!");
    digitalWrite(LED1, LOW);
    delay(50);
    digitalWrite(LED1, HIGH);
    delay(100);
    digitalWrite(LED1, LOW);
    delay(50);
    digitalWrite(LED1, HIGH);
    delay(100);
    digitalWrite(LED1, LOW);
    delay(50);

    for (byte i = 0; i < 4; i++)
    {
        nuidPICC[i] = rfid.uid.uidByte[i];
    }

    byte uidSize = sizeof(nuidPICC);
    char uid[9];

```

```

memset(uid, 0, sizeof(uid));

if ((sizeof(uid) - 1) / 2 < uidSize)
    return;

for (int cnt = 0; cnt < uidSize; cnt++)
{
    sprintf(&uid[cnt * 2], "%02X", nuidPICC[cnt]);
}

mySerial.print(uid);

while (payload != "Registro de ingreso exitoso.")
{
    if (mySerial.available() > 0) payload = mySerial.readString();
    payload = payload.substring(0, payload.length() - 2);

    digitalWrite(LED1, HIGH);

}

//mySerial.print("OK");
digitalWrite(LED1, LOW);
//Serial.println("¡Bienvenido!");

while (!Lock)
{
    while ((M1 == false) || (M2 == false))
    {
        if ((M1 == false) && (digitalRead(FDC1) == HIGH))
        {
            Timer1 = millis();
            while (digitalRead(FDC1) == HIGH)
            {
                if ((millis() - Timer1) > 1000)
                {
                    digitalWrite(M1E, HIGH);
                    delay(218);
                    digitalWrite(M1E, LOW);
                    M1 = true;
                }
            }
        }
    }
}

```

```

    if ((M2 == false) && (digitalRead(FDC2) == HIGH))
    {
        Timer2 = millis();
        while (digitalRead(FDC2) == HIGH)
        {
            if ((millis() - Timer2) > 1000)
            {
                digitalWrite(M2E, HIGH);
                delay(218);
                digitalWrite(M2E, LOW);
                M2 = true;
            }
        }
    }
}

digitalWrite(LED2, HIGH);
delay(150);
payload = "";
Lock = true;
}

rfid.PICC_HaltA();

rfid.PCD_StopCrypto1();

}
}

```

```

#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>

const String ssid = "CRUZ0014";
const String password = "LockBikeRack";
const String host = "http://192.168.10.2/LockBikeRack/";
const String serverIn = "registro_ingreso.php";
const String serverOut = "registro_salida.php";
const int space = 1;
String uid, ans;

unsigned long lastTime = 0;

```

```

unsigned long timerDelay = 5000;

void setup() {
    Serial.begin(9600);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    Serial.println("Connecting");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    ans = "";
    Serial.println("");
    Serial.print("Connected to WiFi network with IP Address: ");
    Serial.println(WiFi.localIP());
}

void loop() {

    if ((millis() - lastTime) > timerDelay) {

        if (WiFi.status() == WL_CONNECTED) {

            WiFiClient client;
            HTTPClient http;

            if (Serial.available() > 0) {

                uid = Serial.readString();
                while (uid != NULL) {

                    if (uid == "EXIT") {

                        http.begin(client, host + serverOut);
                        http.addHeader("Content-Type", "application/x-www-form-urlencoded");
                        String httpRequestData = "space=";
                        httpRequestData.concat(space);
                        int httpResponseCode = http.POST(httpRequestData);
                        String payload = http.getString();
                        http.end();
                        uid = "";
                        Serial.println(payload);
                    } else {

```

```

        http.begin(client, host + serverIn);
        http.addHeader("Content-Type", "application/x-www-form-urlencoded");
        String httpRequestData = "space=";
        httpRequestData.concat(space);
        httpRequestData.concat("&uid=");
        httpRequestData.concat(uid);
        int httpResponseCode = http.POST(httpRequestData);
        String payload = http.getString();

        http.end();
        uid = "";

        Serial.println(payload);
    }
}
} else {
    return;
}

} else {
}
lastTime = millis();
}
}

```

```

1 ▼ <?php if (count($errors) > 0) : ?>
2 ▼     <div class="error">
3 ▼         <?php foreach ($errors as $error) : ?>
4             <p><?php echo $error ?></p>
5             <?php endforeach ?>
6         </div>
7     <?php endif ?>

```

```

1  <?php
2      session_start();
3
4      if (!isset($_SESSION['username'])) {
5          $_SESSION['msg'] = "Primero debes iniciar sesión.";
6          header('location: login.php');
7      }
8      if (isset($_GET['logout'])) {
9          session_destroy();
10         unset($_SESSION['username']);
11         header("location: login.php");
12     }
13 }>
14
15 <?php
16
17     $conexion=mysqli_connect('localhost','root','','DB-LockBikeRack');
18
19 }>
20
21 <!DOCTYPE html>
22 <html>
23 <head>
24     <title>Home</title>
25     <link rel="stylesheet" type="text/css" href="style.css">
26 </head>
27 <body>
28
29 <div class="header">
30     <h2>LOCKBIKERACK</h2>
31 </div>
32 <div class="content">
33     <!-- notification message -->
34     <?php if (isset($_SESSION['success'])) : ?>
35         <div class="error success" >
36             <h3>
37                 <?php
38                     echo $_SESSION['success'];
39                     unset($_SESSION['success']);
40                 ?>
41             </h3>
42         </div>
43     <?php endif ?>
44
45     <!-- logged in user information -->
46     <?php if (isset($_SESSION['username'])) : ?>
47         <p>Bienvenido <strong><?php echo $_SESSION['username']; ?></strong></p>
48         <p> <a href="index.php?logout='1'" style="color: red;">Cerrar sesión.</a> </p>
49     <?php endif ?>
50

```

```

51 <br>
52 <strong>Bici-parqueadero sede central USTA, Bogotá D.C.</strong>
53 <br>
54
55 <div style="overflow-x:auto;">
56
57 <table class= "center" style="border:3px solid #FFCD00;">
58 <tr align="center">
59 <th>Puesto</th>
60 <th>Estado</th>
61 </tr>
62
63 <?php
64 $sql="SELECT * FROM parqueadero;";
65 $result=mysqli_query($conexion,$sql);
66
67 while($mostrar=mysqli_fetch_array($result)){
68 ?>
69
70 <?php echo '<tr align="center">
71 <td>', $mostrar['Puesto'], '</td>';
72 if ($mostrar['Estado'] == 0) {
73 echo '<td align="left">● </td></tr>';
74 } elseif ($mostrar['Estado'] == 1) {
75 echo '<td align="left">● 🚲 </td></tr>';
76 }
77 ?>
78 <?php
79 }
80 ?>
81 </table>
82 </div>
83
84 </div>
85
86 </body>
87 </html>

```

```

1  <?php include('server.php') ?>
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <title>Registration system PHP and MySQL</title>
6      <link rel="stylesheet" type="text/css" href="style.css">
7  </head>
8  <body>
9      <div class="header">
10         <h2>LOCKBIKERACK</h2>
11     </div>
12
13     <form method="post" action="login.php">
14         <?php include('errors.php'); ?>
15         <p>
16             Ingresa para consultar bici-parqueaderos.
17         </p>
18         <div class="input-group">
19             <input type="text" name="email" placeholder="Correo electrónico">
20         </div>
21         <div class="input-group">
22             <input type="password" name="password"
23                 placeholder="Contraseña">
24         </div>
25         <div class="input-group">
26             <button type="submit" class="btn" name="login_email">Iniciar sesión</button>
27         </div>
28         <p>
29             ¿No tienes una cuenta? <a href="register.php">Regístrate</a>
30         </p>
31     </form>
32 </body>
33 </html>

```

```

1  <?php include('server.php') ?>
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <title>Registration system PHP and MySQL</title>
6      <link rel="stylesheet" type="text/css" href="style.css">
7  </head>
8  <body>
9      <div class="header">
10         <h2>REGÍSTRATE</h2>
11     </div>
12     <form method="post" action="register.php">
13         <?php include('errors.php'); ?>
14         <div class="input-group">
15             <input type="text" name="firstname" placeholder="Nombre" value="<?php echo $firstname; ?>">
16         </div>
17         <div class="input-group">
18             <input type="text" name="lastname" placeholder="Apellido" value="<?php echo $lastname; ?>">
19         </div>
20         <div class="input-group">
21             <input type="text" name="id_no" placeholder="Documento de identificación" value="<?php echo $id_no; ?>">
22         </div>
23         <div class="form-group">
24             <div class="col-sm-8">
25                 <label class="radio-inline"> <input type="radio" name="id_type" id="id_type" value="T.I." checked>T.I.</label>
26                 <label class="radio-inline"> <input type="radio" name="id_type" id="id_type" value="C.C.">C.C.</label>
27                 <label class="radio-inline"> <input type="radio" name="id_type" id="id_type" value="C.E.">C.E.</label>
28                 <label class="radio-inline"> <input type="radio" name="id_type" id="id_type" value="PAS.">PAS.</label>
29             </div>
30         </div>
31         <div class="input-group">
32             <input type="text" name="code_no" placeholder="Código" value="<?php echo $code_no; ?>">
33         </div>
34         <div class="input-group">
35             <input type="email" name="email" placeholder="Correo electrónico" value="<?php echo $email; ?>">
36         </div>
37         <div class="input-group">
38             <input type="text" name="program" placeholder="Programa" value="<?php echo $program; ?>">
39         </div>
40         <div class="input-group">
41             <input type="password" name="password_1" placeholder="Contraseña">
42         </div>
43         <div class="input-group">
44             <input type="password" name="password_2" placeholder="Repetir contraseña">
45         </div>
46         <div class="input-group">
47             <button type="submit" class="btn" name="reg_user">Registrar</button>
48         </div>
49         <p>
50             ¿Ya tienes una cuenta? <a href="login.php">Iniciar Sesión</a>
51         </p>
52     </form>
53 </body>
54 </html>

```

```

1  <?php
2  $mysqli = new mysqli('localhost', 'root', '', 'DB-LockBikeRack');
3  $uid = $_POST['uid'];
4  $space = $_POST['space'];
5  $state = 1;
6
7  if (mysqli_connect_errno()) {
8      printf("Error de conexión: %s.\n", mysqli_connect_error());
9      exit();
10 }
11
12 if ($stmt = $mysqli->prepare("SELECT ID_Usuario FROM Usuarios WHERE Codigo_Tarjeta = ?;")) {
13     $stmt->bind_param("s", $uid);
14     $stmt->execute();
15     $stmt->bind_result($ID_Usuario);
16     $stmt->fetch();
17     $stmt->close();
18
19     if ($ID_Usuario != NULL)
20     {
21
22         if ($stmt = $mysqli->prepare("UPDATE Parqueadero SET Estado = ? WHERE Puesto = ?;"))
23         {
24             $stmt->bind_param("ss", $state, $space);
25             $stmt->execute();
26             $stmt->close();
27
28             if ($stmt = $mysqli->prepare("SELECT ID_Parqueadero FROM Parqueadero WHERE Puesto = ?;"))
29             {
30                 $stmt->bind_param("s", $space);
31                 $stmt->execute();
32                 $stmt->bind_result($ID_Parqueadero);
33                 $stmt->fetch();
34                 $stmt->close();
35
36                 if ($stmt = $mysqli->prepare("INSERT INTO Estacion_Guardado (ID_Estacion,
37                                     ID_Usuario,
38                                     ID_Parqueadero,
39                                     Fecha_Ingreso,
40                                     Fecha_Salida) VALUES (NULL, ?, ?, NOW(), NULL);"))
41                 {
42                     $stmt->bind_param("ss", $ID_Usuario, $ID_Parqueadero);
43                     $stmt->execute();
44                     $stmt->close();
45                     echo "Registro de ingreso exitoso.";
46                 }
47                 else{
48                     echo "No fue posible realizar el INSERT.";
49                 }
50

```

```
51         }
52         else{
53             echo "No se encontró el ID_Parqueadero.";
54         }
55     }
56     else{
57         echo "No fue posible realizar el UPDATE.";
58     }
59 }
60 }
61 else{
62     echo "No se encontró el ID_Usuario.";
63 }
64 }
65 }
66 else{
67     echo "No se ejecutó la búsqueda correctamente.";
68 }
69 }
70 $mysqli->close();
71 ?>
```

```

1  <?php
2  $mysqli = new mysqli('localhost', 'root', '', 'DB-LockBikeRack');
3  $space = $_POST['space'];
4  $state = 0;
5
6  if (mysqli_connect_errno()) {
7      printf("Error de conexión: %s.\n", mysqli_connect_error());
8      exit();
9  }
10
11  if ($stmt = $mysqli->prepare("SELECT ID_Parqueadero FROM Parqueadero WHERE Puesto = ?;"))
12  {
13      $stmt->bind_param("s", $space);
14      $stmt->execute();
15      $stmt->bind_result($ID_Parqueadero);
16      $stmt->fetch();
17      $stmt->close();
18
19      if ($stmt = $mysqli->prepare("UPDATE Parqueadero SET Estado = ? WHERE ID_Parqueadero = ?;"))
20      {
21          $stmt->bind_param("ss", $state, $ID_Parqueadero);
22          $stmt->execute();
23          $stmt->close();
24
25          if ($stmt = $mysqli->prepare("UPDATE Estacion_Guardado SET Fecha_Salida = NOW() WHERE Fecha_Salida IS NULL AND ID_Parqueadero = ?;"))
26          {
27              $stmt->bind_param("s", $ID_Parqueadero);
28              $stmt->execute();
29              $stmt->close();
30              echo "Registro de salida exitoso.";
31          }
32          else{
33              echo "No fue posible realizar el UPDATE de salida.";
34          }
35      }
36      else{
37          echo "No fue posible realizar el UPDATE de estado.";
38      }
39  }
40
41  }
42  else{
43      echo "No se encontró el ID_Parqueadero.";
44  }
45
46  $mysqli->close();
47  ?>

```

```

1  <?php
2  $mysqli = new mysqli('localhost', 'root', '', 'DB-LockBikeRack');
3  $space = $_POST['space'];
4  //$space = 1;
5
6  if (mysqli_connect_errno()) {
7      printf("Error de conexión: %s.\n", mysqli_connect_error());
8      exit();
9  }
10
11  if ($stmt = $mysqli->prepare("SELECT Codigo_Tarjeta FROM usuarios WHERE ID_Usuario =
    (SELECT ID_Usuario FROM estacion_guardado WHERE ID_Parqueadero = (SELECT
    ID_Parqueadero FROM Parqueadero WHERE Puesto = ?) AND Fecha_Salida IS NULL);"))
12  {
13      $stmt->bind_param("s", $space);
14      $stmt->execute();
15      $stmt->bind_result($uid);
16      $stmt->fetch();
17      $stmt->close();
18      echo $uid;
19  }
20  }
21  else{
22      echo "No se encontró el UID.";
23  }
24
25  $mysqli->close();
26  ?>

```

```

1  <?php
2  session_start();
3
4  // initializing variables
5  $firstname = "";
6  $lastname = "";
7  $id_type = "";
8  $id_no = "";
9  $code_no = "";
10 $email = "";
11 $program = "";
12 $errors = array();
13
14 // connect to the database
15 $db = mysqli_connect('localhost', 'root', '', 'DB-LockBikeRack');
16
17 // REGISTER USER
18 if (isset($_POST['reg_user'])) {
19     // receive all input values from the form
20
21     $firstname = mysqli_real_escape_string($db, $_POST['firstname']);
22     $lastname = mysqli_real_escape_string($db, $_POST['lastname']);
23     $id_type = mysqli_real_escape_string($db, $_POST['id_type']);
24     $id_no = mysqli_real_escape_string($db, $_POST['id_no']);
25     $code_no = mysqli_real_escape_string($db, $_POST['code_no']);
26     $email = mysqli_real_escape_string($db, $_POST['email']);
27     $program = mysqli_real_escape_string($db, $_POST['program']);
28     $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
29     $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);
30
31     // form validation: ensure that the form is correctly filled ...
32     // by adding (array_push()) corresponding error unto $errors array
33     if (empty($firstname)) { array_push($errors, "Firstname is required"); }
34     if (empty($lastname)) { array_push($errors, "lastname is required"); }
35     if (empty($id_type)) { array_push($errors, "ID type is required"); }
36     if (empty($id_no)) { array_push($errors, "ID number is required"); }
37     if (empty($code_no)) { array_push($errors, "Code number is required"); }
38     if (empty($email)) { array_push($errors, "Email is required"); }
39     if (empty($program)) { array_push($errors, "Program is required"); }
40     if (empty($password_1)) { array_push($errors, "Password is required"); }
41     if ($password_1 != $password_2) {
42         array_push($errors, "Las contraseñas no coinciden.");
43     }
44
45     // first check the database to make sure
46     // a user does not already exist with the same username and/or email
47     $user_check_query = "SELECT * FROM usuarios WHERE Correo='$email' LIMIT 1";
48     $result = mysqli_query($db, $user_check_query);
49     $user = mysqli_fetch_assoc($result);
50

```

```

51 if ($user) { // if user exists
52
53     if ($user['Correo'] === $email) {
54         array_push($errors, "Este correo ya se encuentra en uso. Pruebe con uno
55             distinto.");
56     }
57 }
58
59 // Finally, register user if there are no errors in the form
60 if (count($errors) == 0) {
61     $password = md5($password_1); //encrypt the password before saving in the database
62
63     $query = "INSERT INTO usuarios (ID_Usuario, Nombre, Apellido, Tipo_Identificacion
64         , Identificacion, Codigo, Correo, Contraseña, Programa, Tipo_Usuario,
65         Codigo_Tarjeta) VALUES (NULL, '$firstname', '$lastname', '$id_type', '$id_no',
66         '$code_no', '$email', '$password', '$program', 'Usuario', NULL)";
67     mysqli_query($db, $query);
68     $_SESSION['username'] = $email;
69     $_SESSION['success'] = "Has iniciado sesión.";
70     header('location: index.php');
71 }
72
73 // LOGIN USER
74 if (isset($_POST['login_email'])) {
75     $email = mysqli_real_escape_string($db, $_POST['email']);
76     $password = mysqli_real_escape_string($db, $_POST['password']);
77
78     if (empty($email)) {
79         array_push($errors, "El correo que has ingresado no corresponde a una cuenta
80             existente.");
81     }
82     if (empty($password)) {
83         array_push($errors, "La contraseña que ingresaste es incorrecta. ¿Has olvidado
84             tu contraseña?");
85     }
86
87     if (count($errors) == 0) {
88         $password = md5($password);
89         $query = "SELECT * FROM usuarios WHERE Correo='$email' AND Contraseña='$password'
90             ";
91         $results = mysqli_query($db, $query);
92         if (mysqli_num_rows($results) == 1) {
93             $_SESSION['username'] = $email;
94             $_SESSION['success'] = "Has iniciado sesión.";
95             header('location: index.php');
96         } else {
97             array_push($errors, "Usuario o contraseña incorrectos.");
98         }
99     }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```