

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**IMPLEMENTACIÓN DE UN SISTEMA DE INVENTARIO PARA
EQUIPOS DE LABORATORIO DE “TECNOLOGÍA ELÉCTRICA Y
ELECTRÓNICA – ÁREA DE ELECTRÓNICA DE POTENCIA Y
MICROPROCESADORES”**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO
EN ANÁLISIS EN SISTEMAS INFORMÁTICOS**

DIEGO ARMANDO CHICAIZA CHICAIZA
diego.chicaiza@epn.edu.ec

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO
EN ELECTRÓNICA Y TELECOMUNICACIONES**

ÁNGELA SILVANA LAGUA CERNA
angela.lagua@epn.edu.ec

DIRECTOR: ING. VIVIANA CRISTINA PÁRRAGA VILLAMAR MSC.
viviana.parragav@epn.edu.ec

CODIRECTOR: ING. MÓNICA DE LOURDES VINUEZA RHOR MSC.
monica.vinueza@epn.edu.ec

Quito, Diciembre 2019

DECLARACIÓN

“Nosotros, Diego Armando Chicaiza Chicaiza y Ángela Silvana Laguna Cerna, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación - COESC-, somos titulares de la obra en mención y otorgamos una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional. Entregamos toda la información técnica pertinente. En el caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente”.

Diego Armando Chicaiza Chicaiza

Ángela Silvana Laguna Cerna

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Diego Armando Chicaiza Chicaiza y Ángela Silvana Laguna Cerna, bajo nuestra supervisión.

Ing. Viviana Párraga MSc.

DIRECTOR DEL PROYECTO

Ing. Mónica Vinuesa MSc.

CODIRECTOR DEL PROYECTO

DEDICATORIA

Mi tesis la dedico a mi madre Etelvina Chicaiza Tenorio y a padre José Chicaiza Taco, por su sacrificio y esfuerzo durante toda mi carrera universitaria, aunque hemos pasado momentos muy difíciles siempre han estado brindando su comprensión, cariño y amor.

También dedico esta tesis a esas personas que nunca creyeron en mí que en vez de apoyarme fueron piedra de tropiezo en mi vida universitaria y pese a eso tomé más impulso para poder salir adelante.

Diego Armando Chicaiza Chicaiza

DEDICATORIA

A mis padres, Marina Cerna y Ángel Laguna, quienes incondicionalmente me han apoyado para culminar la carrera. A mis hermanas Viviana y Cristina, que con su ejemplo me impulsaron a no rendirme hasta cumplir una meta más.

A todos mis amigos y compañeros que compartieron gratos momentos en el transcurso de la carrera y a todos los que confían en mí y me aprecian.

Ángela Laguna

AGRADECIMIENTO

En primer lugar, quiero agradecer a DIOS por brindarme salud, fuerza y sobre todo mucha sabiduría para poder tomar buenas y malas decisiones en toda mi carrera universitario.

A mis padres José y Etelvina por enseñarme tres cosas fundamentales como son; el respeto, la honradez y la perseverancia ya que eso me lo han demostrado día a día.

A mi hermano David, quien fue un pilar fundamental con sus palabras de aliento que decía “todo problema complicado hay que respirar y pensar dos veces con cabeza fría y el problema estará resuelto antes de lo que te imagines”.

A mi sobrino Ezequiel, que con su sonrisa y locuras me acompañaba en las noches y fines de semana mientras realizaba mis deberes y proyectos universitarios.

Y finalmente agradecer a la Escuela Politécnica Nacional por haberme dado una oportunidad en la Escuela de Formación de Tecnólogos en donde me forme como un profesional.

Diego Armando Chicaiza Chicaiza

AGRADECIMIENTO

Agradezco a DIOS por la dicha de permitirme compartir un logro más junto a mis seres queridos, por brindarme fortaleza para no desfallecer en este arduo camino y por haber puesto en el a todas las personas que me han acompañado y apoyado de una u otra manera durante todo este período.

Agradezco a cada uno de los integrantes de mi familia, a mis padres por los valores inculcados, los consejos, por estar constantemente impulsándome y guiando en el transcurso de mi carrera profesional.

Agradezco a mi directora de proyecto Ing. Viviana Párraga MSc. por su paciencia y excelente dirección en este proyecto.

A mis compañeros y amigos que han estado alentándome constantemente, brindándome su compañía y haciendo los días más amenos con su carisma para poder terminar satisfactoriamente este proyecto.

Ángela Laguna

ÍNDICE DE CONTENIDOS

DECLARACIÓN	i
CERTIFICACIÓN	ii
DEDICATORIA	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDOS	vii
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xi
RESUMEN.....	1
<i>ABSTRACT</i>	2
1. INTRODUCCIÓN.....	3
1.1. Objetivo general	4
1.2. Objetivos específicos.....	4
1.3. Justificación	4
1.4. Marco teórico.....	5
2. METODOLOGÍA	7
2.1. Metodología general del proyecto.....	7
2.2. Desarrollo de la metodología <i>Scrum</i>	12
3. RESULTADOS Y DISCUSIÓN	12
3.1. Requerimientos	12
3.2. Selección de dispositivos.....	15
3.3. Funcionamiento del sistema de inventario	18
3.4. Diseño lógico.....	22
3.5. Diseño Físico.....	29
3.6. Programación del dispositivo	33
3.7. Creación del sistema <i>web</i>	36

3.8.	Estructura del proyecto de la aplicación <i>web</i>	37
3.9.	Creación y configuración de la base de datos.....	38
3.10.	Diagramas de la base de datos.....	39
3.11.	Desarrollo de la aplicación <i>web</i>	40
3.12.	Instalación.....	45
3.13.	Pruebas.....	47
4.	CONCLUSIONES Y RECOMENDACIONES	56
4.1.	Conclusiones.....	56
4.2.	Recomendaciones.....	58
5.	BIBLIOGRAFÍA.....	59
6.	ANEXOS.....	64

ÍNDICE DE FIGURAS

Figura 2.1 Diseño básico de <i>hardware</i> y <i>software</i>	8
Figura 2.2 Diagrama de flujo del proceso seguido para realizar el proyecto.	11
Figura 3.1 Arduino UNO y cable USB tipo A-B.....	15
Figura 3.2 Partes del Arduino UNO.	15
Figura 3.3 <i>Ethernet Shield</i>	16
Figura 3.4 Módulo RFID-522 y <i>Tag</i> RFID.....	16
Figura 3.5 Lector de huella digital para Arduino.	17
Figura 3.6 Antena RFID de gama UHF.	17
Figura 3.7 Zumbador electrónico SFM-27 DC 3 - 24V.	18
Figura 3.8 Diseño básico del sistema dentro del laboratorio.	19
Figura 3.9 Diagrama de flujo de registro de estudiante y equipo.....	19
Figura 3.10 Diagrama de flujo de registro de préstamo.....	20
Figura 3.11 Diagrama de flujo de registro de la devolución.....	21
Figura 3.12 Diagrama de flujo del funcionamiento de la alarma.....	22
Figura 3.13 Partes y conexiones del proyecto como un cliente 1.....	23
Figura 3.14 Partes y conexiones del proyecto como un cliente 2.....	23
Figura 3.15 Conexión final del proyecto con ambos clientes.....	24
Figura 3.16 Conexión del lector de huella.....	25
Figura 3.17 Conexión del módulo RFID-RC522.....	26
Figura 3.18 Led con su respectiva resistencia de protección.....	27
Figura 3.19 Conexión de la antena UHF y zumbador electrónico.....	27
Figura 3.20 Conexión de la placa <i>Ethernet Shield</i> en la placa Arduino Uno.....	28
Figura 3.21 Diagrama final de conexión como cliente 1.....	29
Figura 3.22 Diagrama final de conexión como cliente 2.....	30
Figura 3.23 Diagrama para establecer la conexión (modo cliente)	32
Figura 3.24 Librerías para el cliente 1.....	33
Figura 3.25 Librerías para el cliente 2.....	35
Figura 3.26 Comando para crear un nuevo proyecto en Laravel.....	36
Figura 3.27 Comando para la ejecución del proyecto de Laravel.....	37
Figura 3.28 Ejecución del proyecto en el navegador.....	37
Figura 3.29 Estructura del proyecto.....	38
Figura 3.30 Creación de la base de datos.....	38
Figura 3.31 Base de datos creada.....	38
Figura 3.32 Archivo.env modificado y configurado.....	39

Figura 3.33 Modelo físico de la base de datos.....	39
Figura 3.34 Login del sistema <i>web</i>	40
Figura 3.35 Módulo de Estudiante.....	40
Figura 3.36 Modal de registro de nuevo estudiante.....	41
Figura 3.37 Módulo Equipo.....	41
Figura 3.38 Modal de registro de nuevo equipo.....	42
Figura 3.39 Módulo de usuario.....	42
Figura 3.40 Modal de registro de usuario.....	43
Figura 3.41 Módulo de préstamos y devoluciones.....	43
Figura 3.42 Modal de préstamo de equipo.....	44
Figura 3.43 Modal de devolución de equipos.....	44
Figura 3.44 Caja del cliente 1 con su respectivo montaje.....	45
Figura 3.45 Caja del cliente 2 con su respectivo montaje.....	46
Figura 3.46 Instalación de antena UHF y zumbador.....	46
Figura 3.47 Instalación del cableado del cliente 2.....	47
Figura 3.48 Instalación del cableado del cliente 2.....	47
Figura 3.49 Registro de usuario (administrador).....	48
Figura 3.50 Usuario ya registrado.....	48
Figura 3.51 Visualización de usuario ya registrado en la base de datos.....	49
Figura 3.52 Registro del usuario (encargado del laboratorio).....	49
Figura 3.53 Usuario encargado de laboratorio registrado en la base de datos.....	50
Figura 3.54 Datos de un nuevo estudiante.....	50
Figura 3.55 Estudiante ya registrado en la base de datos.....	51
Figura 3.56 Ingreso de datos de un nuevo equipo.....	51
Figura 3.57 Equipo registrado en la base de datos.....	52
Figura 3.59 Préstamo del equipo.....	53
Figura 3.60 Mensaje de actividad guardada.....	53
Figura 3.61 Correo de notificación de préstamo.....	53
Figura 3.62 Devolución del equipo.....	54
Figura 3.63 Correo de notificación de devolución del equipo.....	54
Figura 3.64 Préstamo no autorizado.....	55
Figura 3.65 Activación del zumbador por equipo no prestado.....	55

ÍNDICE DE TABLAS

Tabla 2.1 Equipo de trabajo y asignación de roles.....	12
Tabla 3.1 Conexión de los pines del lector de huella a Arduino Uno.....	24
Tabla 3.2 Conexión de los pines del lector de huella a Arduino Uno.....	26
Tabla 3.3 Conexión de la antena UHF.....	28

RESUMEN

El presente proyecto de titulación se realizó para cubrir las necesidades que se tiene en el laboratorio de “Tecnología Eléctrica y Electrónica – Área de Electrónica de Potencia y Microprocesadores” (TEE – AEPM). El documento consta de una breve introducción, de objetivos planteados para conseguir el desarrollo del proyecto, también tiene una justificación que ayudó a resolver la problemática de préstamos en el laboratorio. Dentro del Marco teórico se incluyeron conceptos puntuales de la metodología y herramientas a utilizar para el desarrollo de la aplicación web. El proyecto también cuenta con una metodología general que abarca la construcción del área técnica de equipos electrónicos y para el área de *software* se utilizó la metodología denominada *SCRUM*.

Dentro de resultados y discusión se procedió a realizar una lista de los requerimientos basados en las conversaciones mantenidas con las persona encargada del laboratorio de TEE – AEPM y así poder orientarse de qué equipos se van a utilizar, los cuales debían tener buenas características y especificaciones técnicas referente al área de electrónica, para el área de desarrollo del *software* se tomó los mismos requerimientos para poder utilizar las herramientas y programas adecuados para el desarrollo de la aplicación *web*.

Para el desarrollo de este proyecto se realizó un diseño lógico y físico del sistema de inventario donde los diagramas de flujo formaron parte de una explicación sobre el funcionamiento del sistema de inventario. Los diseños lógicos describen las conexiones de cada equipo mismos que conforman respectivamente el cliente 1 y cliente 2 los cuales se conectan al servidor y posteriormente para poder realizar el diseño físico. Los clientes 1 y 2 cuentan con una programación mediante una placa Arduino que a su vez utiliza un lenguaje de programación de alto nivel denominado processing. Para el desarrollo del *software* se utilizó las herramientas de desarrollo libre como son: Laravel, *MySQL* y *Bootstrap*.

Luego se procedió a la instalación dando una ubicación exacta para los equipos electrónicos que consta del cliente 1, cliente 2 y una antena RFID que detecta frecuencia de la gama UHF y para el *software* se utilizó un ordenador que cumpla la funcionalidad de servidor y así realizar las respectivas pruebas sobre el funcionamiento del sistema de inventario.

Finalmente se describen las conclusiones donde se detalla lo logrado con la utilización de los equipos electrónicos y aplicación *web*. Las recomendaciones sobre el sistema de inventario detallan lo que se podría mejorar e incluir.

Palabras clave: *SCRUM*, Gama UHF, Tecnología RFID, programación, Laravel, *MySql*.

ABSTRACT

This The present degree project was carried out to cover the needs it has in the laboratory of "Electrical and Electronic Technology - Area of Power Electronics and Microprocessors" (TEE - AEPM). The constant document of a brief introduction, of objectives set to achieve the development of the project, also has a justification that helped solve the problem of loans in the laboratory. The theoretical framework includes specific concepts of the methodology and tools to use the development of the web application. The project also has a general methodology that covers the construction of the technical area of electronic equipment and for the software area the selected SCRUM methodology was modified.

Within the results and the discussion, a list of the requirements specified in the conversations held with the person in charge of the TEE - AEPM laboratory will be carried out and in this way, the guiding power of which equipment will be used, which should have good characteristics and technical specifications, this refers to the area of electronics, for the area of software development, the same requirements are required to use the appropriate tools and programs for the development of the web application.

For the development of this project, a logical and physical design of the inventory system was carried out where the flowcharts were part of an explanation of the operation of the inventory system. The logical designs describe the connections of each identical equipment that respectively make up client 1 and client 2 which are connected to the server and subsequently to perform the physical design. Clients 1 and 2 have a programming using an Arduino board that in turn uses a high level programming language indicated processing. For software development, look for free development tools such as: Laravel, MySQL and Bootstrap.

Then the installation was carried out giving an exact location for the electronic equipment consisting of client 1, client 2 and an RFID antenna that detects the frequency of the UHF range and for the software a computer that meets the functionality of the server is electronically and thus Perform the respective tests on the operation of the inventory system.

Finally, the conclusions describing what was achieved with the use of electronic equipment and the web application are described. The recommendations on the inventory system detail what could be improved and included.

Keywords: SCRUM, UHF Range, RFID Technology, programming, Laravel, MySql.

1. INTRODUCCIÓN

La Escuela de Formación de Tecnólogos (ESFOT) cuenta con laboratorios para distintas asignaturas, sin embargo, cuando los alumnos piden prestados equipos en los mismos, para realizar prácticas o comprobaciones de algún proyecto estos préstamos pueden ser fuera de las instalaciones del laboratorio o en otras aulas.

El traslado de los equipos en ocasiones ha generado malestar, debido a que algunos de éstos se pierden o son devueltos en mal estado por no llevar un registro de quien realiza el préstamo y cuántos son prestados. También ha provocado inconvenientes para la realización de posteriores prácticas dentro de los laboratorios.

Con el transcurso de los semestres se ha podido observar la necesidad del préstamo seguro de equipos fuera del laboratorio, ya que la persona encargada del laboratorio en ocasiones no se encuentra o no conoce a los estudiantes, por lo que no puede prestar los equipos de no ser para utilizar ese momento y por un par de horas, ya que, también se dictan clases en el mismo.

La ejecución de este proyecto permitió analizar el problema del laboratorio de TEE-AEPM que está dentro de la ESFOT para que posteriormente los estudiantes de las carreras tecnológicas no se vean afectados, ya que al ocurrir lo antes mencionado a veces no se les realiza el préstamo de los equipos. En base a esta necesidad se planteó realizar un sistema de inventario para equipos de laboratorio de TEE-AEPM, en el que se creó un dispositivo que detecte la entrada y salida de los equipos, y a su vez almacene información en una base de datos gestionada mediante una aplicación.

La realización de este proyecto ayuda a los docentes y estudiantes a facilitar el préstamo de los equipos que se encuentra en el laboratorio de TEE-AEPM, ya que el proyecto cuenta con un sistema web para el registro de estudiantes y equipos. Cada equipo cuenta con etiquetas de identificación para facilitar el préstamo y el estudiante hace uso de su huella dactilar para realizar el préstamo evitándose molestias por retención de la cédula y llenar un formulario de préstamos, los cuales están expuestos a pérdidas.

Para la devolución de los equipos el estudiante deberá colocar simplemente el tag de corta distancia en el módulo RFID-RC522 y se procederá a la entrega del equipo prestado evitándose así llenar un formulario de devoluciones.

Este sistema es de fácil utilización, ya que, la persona que requiera un equipo deberá primero registrarse en el sistema con la ayuda de un encargado de laboratorio para guardar sus datos personales, posterior a eso, se podrá realizar el préstamo. El préstamo

comienza pasando el equipo por el lector RFID-RC522 y el usuario colocando su huella dactilar para realizar el registro de préstamo, seguidamente se envía una notificación mediante correo electrónico donde se recuerda la información del equipo prestado con su fecha máxima de devolución. Para la devolución del equipo el usuario acercará el tag de corta distancia al lector RFID-RC522 y se registrará la entrega del equipo, misma que se notificará mediante correo electrónico.

1.1. Objetivo general

Implementar un sistema de inventario para equipos de laboratorio de “Tecnología Eléctrica y Electrónica – Área de Electrónica de Potencia y Microprocesadores”.

1.2. Objetivos específicos

- Especificar requerimientos de hardware y software para el sistema.
- Realizar los diseños de hardware y software para el sistema.
- Desarrollar software para el sistema.
- Verificar el funcionamiento de hardware y software del sistema.

1.3. Justificación

El *software* para el control de inventarios facilitó el proceso en las actividades para los usuarios del laboratorio de TEE – AEPM. Dentro de este proyecto los beneficiados fueron principalmente el personal que ahí laboran (usuarios), quienes otorgan atención y servicio de abastecimiento tanto de elementos electrónicos, equipos, entre otros; los cuales contaron con un servicio de atención rápido, eficiente y oportuno para cubrir las necesidades.

El *hardware* está compuesto por la tecnología RFID y un control de registro mediante un lector de huella digital que ayudó al proceso de automatización y seguridad de varios activos, esta tecnología permitió identificar inalámbricamente por frecuencias las etiquetas que se encuentran en los equipos. Las etiquetas contienen información que puede ser manipulada en diversas distancias dentro de una misma área, esta tecnología permitió leer los equipos cuando ingresan o salen del laboratorio y son más seguras, ya que no pueden ser clonadas fácilmente e incluso brindan más beneficios al poder ser reescritas varias veces. Los usuarios que deseen llevarse el equipo deberán registrarse

mediante un lector de huella digital antes de realizar un préstamo para asegurar que el usuario se lleva el equipo y realizar la devolución con éxito.

1.4. Marco teórico

Metodología *Scrum*

Para el desarrollo del *software* se utilizó una metodología *SCRUM* que permitió obtener los mejores resultados aplicando un conjunto de buenas prácticas y a la vez entregas parciales del producto final. [1]

Componentes y marco *Scrum*

Roles Centrales

Los roles centrales son aquellos que su participación es indispensable para la realización del proyecto, están comprometidos con el proyecto y son responsables del éxito de cada *sprint* y del proyecto en general, [2] estos son:

- ***Product owner* (Dueño del producto)**

Su principal misión es encargarse de que exista una priorización clara de los objetivos a conseguir, con el propósito de maximizar el valor del trabajo que lleva a cabo el equipo. [3]

- ***Scrum máster* (Maestro *Scrum*)**

El *Scrum Máster* tendría una figura similar a la de un *coach*/mentor que acompañará al equipo durante todo el desarrollo del proyecto y asegurará que se cumplan las buenas prácticas, actuando como un facilitador y solucionador de problemas. [4]

- ***Development team* (Grupo de desarrollo)**

Se refiere al conjunto de personas más “técnicas” que de manera conjunta desarrollan el producto del proyecto. Tienen un objetivo común, comparten la responsabilidad del trabajo que realizan (así como de su calidad) en cada iteración y en el proyecto. [5]

Artefactos

Los artefactos de *Scrum* son herramientas de apoyo que ayudan al equipo a lo largo del proyecto, esto es:

- ***Sprint Backlog***

Es el conjunto de elementos de *Product Backlog* seleccionados para el *Sprint*, más un plan para entregar el incremento del producto y realizar el *Sprint Goal*. [6]

- ***Sprint Retrospective***

Retroalimentación es el último evento de un *sprint* dentro de la metodología *Scrum* que permite inspeccionar el trabajo y crear un plan de mejora que se pondrá en marcha en el siguiente *sprint*.

Herramientas de desarrollo web

XAMPP

Es un servidor independiente multiplataforma, de *software* libre, que consiste principalmente en la base de datos *MySQL*, el servidor *web* Apache y los intérpretes para lenguajes de *script*: PHP y *Perl*. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, *MySQL*, PHP, *Perl*. [7]

MySQL

MySQL sirve para almacenar toda la información que se desee en bases de datos relacionales, como también para administrar todos estos datos sin apenas complicaciones gracias a su interfaz visual y a todas las opciones y herramientas de las que dispone. [8]

Laravel

El objetivo de *Laravel* es el de ser un *framework* que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el “código espagueti” y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros *frameworks* y utiliza las características de las últimas versiones de PHP. [9]

Bootstrap

Bootstrap es un *framework* (librerías de CSS) que facilita y estandariza el desarrollo de sitios *web*. [10]

HTML

HTML es el lenguaje con el que se define el contenido de las páginas *web*. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página *web*, como imágenes, listas, vídeos, etc. [11]

CSS

Es una herramienta que permite crear páginas *web* de una manera más rápida. Gracias a las CSS se puede ser más dueños de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, etc. [12]

JavaScript

Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página *web* y en programas más grandes, orientados a objetos mucho más complejos. Con *Javascript* se puede crear diferentes efectos e interactuar con los usuarios. [13]

2. METODOLOGÍA

2.1. Metodología general del proyecto

Para realizar el sistema de inventario se comenzó dialogando con la persona encargada del laboratorio y se tomaron en cuenta las ventajas y desventajas de aplicar sistemas de inventarios. Con ello se tomó las decisiones referentes a la infraestructura que se elaboró y a la vez se determinó varios aspectos entre ellos: materiales, accesorios necesarios para la implementación, mecanismos de instalación y conexiones eléctricas para el sistema. También se revisó los requerimientos para el desarrollo del sistema *web* utilizando las técnicas y herramientas de *software* adecuadas.

El diseño del sistema fue realizado en base a los esquemas de las figuras 2.1, 2.2 y 3.8.

Para el diseño del sistema se determinó una lista de los instrumentos y elementos necesarios para su posterior implementación, lo cual contó principalmente con una tecnología RFID (Identificación por radio frecuencia). También se procedió a diagramar los planos eléctricos y electrónicos de las conexiones para una mejor visualización de los

dispositivos y áreas con medidas reales, dentro del diagrama se desarrolló un esquema lógico del funcionamiento del sistema para su posterior comunicación de los equipos con la base de datos mediante la utilización de una interfaz y de etiquetas que cuentan con la misma tecnología para que la comunicación sea inalámbrica. [14]

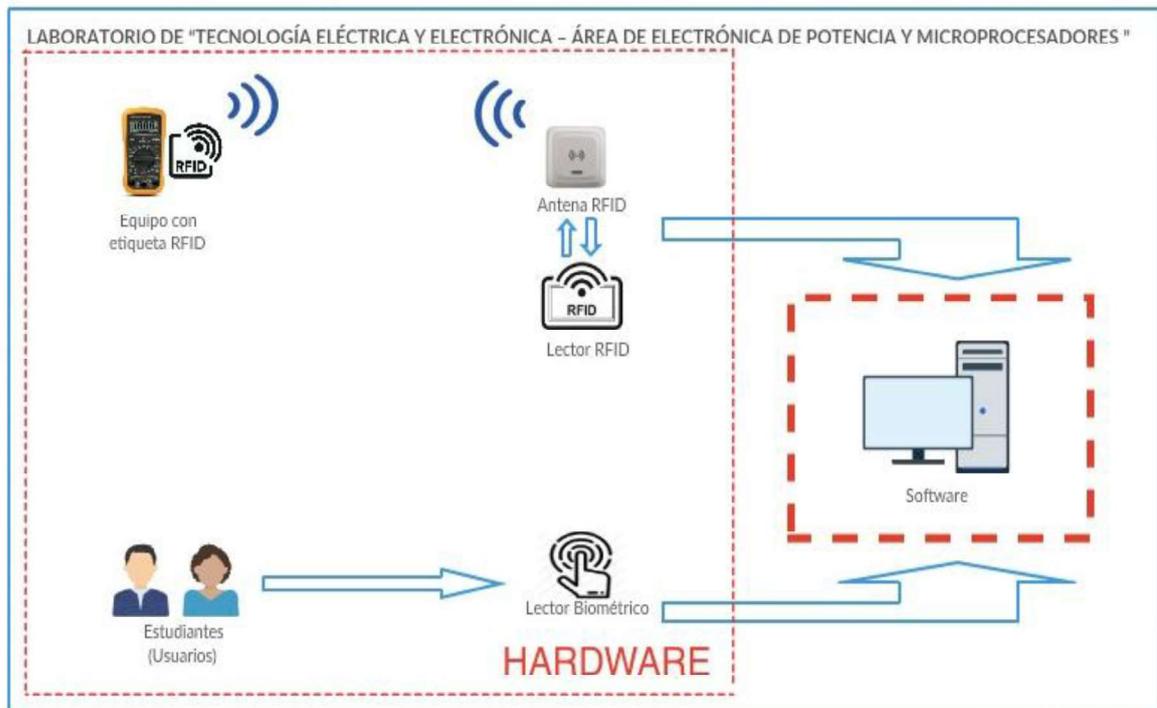


Figura 2.1 Diseño básico de *hardware* y *software*.

Para la implementación de *hardware* se utilizó la tecnología RFID o identificación por radio frecuencia, que permitió que los equipos del laboratorio sean identificados de manera inalámbrica, utilizando los datos que fueron transmitidos desde etiquetas colocadas en los equipos a través de ondas de radio. Para la correcta y veraz transmisión de la información se necesitó seleccionar minuciosamente una antena, ya que esta fue el cerebro del sistema de una variedad de ellas. En este proyecto se utilizó una antena RFID fija para leer y codificar las etiquetas. Además, se realizó un análisis de los pines de conexión disponibles en la antena, ya que mediante ellos se realizó la conexión hacia la base de datos y a su vez se analizó la distancia y área de cobertura. La mayoría de las antenas tenían un cable *Ethernet* para enviar y recibir los datos, utilizando alimentación de CA o a través de un adaptador *Power-over-Ethernet* (POE), que era direccionable por IP y que actúa simultáneamente como cable de alimentación y de comunicación de datos, también había antenas USB que eran más pequeñas, dando la posibilidad de conectar directamente a una PC, aunque éstos no eran direccionables. En base a todas estas

opciones se realizó un análisis de la antena que más ventajas brindó al proyecto siendo usada la antena con pines de conexión para datos *wiegand* y una alimentación de 12V. [15]

Para que el sistema funcione correctamente se configuró la antena, se realizó varias pruebas y se realizó la sincronización con los demás elementos. La antena se puede programar para leer todas o solo ciertas etiquetas, para leer etiquetas en momentos determinados, leer etiquetas una vez en modo inventario o leer etiquetas una y otra vez para obtener una información específica, siendo la última la más conveniente para este proyecto ya que se pudo obtener los datos de los equipos que ingresan y salen. [15]

La antena RFID se encargó de enviar la energía de radio frecuencia a las etiquetas para energizarlas y posteriormente obtener la respuesta de la etiqueta, con la información del equipo que la posee, para ello se necesitó de cables que interconecten la antena con el módulo Arduino y el módulo *Ethernet Shield* que fue por donde se envió y recibió la información y los diferentes comandos. Los cables se seleccionaron en términos de conectores, longitud y en base a su clasificación de aislamiento, ya que si no eran los correctos disminuía la posibilidad de reconocer las etiquetas. La antena tuvo una diferente funcionalidad dependiendo del tamaño, la ganancia, clasificación de IP, polarización y tipo de conector por ende se seleccionó correctamente estos aspectos. Lo que se buscó fue una ganancia alta para que la antena sea más potente y tenga un rango de lectura amplio y confiable. La polarización se seleccionó entre lineal o circular, analizando que pueda captar todas las etiquetas y que la energía se divida para los dos ejes. También se seleccionó la potencia de la antena de acuerdo con el tipo de sistema que se desarrolló. [15]

Se determinó un lugar específico para la colocación de etiquetas de forma que estas no sean tan visibles y que no puedan ser manipuladas fácilmente de los equipos del laboratorio, considerando su modo de funcionamiento y operación. Además, se seleccionó el rango de frecuencia adecuado para el alcance deseado, el cual también fue compatible con la antena RFID. Previamente, se revisó el material del equipo donde se colocó la etiqueta ya que de este dependió el tipo de etiqueta que lleva cada uno. Otra de las características que se tomó en cuenta fue si se usaría una etiqueta activa o pasiva para con ello ver si necesitaba fuente de alimentación interna o externa. Al contar con diferentes memorias para sus funcionamientos como identificadores únicos y almacenamiento de información se puede configurar y cifrar de diferentes formas, las etiquetas estuvieron orientadas hacia la antena dependiendo el tipo de polarización de esta. [15]

La comunicación de la antena con las etiquetas se basó en principios científicos como son la física y el electromagnetismo, el circuito de la etiqueta y el circuito de la antena fueron acoplados o en otras palabras se transfirió energía entre ellas y se determinó el rango de lectura y frecuencia del sistema. [15]

Para el control de información de los usuarios prestamistas se utilizó un lector de huella digital que también sirvió como control del préstamo de los equipos. Los lectores de huellas digitales están diseñados para conseguir una imagen de cada huella digital mediante un escáner de luces led lo cual es enviado por código binario a una BDD que está programada y configurada para posteriormente utilizarlo con una aplicación *web*. De esa manera se pudo tener un control de las personas que utilizan los equipos. Para el almacenamiento de la información de las huellas de los estudiantes se realizó una tabla con el nombre de estudiantes dentro de la base de datos que lleva de nombre Inventario ESFOT, que a su vez almacenará datos de los estudiantes. [16]

Para la realización de la aplicación *web* se remitió a las historias de usuarios en donde se describen los módulos tales como: módulo de usuarios, módulo equipo, módulo estudiante, módulo de préstamos y devoluciones. Los módulos indicados anteriormente se realizaron de la manera que sea de fácil uso para el usuario utilizando HTML y CSS para los estilos de la aplicación *web*.

Respecto al control de prestaciones, en el sistema se implementó una opción de notificación al usuario mediante correo electrónico para préstamos y devoluciones del equipo. Para esto se utilizó una *Application Programming Interface* (API) que agregó un evento a través del cliente PHP. Para el envío de las notificaciones de correos electrónicos se implementó la API de *gmail* la cual envía mensajes de forma gratuita.

Finalmente, una vez realizada la implementación de los instrumentos, las conexiones eléctricas, la elaboración del *software*, la aplicación y su respectiva instalación, se procedió a realizar varias pruebas con las posibles circunstancias que podrían ocurrir verificando que el sistema trabaje correctamente al detectar las etiquetas en los equipos. Mediante los resultados de las pruebas realizadas al sistema de inventario se corrigió y modificó ciertos aspectos para obtener un mejor funcionamiento. Cuando el sistema pasó las pruebas correspondientes y estas fueron satisfactorias, se procedió a realizar un instructivo para el funcionamiento del sistema, el cual brindó la posibilidad de ingresar nuevos equipos y usuarios.

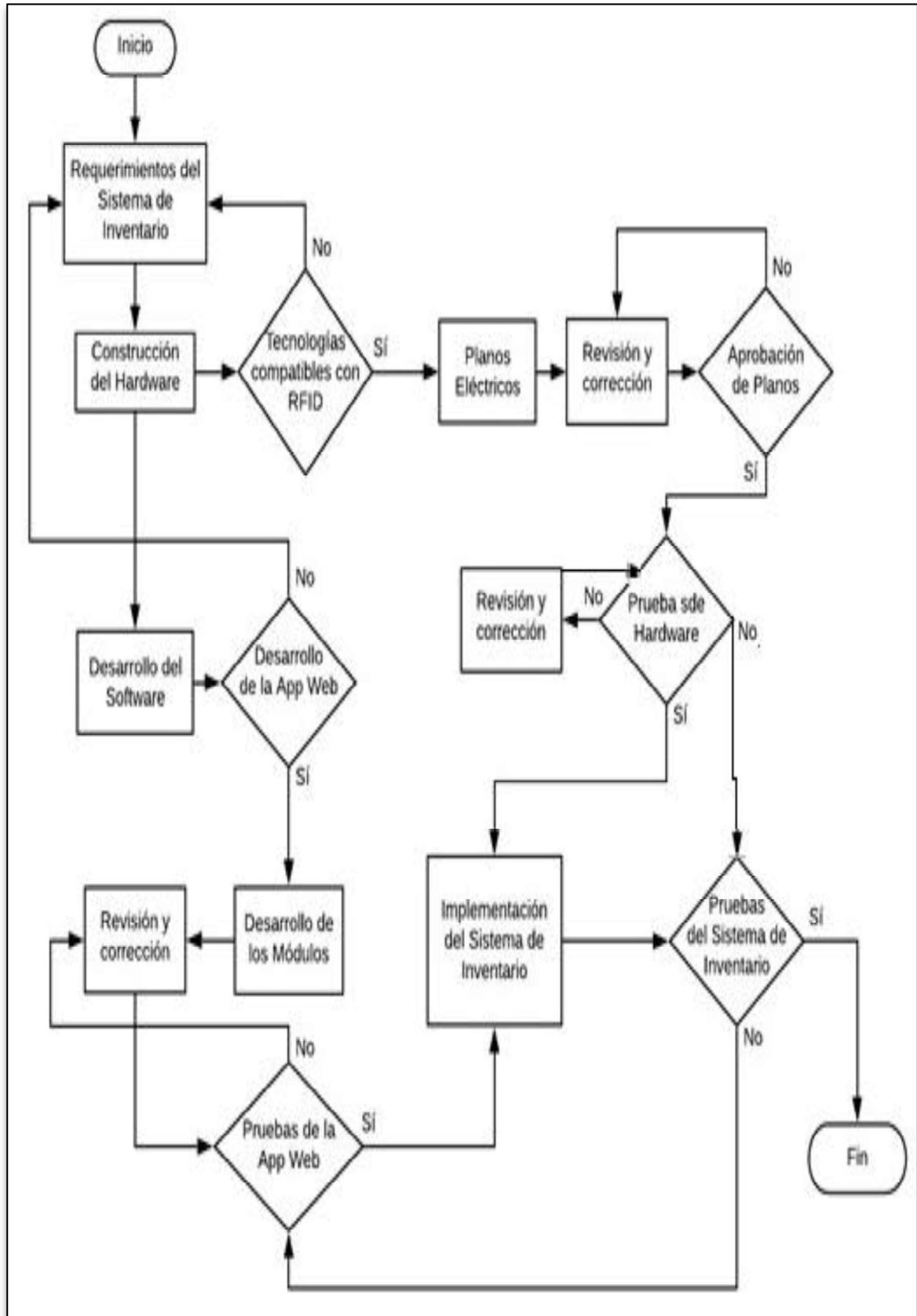


Figura 2.2 Diagrama de flujo del proceso seguido para realizar el proyecto.

2.2. Desarrollo de la metodología *Scrum*

Para el desarrollo de la metodología *Scrum* se eligió un responsable del proyecto que debe tener en claro la idea que se va a desarrollar, el equipo de trabajo debe contar con habilidades necesarias para poder llevar a cabo el desarrollo del sistema, dentro del equipo de trabajo se eligió a la Ing. Viviana Párraga como *Scrum master* y con ella se acordó los objetivos y la planificación del desarrollo del sistema. Con el *Scrum master* se organizó las fechas de entrega y revisión de los avances del proyecto y si fuese el caso realizar una retroalimentación.

Conformación Equipo *Scrum*

La tabla 2.1 detalla las personas y entidades que formaron parte para el desarrollo de la aplicación *web* con sus respectivos roles.

Tabla 2.1 Equipo de trabajo y asignación de roles

Personas/Entidades	Rol
Ing. Mónica Vinueza	<i>Product Owner</i>
Ing. Viviana Párraga	<i>Scrum Máster</i>
Chicaiza Diego	<i>Development Team</i>
Lagua Angela	<i>Development Team</i>

Historias de usuario

Las historias de usuario detallan los requerimientos que facilitaron el desarrollo del sistema *web*. Todo esto se presenta en el anexo I.

3. RESULTADOS Y DISCUSIÓN

3.1. Requerimientos

Para el desarrollo del proyecto principalmente se determinó las necesidades y requerimientos, solicitando información a la persona encargada del laboratorio, la cual manifestó que los equipos que más solicitan son los multímetros y fuentes DC, mismos que se permite a los estudiantes sacar del laboratorio por un par de horas, pero dejando sus datos y cédula de identidad, lo que dificulta al estudiante realizar otras actividades donde necesiten usar estos. Los equipos que son prestados dentro del laboratorio solo pueden ser usados por un momento debido a que ahí se dictan clases y si el equipo es prestado para fuera del laboratorio nadie les asegura a los encargados del laboratorio

que después sean devueltos, por motivos incluso de algún olvido en el registro. En base a esta información se estableció los siguientes requerimientos para el proyecto:

- Registro de estudiantes.
- Registro de los equipos.
- Base datos.
- Interfaz gráfica.
- Sistema de seguridad.
- Préstamo de equipos.
- Devolución de equipos.
- Notificación de las actividades.
- Información remota de la base de datos.
- Ubicación estratégica de la antena.

Registro de estudiantes

Debido a que es necesario saber los datos personales tanto como los nombres, apellidos, número de cédula de identidad, la carrera a la que pertenece, entre otros datos de las personas que solicitaron algún equipo en el laboratorio; por lo que se consideró para el proyecto realizar esto mediante un lector de huella digital.

Registro de los equipos

Puesto que es necesario saber cuántos, qué equipos son los que se permite prestar, en qué estado se encuentran y si salen del laboratorio con una autorización o no; para ello se utilizó dos tipos de *Tags* RFID uno de corta distancia y otro de gama UHF.

Base de datos

Dado que, es necesario contar con un registro de los equipos que son prestados y a qué personas se entregan los mismos, para evitar pérdidas u olvidos, se recurrió a utilizar una base de datos ya que todo esto se puede observar con sus respectivas fechas y actividades tanto de préstamos como devoluciones. En el proyecto se manejó una base de datos basada en *MySQL*.

Interfaz gráfica

De acuerdo con los requerimientos es necesario una interfaz gráfica donde la persona encargada de laboratorio interactúe de forma rápida y eficiente al realizar los registros de usuarios como de equipos, préstamos, devoluciones y al verificar alguna observación.

Sistema de seguridad

A causa de que, en el laboratorio se cuenta con dos áreas tanto el Área de Electrónica de Potencia que es donde se encuentran los equipos y el Área de Microprocesadores que es el lugar donde pasa la encargada de laboratorio fue necesaria la utilización de una antena UHF que permita identificar con la ayuda de *Tags* y un zumbador electrónico cuando un equipo sea sustraído o salga sin autorización.

Préstamo de equipos

En vista de que, los estudiantes solicitan los equipos para realizar prácticas o pruebas de algún proyecto fuera del laboratorio, se utiliza la huella digital y el *tag* RFID de corta distancia para que se registre al estudiante que solicita el préstamo y saber que equipo fue entregado.

Devolución de equipos

Debido a que, se utilizan para dictar clases, esta acción también es registrada ya que en ocasiones se puede llegar a olvidar y sería la pérdida de un activo para la ESFOT. La devolución del equipo se realiza mediante el tag de corta distancia para verificar que el equipo se devuelve de la misma manera en la que se prestó.

Notificación de las actividades

No es necesario presentar la cédula de identidad al momento del préstamo o devolución de los equipos ya que, se tendrá un respaldo de cuándo se prestó el equipo y cuándo es devuelto mediante una notificación de correo electrónico a la persona que realiza dicha actividad, en el correo se informa la fecha del préstamo o devolución, el nombre del equipo, la persona que realiza la actividad y en el caso de préstamo la fecha máxima de devolución.

Información remota de la base de datos

De acuerdo con los requerimientos del proyecto fue necesario subir la aplicación *web* a un servidor con su respectiva base de datos para que la información se encuentre disponible desde cualquier computador con acceso a Internet y con ello poder interactuar con el mismo.

Ubicación estratégica de la antena

La antena se maneja mediante un campo de ondas de frecuencia por lo que, se vio necesario realizar varias pruebas para seleccionar su mejor ubicación, buscando una

óptima detección y menor interferencia con los equipos al trasladarse de un área del laboratorio a otra.

3.2. Selección de dispositivos

Basándose en los requerimientos del proyecto, los componentes electrónicos seleccionados se encuentran a continuación:

Arduino uno

Placa Arduino Uno de 32 puertos entre digitales y analógicos, cuenta con un microcontrolador atmega, entradas para alimentación externa o por USB entre otros aspectos que se pueden observar en las figuras 3.1 y 3.2. Este componente ha sido elegido entre otros por su bajo costo, por la compatibilidad que tiene con la mayoría de las *shields* y por la comunicación SPI; en otros componentes como por ejemplo Arduino Leonardo no se podría utilizar esta comunicación si a su vez se necesita conectar a él un *shield*, debido a que la placa Arduino Leonardo utiliza los pines del conector ICSP (*In Circuit Serial Programming*) para la comunicación SPI y el *shield* en ocasiones no cuenta con un conector de seis pines para ICSP [17].

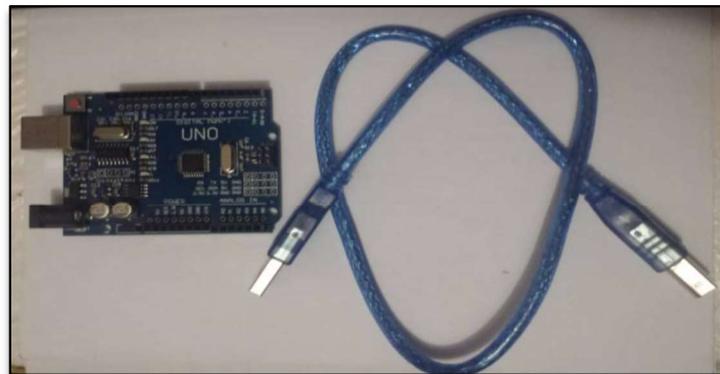


Figura 3.1 Arduino UNO y cable USB tipo A-B.

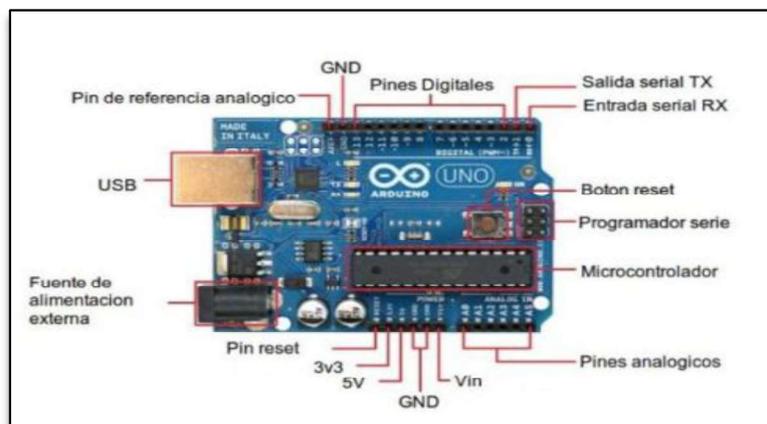


Figura 3.2 Partes del Arduino UNO. [17]

Ethernet Shield

El módulo *Ethernet Shield* es muy similar a la placa de Arduino Uno y son compatibles ya que fue creada para añadir ciertas características a Arduino principalmente que mediante su puerto RJ45 brinda conectividad hacia la red, véase figura 3.3.

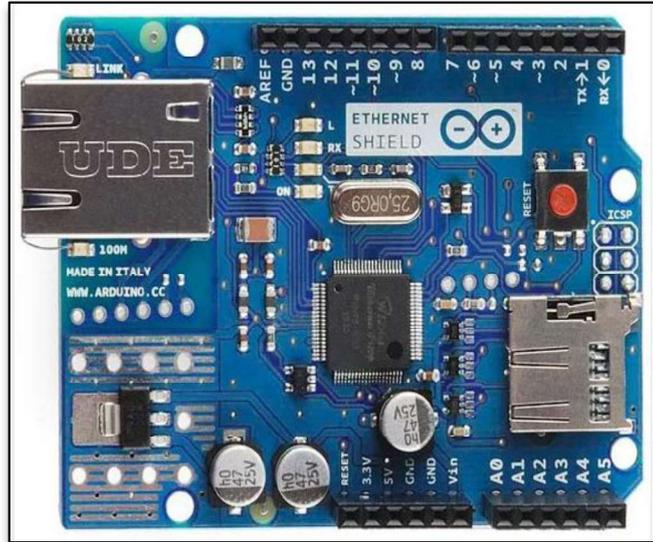


Figura 3.3 *Ethernet Shield*. [18]

Módulo RFID-RC522

Este módulo es utilizado juntamente con Arduino para poder detectar los *tags* RFID de 13,56 MHz que cuenta con un NFC que es el estándar general de comunicación de campo cercano, véase figura 3.4. Este módulo fue elegido por su bajo costo y por la necesidad de su principal característica que es la lectura de *tags* a corta distancia.



Figura 3.4 Módulo RFID-522 y Tag RFID.

Lector de huella digital

Equipo utilizado con Arduino encargado de tomar capturas de las huellas digitales y procesarlas para después enviar información de forma binaria, donde se interpreta en 1's y 0's en las líneas de huella del dedo y sus espacios. Este lector se utilizó por su bajo costo, pequeño tamaño, bajo peso y la facilidad de conexión al Arduino ya que es necesaria una alimentación de máximo 6V y 100mA cuyos valores son brindados directamente de la placa Arduino mientras al utilizar otros lectores se necesita fuentes externas de alimentación de energía que trabajan con 12V y de 1 a 3 mA, véase figura 3.5.

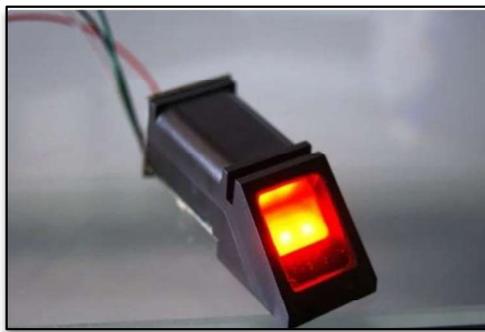


Figura 3.5 Lector de huella digital para Arduino. [19]

Antena UHF

Equipo utilizado con variedad de dispositivos, en este caso con los módulos Arduino y *Ethernet Shield*, cuenta con cables de datos wiegand, con una frecuencia de 902 a 928 MHz, con una ganancia de 8 dBi, no necesita línea de vista como otros equipos ya que posee una polarización circular que permite identificar los *tags* a diferentes alturas y puede leer las etiquetas una y otra vez pudiendo ser estas reutilizadas, véase figura 3.6.



Figura 3.6 Antena RFID de gama UHF.

Zumbador electrónico

Bocina que interactúa con Arduino con una alimentación de 5 Voltios, aunque esta se la puede utilizar con una alimentación que va desde los 3 a 24 Voltios. Esta bocina es utilizada por su bajo costo, pequeño tamaño y por el volumen moderado del sonido que emite para que pueda alertar al encargado del laboratorio sin interrumpir las clases que se dictan, véase figura 3.7.

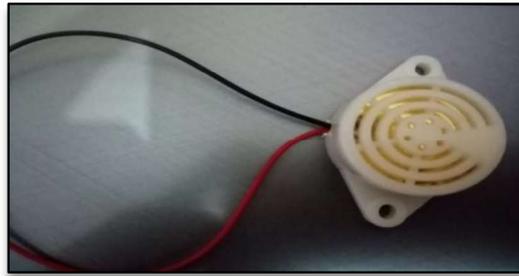


Figura 3.7 Zumbador electrónico SFM-27 DC 3 - 24V.

3.3. Funcionamiento del sistema de inventario

El sistema realiza préstamo de equipos cuando estén registrados tanto el usuario y el equipo a prestarse, los cuales cuentan con la etiqueta RFID. La antena RFID envía la energía de radiofrecuencia a las etiquetas para energizarlas y transmitir la información, luego al modular esa señal se retorna y escucha su respuesta. La antena RFID conectada al sistema funciona como medio de comunicación para enviar y recibir información que es almacenada en una base de datos, así como también para activar alertas.[15]

Los usuarios que deseen realizar un préstamo de equipos deben colocar su dedo pulgar derecho en el lector de huella digital donde el sistema indica si es un usuario ya registrado o nuevo. Si es un usuario nuevo, es decir, aún no se ha registrado en el sistema, se solicita sus datos básicos que son llenados en la interfaz de usuario por el administrador del laboratorio y se almacena en la base de datos en conjunto con su huella digital. La base de datos de usuarios se llena conforme se realicen los préstamos. En caso de que la persona ya se encuentre registrada en el sistema simplemente continúa con el proceso de préstamo. A continuación, el usuario pasa el equipo por un módulo lector RFID-RC522 para que se le asigne como prestado el equipo a llevarse. Con los dos parámetros registrados: huella digital y equipo, la etiqueta RFID realiza una comunicación con el sistema, permitiendo pasar por el siguiente lector RFID- UHF colocado en la puerta, sin accionar la alarma, caso contrario y de no cumplir con ambas condiciones al salir el equipo del laboratorio se activa una alarma de prevención que

indica que el equipo está saliendo sin registrar el préstamo, la ubicación de los equipos se puede observar en la figura 3.8.

Para las devoluciones de los equipos se ingresa al laboratorio sin que la alarma se accione, el usuario que pidió prestado el equipo coloca el tag de corta distancia en el módulo RFID-RC522 y posteriormente el equipo vuelve a activarse, el inventario se modifica automáticamente registrando la devolución. El proceso completo se detalla correspondientemente en las figuras 3.9, 3.10, 3.11 y 3.12.

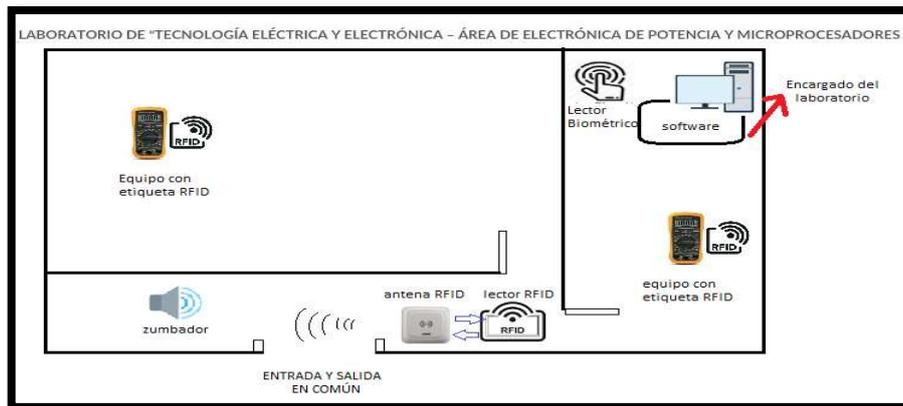


Figura 3.8 Diseño básico del sistema dentro del laboratorio.

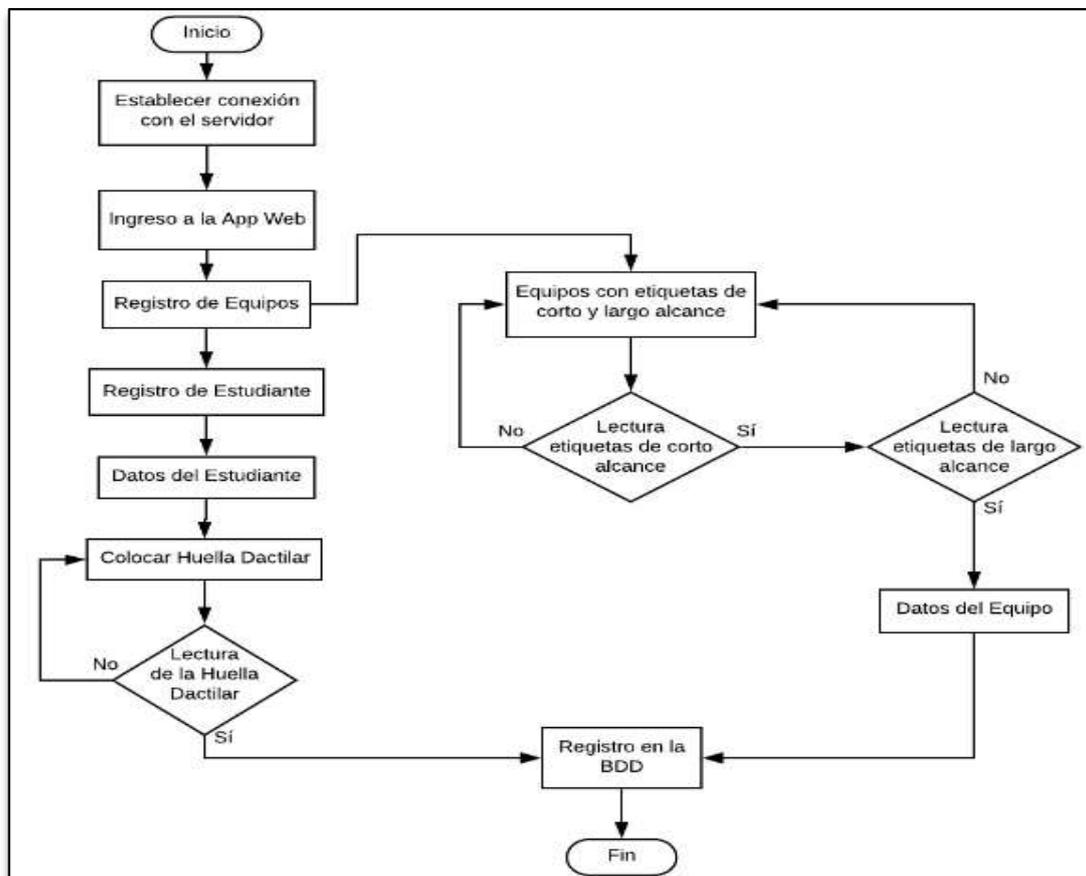


Figura 3.9 Diagrama de flujo de registro de estudiante y equipo.

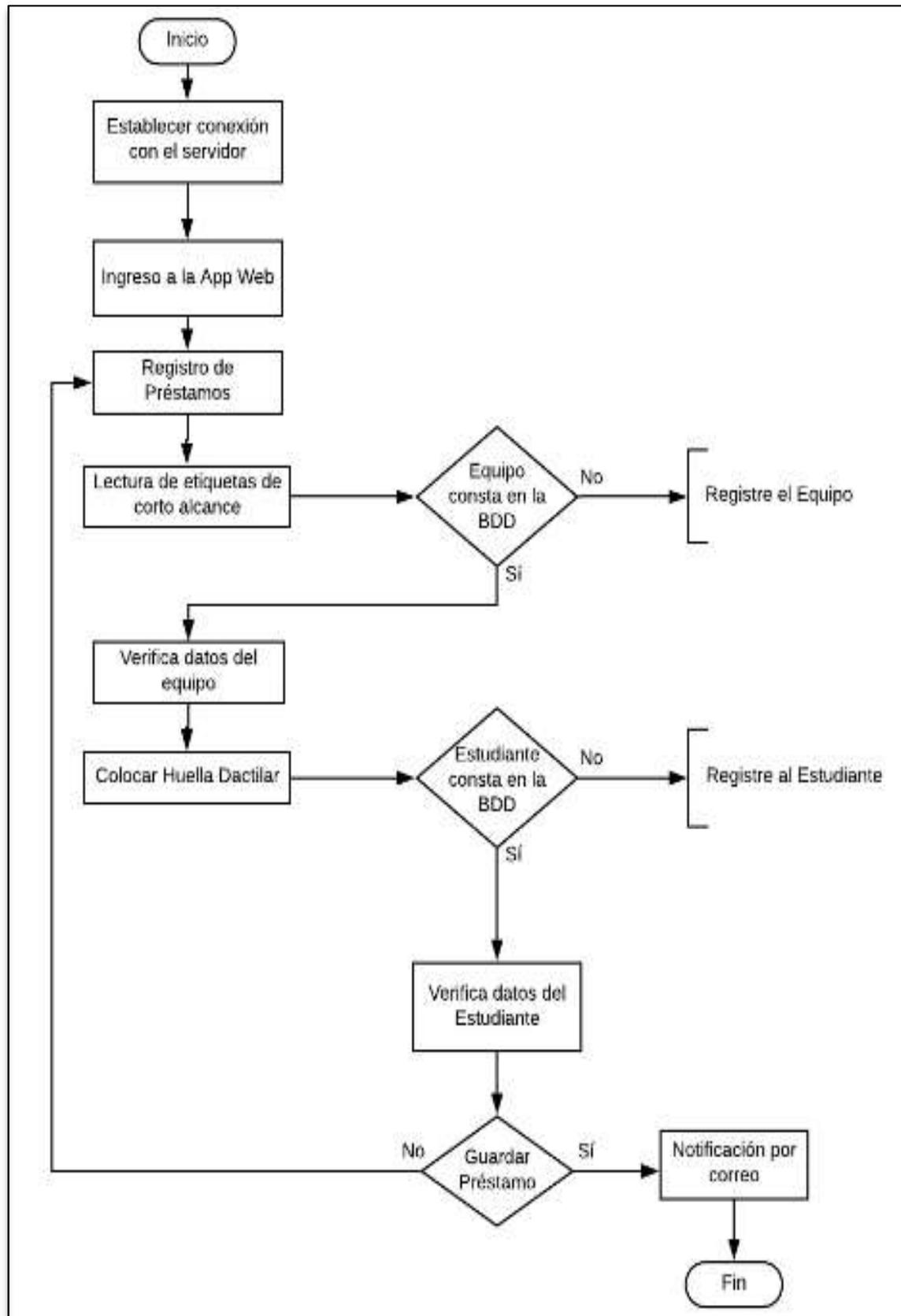


Figura 3.10 Diagrama de flujo de registro de préstamo.

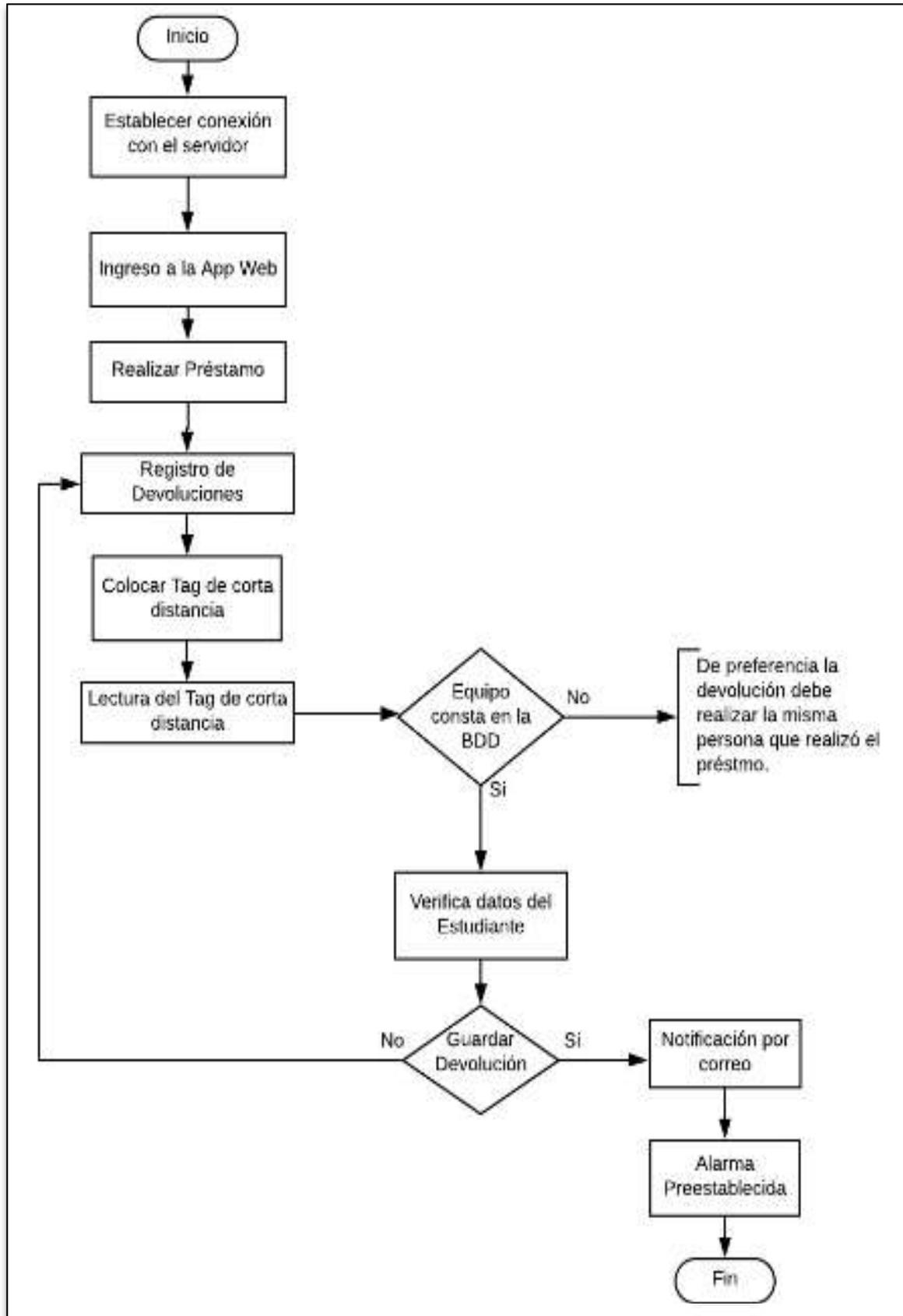


Figura 3.11 Diagrama de flujo de registro de la devolución.

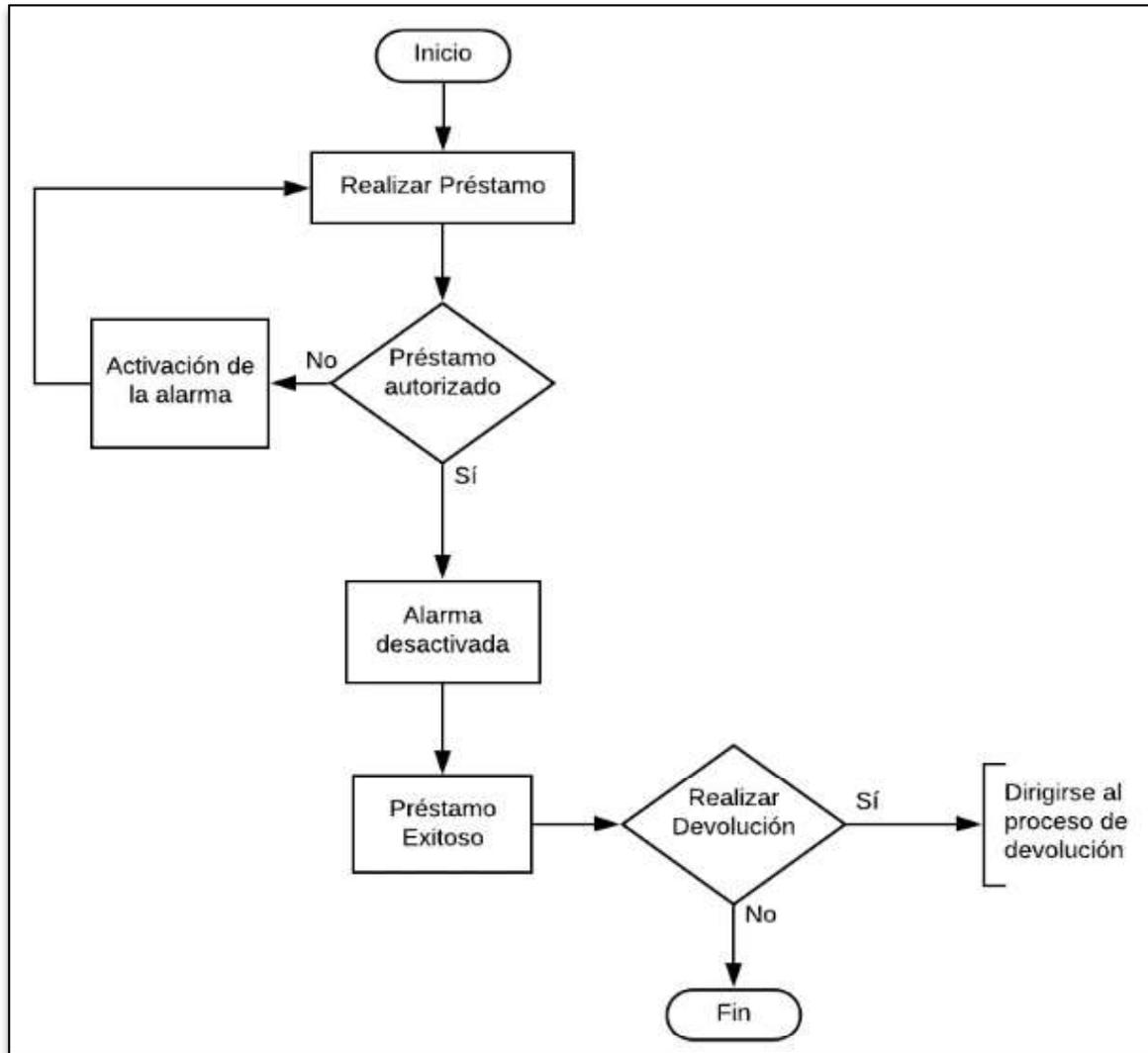


Figura 3.12 Diagrama de flujo del funcionamiento de la alarma.

3.4. Diseño lógico

Los dispositivos que requieren de una conexión con el servidor se han clasificado en Cliente 1 y Cliente 2 debido a que se encuentran en diferentes áreas del laboratorio, pero utilizan la misma red para interactuar con el servidor, se los puede identificar físicamente por las cajas de madera. El cliente 1 se encuentra en el lugar donde se realiza el registro del préstamo, mientras que, el cliente 2 es aquel que se encuentra en la puerta del laboratorio para detectar equipos que salgan sin registrarse.

Partes del proyecto implementado

En los diagramas de bloques de las figuras 3.13, 3.14 y 3.15 se muestran las conexiones del proyecto, indicando tanto las entradas como salidas de los equipos que se comunican entre sí el momento de interactuar con el servidor, como se detalla a continuación:

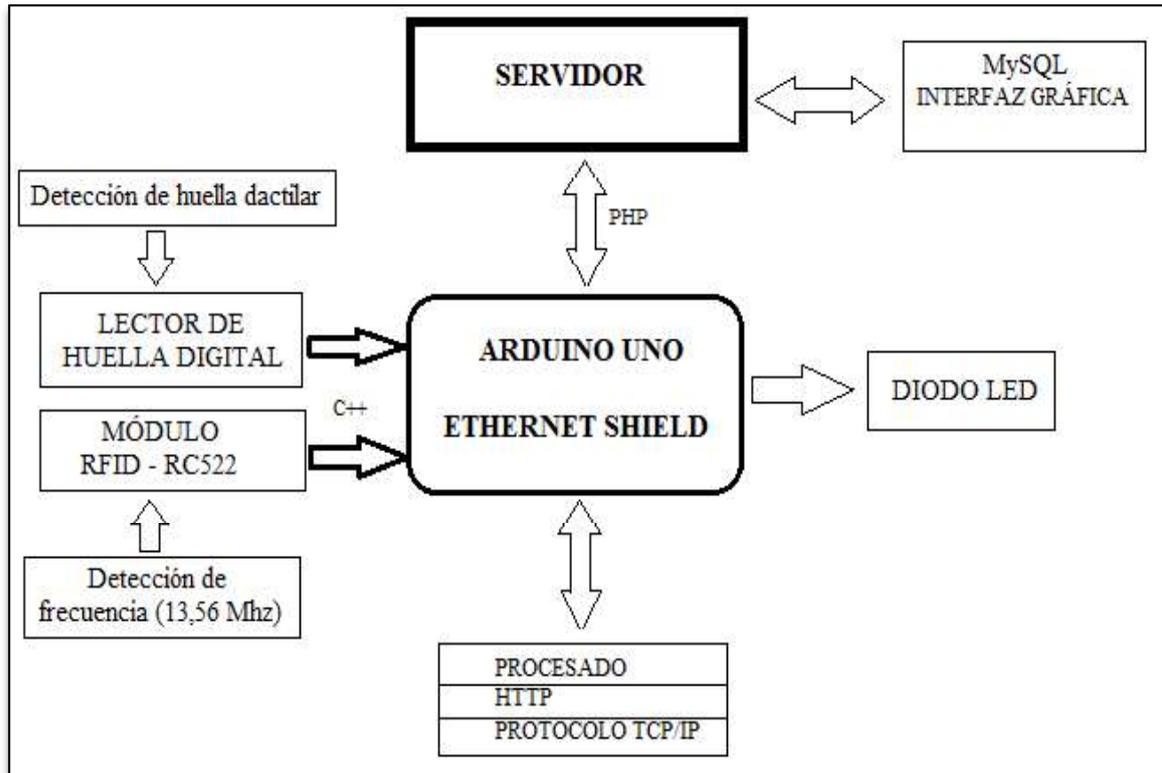


Figura 3.13 Partes y conexiones del proyecto como un cliente 1.

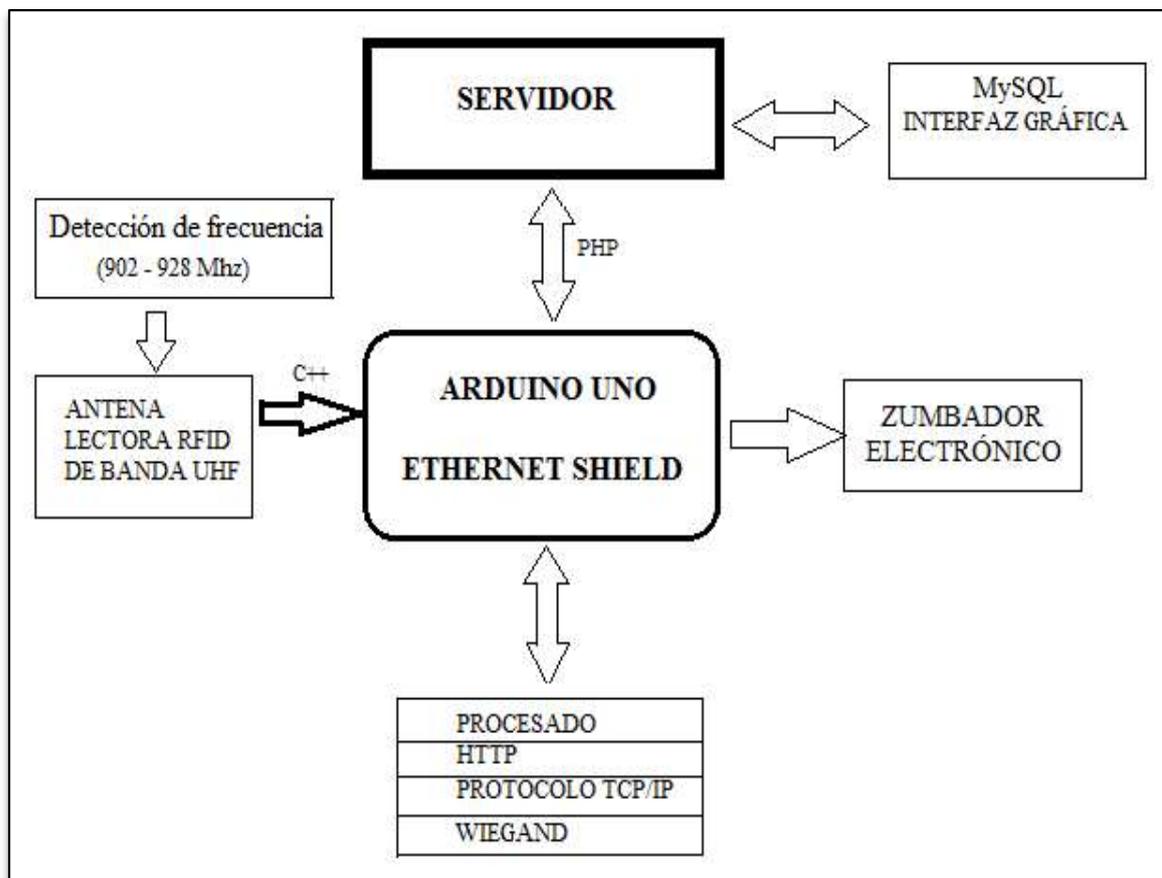


Figura 3.14 Partes y conexiones del proyecto como un cliente 2.

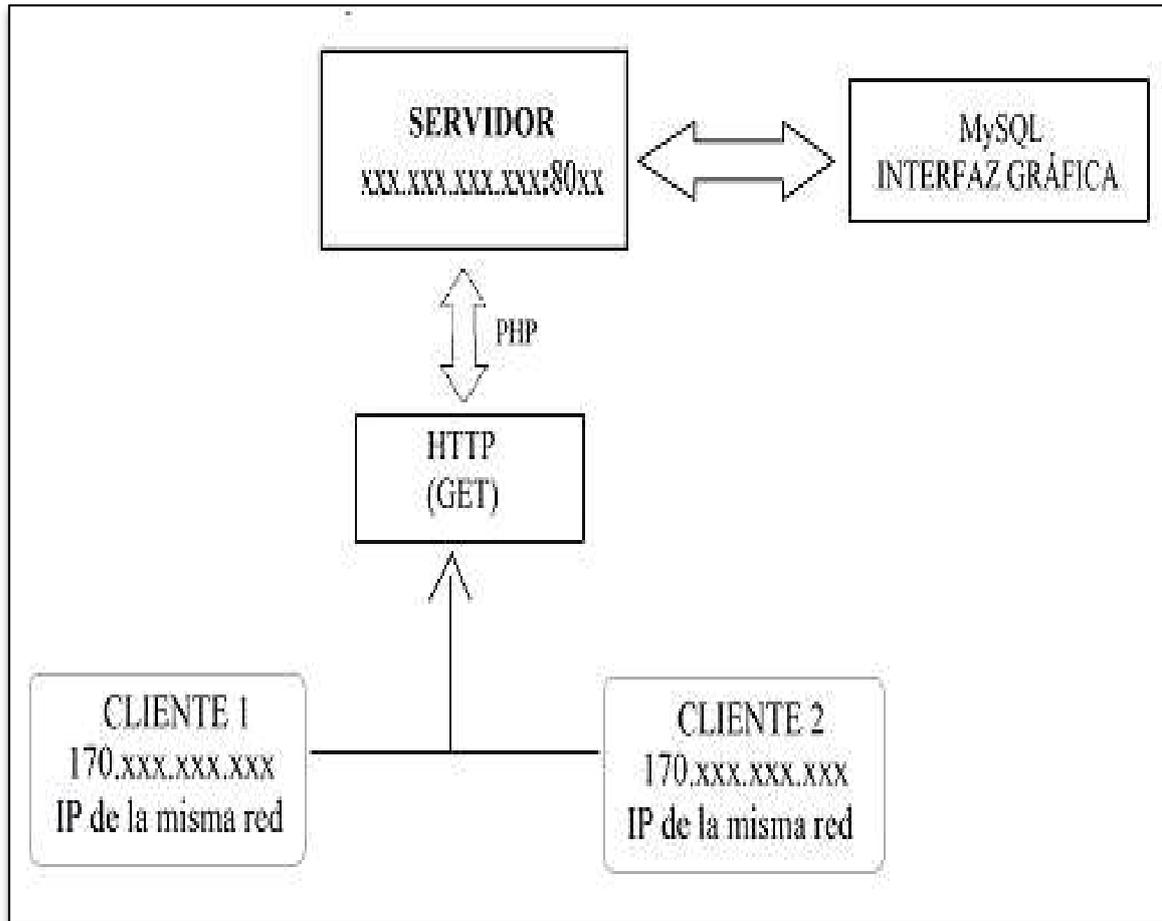


Figura 3.15 Conexión final del proyecto con ambos clientes.

Conexión del lector de huella digital

El lector de huella digital se conectó mediante sus puertos correspondientes de transmisión y recepción a los pines digitales de la placa de Arduino Uno, posteriormente se energizó el lector con un voltaje de 3,3V, que es lo que suministra la placa mediante uno de sus pines y se aterrizó con GND como se observa en la Tabla 3.1. y Figura 3.16.

Tabla 3.1 Conexión de los pines del lector de huella a Arduino Uno.

PLACA DE ARDUINO UNO	LECTOR DE HUELLA DIGITAL
3.3V	3V3
D~2	TX
D~3	RX
GND	GND

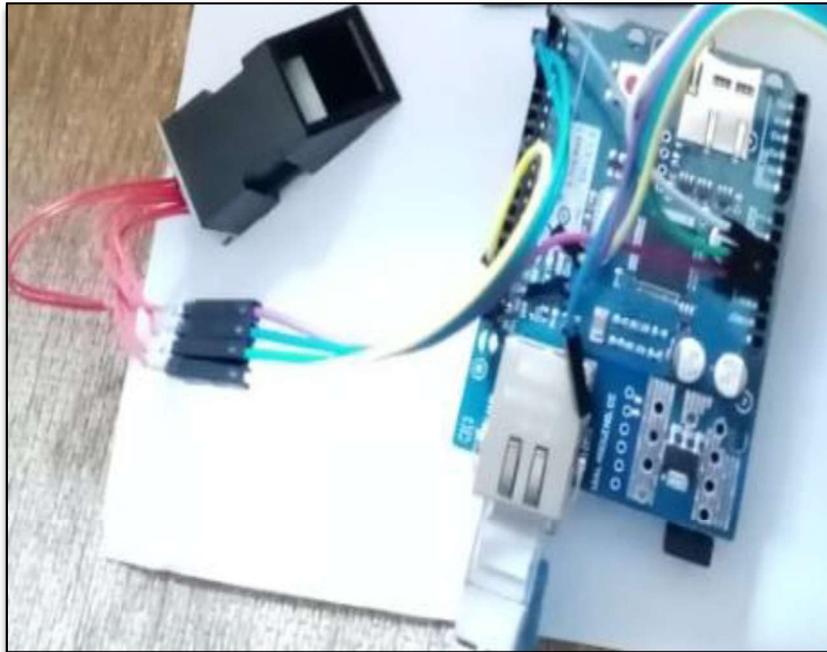


Figura 3.16 Conexión del lector de huella.

Conexión del módulo RFID-RC522

El módulo RFID-RC522 se conectó mediante sus pines correspondientes a los puertos digitales de la placa de Arduino Uno utilizando un protocolo SPI y un sistema de modulación y demodulación para dispositivos pasivos de 13,56Mhz, posteriormente se energizó el módulo RFID con un voltaje de 3,3V, que suministra la placa mediante uno de sus pines y se aterrizó con GND como se observa en la Tabla 3.2 y Figura 3.17. [18]

Los puertos del módulo RFID-RC522 se detallan a continuación: [19]

SDA/SS (*Slave Select*): Es aquella línea por donde el dispositivo maestro habilita al esclavo e indica qué dispositivo es el que tiene que enviar/recibir datos.

SCK/CLK (*Clock*): Línea de reloj que proviene del maestro, la cual es la encargada de enviar un tren de pulsos para sincronizar las comunicaciones entre dispositivos.

MOSI (*Master Out Slave In*): Línea que envía datos de forma serial desde el maestro a los esclavos.

MISO (*Master In Slave Out*): Línea donde los dispositivos esclavos envían información a los maestros.

IRQ (*Interrupt Request Pins*): Línea utilizada para solicitar una interrupción.

Tabla 3.2 Conexión de los pines del lector de huella a Arduino Uno.

PLACA DE ARDUINO UNO	MÓDULO RFID-RC522
D8	SDA
D13	SCK
D11	MOSI
D12	MISO
-	IRQ
GND	GND
9	RST
3.3V	3.3V

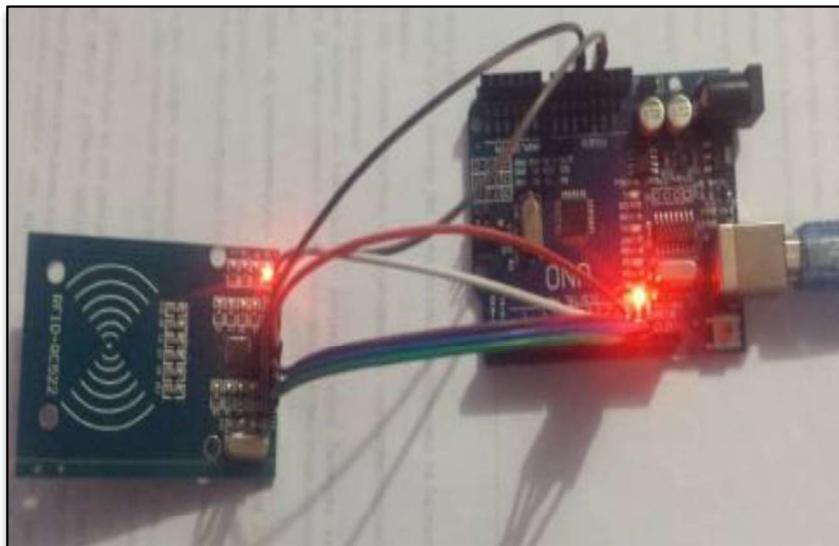


Figura 3.17 Conexión del módulo RFID-RC522.

Conexión de fuente señalizadora de luz

Se conectó un led azul a la placa Arduino Uno, donde el cátodo fue conectado a tierra en uno de los respectivos pines GND del Arduino y el ánodo al pin digital número 5 con una resistencia de protección para el led de 330 ohm como se observa en la figura 3.18.

La placa Arduino Uno emite a la salida de sus puertos un valor de 5V y la corriente aproximada de consumo del led es de 15mA, por ende, el valor de la resistencia para protección del led sería de: $5V/15mA=333,33$ ohm. Un valor real de una resistencia física cercana a 333,33 ohm obtenidos sería de 330 ohm.

Este led es utilizado para que el usuario pueda observar si el *tag* y la huella dactilar han sido captados ya que se enciende al ocurrir ésto y a su vez emite una luz parpadeante cuando el usuario aún no se encuentra en el sistema al colocar su huella dactilar, es decir aún no cuenta con un ID.

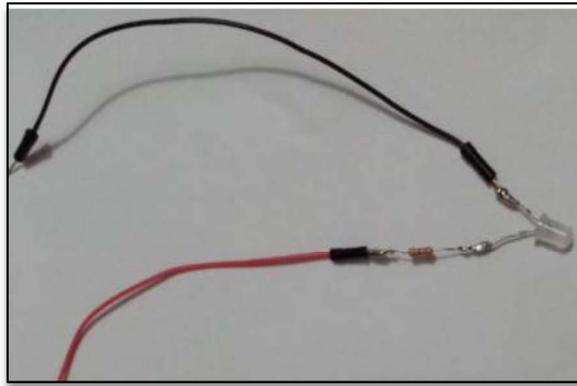


Figura 3.18 Led con su respectiva resistencia de protección.

Conexión del zumbador electrónico

El zumbador electrónico trabaja con un voltaje de 3 a 24V, por lo que, se conectó directamente al pin digital 8 de la placa Arduino Uno y posteriormente se lo puso a tierra mediante el pin GND, igualmente de la placa Arduino como se muestra en la figura 3.19.

El zumbador electrónico se utilizó para brindar una alerta acústica y avisar así al encargado del laboratorio si algún equipo está saliendo sin autorización.

Conexión de la antena RFID UHF

La antena UHF fue conectada a la placa Arduino mediante los puertos digitales y su cableado posterior que se identifica mediante colores como se detalla en la Tabla 3.3 y Figura 3.19, la antena funciona con un voltaje de 12V por lo que se empleó un transformador de 110V a 12V y con una corriente de 3.0A para obtener un correcto funcionamiento.

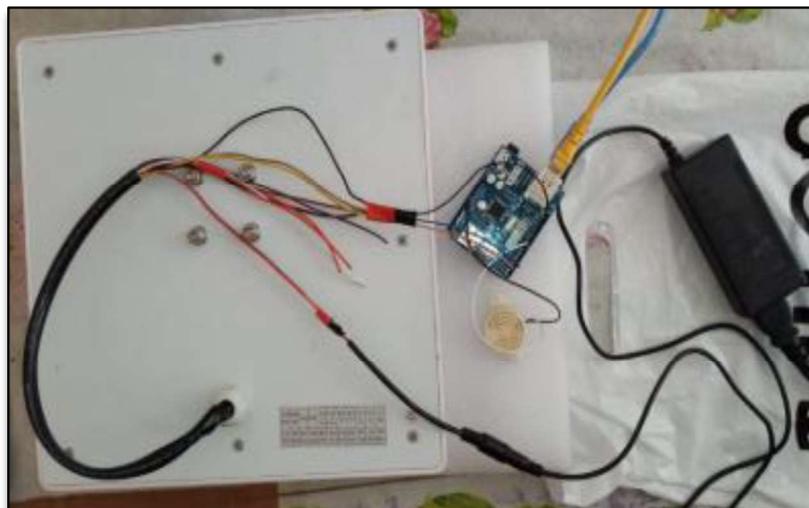


Figura 3.19 Conexión de la antena UHF y zumbador electrónico.

Tabla 3.3 Conexión de la antena UHF.

COLOR DEL CABLEADO DE LA ANTENA UHF	PINES DE LA ANTENA UHF	PLACA ARDUINO UNO
Red	Power	← (Transformador) →
Black	GND	← (Transformador) →
Yellow	WG DATA 0	← D~2 →
Grey	WG DATA 1	← D~3 →
Black	GND	← GND →

Conexión de la placa *ethernet shield*

La placa *ethernet shield* (escudo o mochila) está creada para ser utilizada juntamente con la placa de Arduino Uno, su acoplamiento es simple ya que todos los puertos coinciden y solo es cuestión de colocarla encima para contar con los beneficios, así como se observa en la figura 3.20.

El *ethernet shield* es utilizado tanto para el cliente 1 como para el cliente 2 como se especificaba anteriormente en los diagramas de bloques de la figura 3.13 y figura 3.14 y a su vez no se han mencionado sus puertos ya que son los mismos que en la placa Arduino Uno y lo que sobresale en ella es su puerto RJ45 para conectarse a la red.



Figura 3.20 Conexión de la placa *Ethernet Shield* en la placa Arduino Uno.

3.5. Diseño Físico

Diagrama final de conexiones del proyecto

En la Figura 3.21 se puede observar el diagrama final de conexión del proyecto como cliente 1 y en la Figura 3.22 el diagrama final de conexión del proyecto como cliente 2, en las cuales se detalla su funcionamiento y sus respectivos pines fueron mencionados anteriormente:

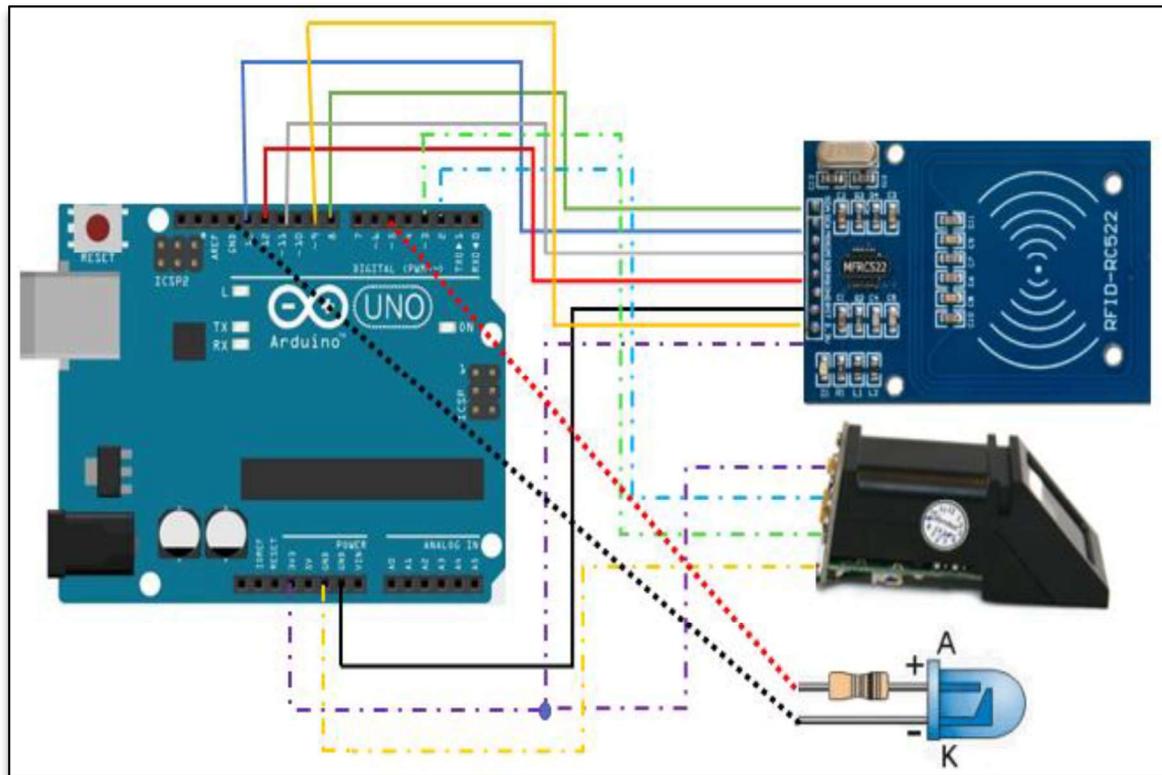


Figura 3.21 Diagrama final de conexión como cliente 1.

En el diagrama de conexión como cliente 1 de la Figura 3.21. debe tener en cuenta que la información captada por el módulo Arduino RFID-RC522 proviene de un *tag*, el cual emite ondas de radio a una frecuencia de 13,56Mhz, al captar esta información el módulo RFID la procesa y la envía a la placa Arduino mediante una interfaz SPI.

La información emitida por el lector de huella digital se da gracias a que internamente realiza un procesamiento digital de imágenes a través de un DSP (procesador de señales digitales), este lector de huellas internamente también realiza la comparación de los datos guardados en su memoria *flash* o en otras palabras de las imágenes captadas de las huellas encontradas al encenderse la luz del led dentro del mismo y las envía mediante

un protocolo serial hacia el microcontrolador en este caso que se encuentra en el Arduino Uno.

El diodo led emite una luz a causa de las órdenes que recibe de la placa Arduino y esto resulta al obtener primero la información del módulo RFID-RC522 y del lector de huella digital.

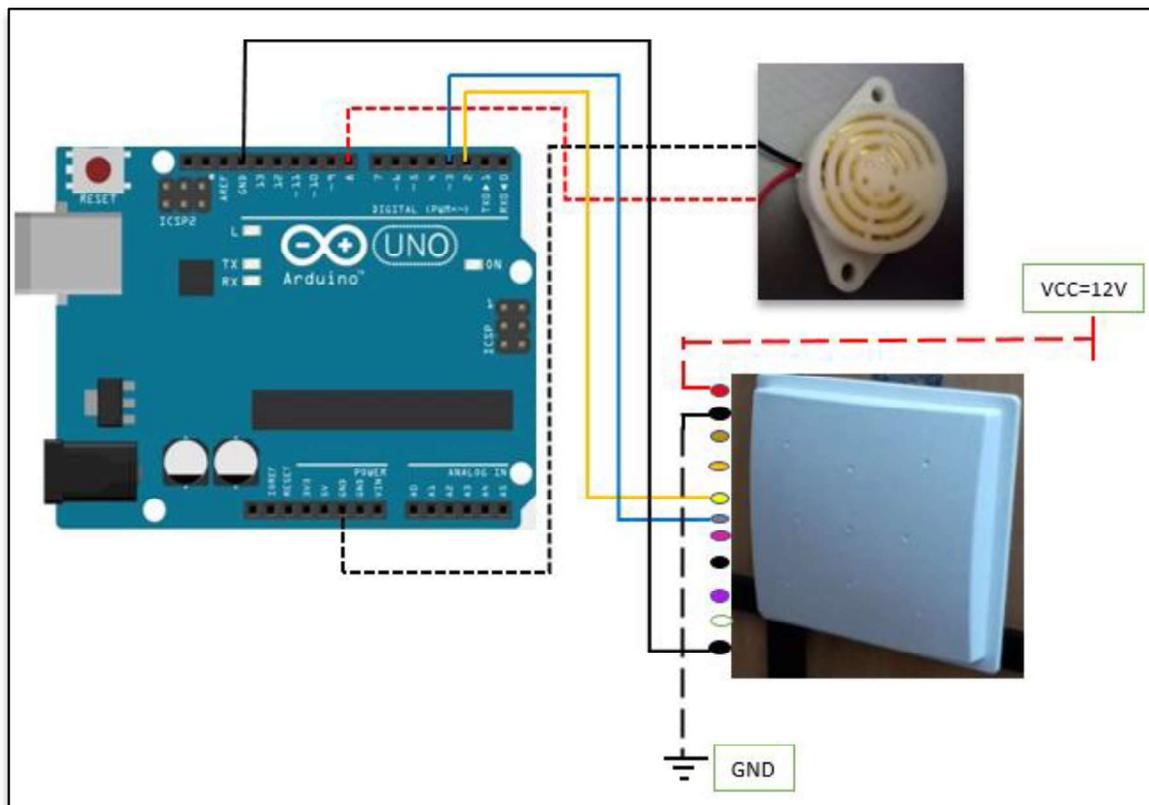


Figura 3.22 Diagrama final de conexión como cliente 2.

En el diagrama de conexión como cliente 2 de la Figura 3.22 se describe el funcionamiento de cómo la información captada por la antena RFID UHF proviene de los *tags* mismos que trabajan con ondas de radio a una frecuencia de 902 a 928Mhz y a una distancia menor de 1,50 metros lo cual es mucho mayor que el módulo RFID que solo permite lecturas a máximo 6 cm de distancia. Al captar esta información la antena UHF la procesa y la envía a la placa Arduino mediante un protocolo de comunicación llamado *Wiegand*, donde su nombre se debe a una marca de la sociedad “*Sensor Engineering Company*”.

El protocolo de comunicación *Wiegand* trata de una comunicación asíncrona que involucra las líneas de *WG DATA 0*, *WG DATA 1* y *GND*, donde si la antena se encuentra sin transmitir la línea *GND* estaría en un estado bajo y las líneas de datos en estado alto,

la transmisión se realiza de bit en bit por lo que para enviar un bit 1 o un bit 0 tendría que ocurrir lo siguiente:

Para transmitir un bit 1 se enviaría por la línea *WG DATA 1* un pulso a bajo que por lo general tiene una duración de 50 microsegundos, haciendo que la línea *WG DATA 0* permanezca en alto.

Para transmitir un bit 0 se enviaría por la línea *WG DATA 0* un pulso a bajo que por lo general tiene una duración de 50 microsegundos, haciendo que la línea *WG DATA 1* permanezca en alto.

La separación que existe entre cada pulso para ambos casos es de alrededor de unos 2 milisegundos. [20]

Al obtener toda la información del módulo RFID-RC522, del lector de huella digital y de la antena UHF viene a funcionar el Zumbador electrónico, debido a las órdenes que recibe desde el código ingresado en la placa Arduino dependiendo de las reglas que tiene.

Al estar conectada la placa Arduino directamente con el *ethernet shield* en ambos casos actúan como uno solo, pero la placa *ethernet shield* es la encargada de enviar la información hacia la red gracias al protocolo HTTP y a la comunicación de datos TCP donde se establece primero la comunicación y después se envían los datos mediante la dirección IP ocupada y su número de puerto. El método para establecer la conexión utilizado es el de cliente donde se envía la solicitud de conexión a un servidor como se detalla en el diagrama de la figura 3.23.

Las conexiones en el cliente 1 para alimentar la placa Arduino se la realizó mediante un cable USB A a B desde un computador y en el caso del cliente 2 se alimentó a la placa Arduino mediante un transformador de 12V y 1A, y para ambos casos el *ethernet shield* está conectado mediante un cable UTP de categoría 5e que terminan con conectores RJ45 usando la normativa EIA/TIA 568B.

Finalmente, al tener todas las conexiones y funcionamiento del *hardware* se procede a enviar los datos al servidor mediante el uso del protocolo HTTP donde lo que más influye para poder realizar el ingreso de la información hacia la base de datos es el comando **get** que permitió interconectar la información del Arduino mediante el *ethernet shield* hacia el servidor donde se procede a guiar la información hacia la interfaz de usuario como se explicará posteriormente.

La comunicación de los dos clientes después de establecer la conexión TCP se detalla en un diagrama completo en el Anexo II.

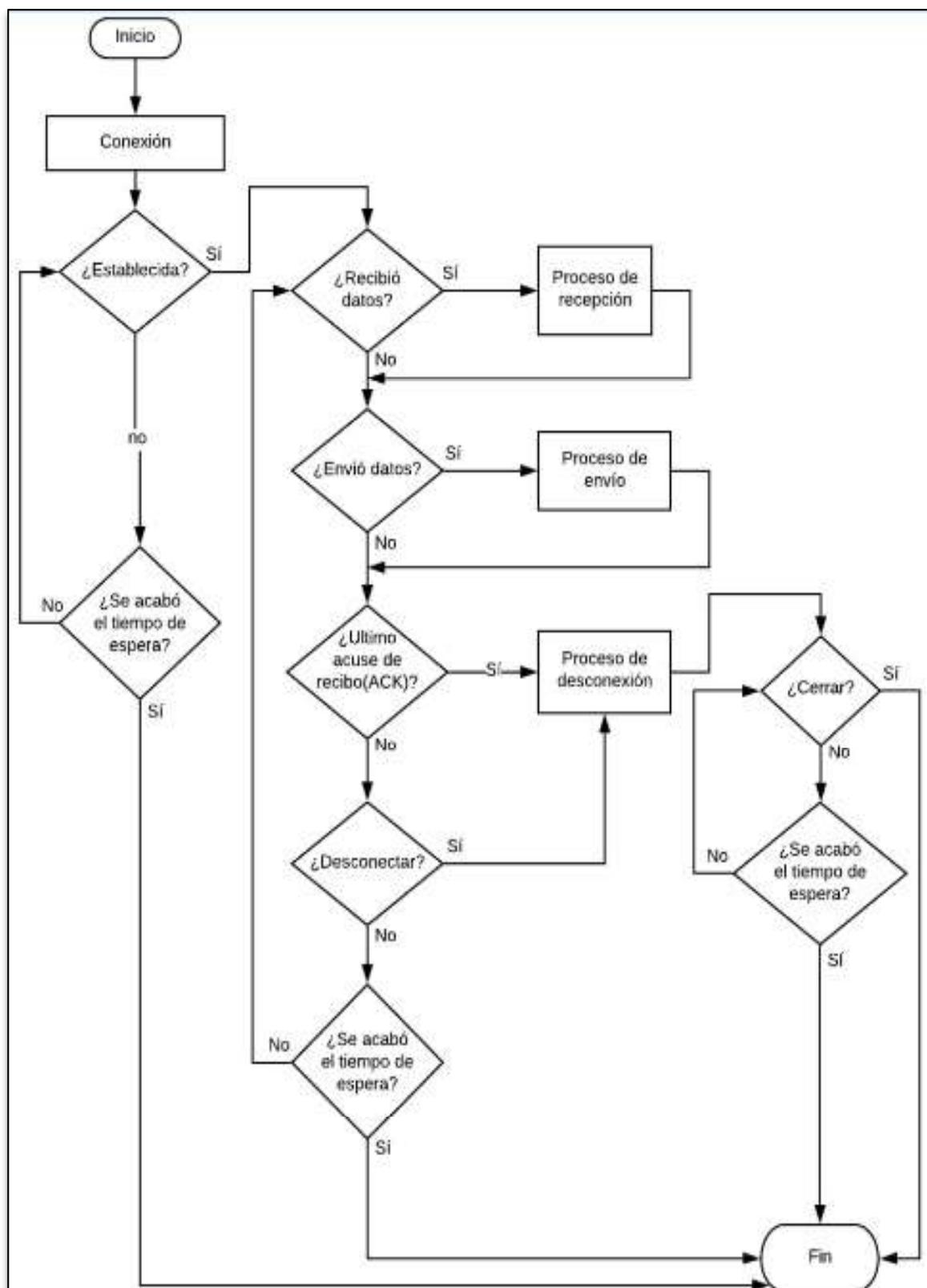


Figura 3.23 Diagrama para establecer la conexión (modo cliente) [21].

3.6. Programación del dispositivo

Las librerías que se muestran en la figura 3.24 del cliente 1 y la figura 3.25 del cliente 2, describen el funcionamiento de algunos de los elementos utilizados para el proyecto como se detalla a continuación y el código completo del cliente 1 en el Anexo III al igual que para el cliente 2 en el Anexo IV.

```
rfid_bio_eth
#include <SPI.h>
#include <Ethernet.h>
#include <MFRC522.h>
#include <Adafruit_Fingerprint.h>
```

Figura 3.24 Librerías para el cliente 1.

- **Librería SPI [22]**

La librería SPI se encuentra incluida en el IDE de Arduino, utiliza la comunicación de datos en serie para el *ethernet shield*, el lector de huella digital y para el módulo RFID-RC522 donde algunos de los comandos utilizados son:

#include<SPI.h>: Esta línea permite importar la librería SPI.

SPI.begin(): Esta línea se utiliza en la sección *setup* donde se inicializa el bus SPI.

Para el uso de esta librería también se tuvo que definir el pin SS con la línea de comando **#define SS_PIN 8**, donde se utiliza el pin número 8 de la placa Arduino.

Serial.begin(9600): Esta línea permite configurar la velocidad de transmisión en el puerto serie del módulo RFID-RC522 a 9600 baudios que es lo que se necesita para que opere correctamente.

- **Librería MFRC522. [23]**

Esta librería trabaja juntamente con la librería SPI ya que también utiliza la comunicación de datos en serie por lo que se define el pin SS pero también el pin de RESET con la línea de comando **#define RST_PIN 9** donde se utilizó el pin número 9 de la placa Arduino.

#include<MFRC522.h>: Esta línea permite importar la librería MFRC522 para el módulo RFID-RC522.

MFRC522 rfid(SS_PIN, RST_PIN): Esta línea crea la instancia de la clase MFRC522.

MFRC522::MIFARE_Key key: Esta línea se encarga de formar una estructura de tipo llave.

rfid.PCD_Init(): Esta línea permite que se inicialice el módulo RFID.

if (! rfid.PICC_IsNewCardPresent()): Esta línea busca si hay alguna tarjeta.

if (! rfid.PICC_ReadCardSerial()): Esta línea lee el uid de la tarjeta.

rfid.PICC_HaltA(): Esta línea permite detener el picc.

rfid.PCD_StopCrypto1(): Esta línea permite parar el cifrado en el módulo RFID.

- **Librería Adafruit_Fingerprint. [24]**

Esta librería al igual que las anteriores utiliza comunicación de datos en serie, permite al lector de huella digital realizar sus funciones y algunos de los comandos empleados para este proyecto se describen a continuación:

SoftwareSerial mySerial(2, 3): Esta línea permite utilizar el puerto 2 y 3 de la placa de Arduino, estos pines son necesarios debido a que en la placa Arduino se debe usar el *software* serial.

#include<Adafruit_Fingerprint.h>: Esta línea permite importar la librería *Adafruit_Fingerprint*.

finger.begin(57600): Esta línea permite ingresar la velocidad de datos para el puerto serie que en este caso es de 57600 baudios, valor que normalmente utiliza el lector de huella digital.

id_finding = getFingerprintIDez(): Esta línea permite identificar si la huella está registrada en el lector.

finger.getTemplateCount(): Esta línea permite al sistema realizar un recuento de las huellas obtenidas.

uint8_t p = finger.getImage(): Esta línea permite obtener la imagen de la huella digital.

```
uhf_eth
#include <Wiegand.h>
#include <SPI.h>
#include <Ethernet.h>
```

Figura 3.25 Librerías para el cliente 2.

- **Librería *Wiegand*. [25]**

Esta librería es utilizada debido a la antena RFID UHF ya que se requiere transmitir señales a larga distancia, además utiliza pines de interrupción de la placa Arduino para poder leer los pulsos de la interfaz *wiegand* y con ello devuelve los códigos. Esta librería también se encarga de decodificar el protocolo *wiegand*.

#include<Wiegand.h>: Esta línea permite importar la librería *Wiegand*.

WIEGAND wg: Esta línea realiza la declaración de la clase de *wiegand*.

Serial.begin(9600): Esta línea permite inicializar el puerto serial con una velocidad de 9600 baudios para que trabaje correctamente la antena RFID UHF.

wg.begin(): Esta línea permite inicializar la clase *wiegand*.

if(wg.available()): Esta línea pregunta si se ha leído alguna tarjeta para obtener su código.

mensaje = wg.getCode(): Esta línea obtiene el código y lo guarda en la variable.

- **Librería *Ethernet* [26]**

Con esta librería se controla el módulo de *ethernet shield*, pero esta consta de 5 clases que tienen sus diferentes métodos como se detalla a continuación:

Ethernet class: Esta clase se encarga de inicializar la librería y de configurar la red, para ello se utilizó **begin()** que inicializa la librería como constructor.

IPAddress class: Esta clase es utilizada para trabajar fácilmente con IPs locales y remotas mediante el comando **IPAddress()** la cual define una dirección IP.

Server class: Esta clase crea un servidor para enviar y recibir datos de los clientes, algunos de los comandos utilizados son:

Server(): Este comando es un constructor de la clase *server* el cual no se puede utilizar directamente.

Begin(): Mediante este comando se da la orden al servidor que empiece a escuchar.

Available(): Realiza la devolución del cliente y tiene datos listos para leer.

Print(): Se encarga de escribir datos a los clientes que se encuentren en el servidor.

println(): Se encarga de escribir datos a los clientes que se encuentren en el servidor pero seguido de una nueva línea.

Cliente class: Esta clase crea un cliente para enviar y recibir datos del servidor, algunos de los comandos utilizados son:

client: Este comando es un constructor de la clase cliente el cual no se puede utilizar directamente.

ethernetClient(): Se utiliza para crear un cliente que pueda conectarse a una IP y puerto determinado.

If(EthernetClient): Este comando pregunta si el cliente se encuentra preparado.

Available(): Realiza la devolución del número de *bytes* listos para leer.

Print(): Se encarga de escribir datos al servidor que se encuentre conectado.

Println(): Se encarga de escribir datos al servidor que se encuentre conectado pero seguido de una nueva línea.

Stop(): Se encarga de desconectar al cliente del servidor.

EthernetUDP class: Esta clase se encarga de habilitar el envío y recepción de mensajes UDP, consta de algunos comandos, pero en este caso no han sido utilizados.

3.7. Creación del sistema web

Para la creación de un nuevo proyecto en el *Framework* de Laravel se utilizó el comando **composer create-project laravel/laravel Nombre_del_proyecto "Versión (5.6.)"** Por medio de *composer*, la cual se visualizará en la siguiente imagen de la Figura 3.26.

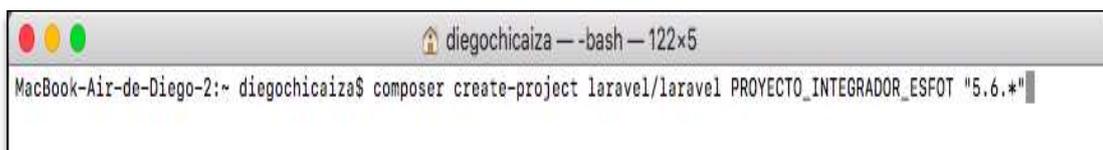


Figura 3.26 Comando para crear un nuevo proyecto en Laravel.

Para levantar el proyecto se utilizó el comando *php artisan serve* que se debe ejecutar en la carpeta de archivos que se creó el proyecto, el servidor de laravel se levantará en una IP 1xx.xxx.xxx.xxx y se iniciará en un puerto 8xxx (Véase Figura 3.27).

Para corroborar su correcto levantamiento se tendrá que abrir un navegador y colocar la IP con el puerto asignado por Laravel como se muestra en la Figura 3.28.



```
PROYECTO_ESFOT — php · php artisan serve — 122x5
MacBook-Air-de-Diego-2:~ diegochicaiza$ cd /Users/diegochicaiza/Dropbox/Tesis/Versiones\ del\ Software/Version\ 2/PROYECTO_ESFOT
MacBook-Air-de-Diego-2:PROYECTO_ESFOT diegochicaiza$ php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

Figura 3.27 Comando para la ejecución del proyecto de Laravel.



Figura 3.28 Ejecución del proyecto en el navegador.

3.8. Estructura del proyecto de la aplicación web

Al crear el proyecto, Laravel cuenta con una estructura organizada en sus archivos como se muestra en la Figura 3.29.

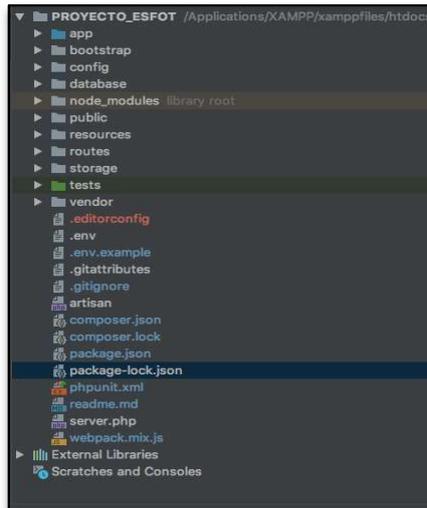


Figura 3.29 Estructura del proyecto.

3.9. Creación y configuración de la base de datos

Para la creación de la base de datos se debe abrir un navegador e ingresar por la URL la siguiente dirección ***http://localhost/phpmyadmin/*** (Figura 3.30.) de esa manera se podrá crear una nueva base de datos como se observa en la figura 3.31.



Figura 3.30 Creación de la base de datos.



Figura 3.31 Base de datos creada.

También se debe realizar la configuración para la conexión de base de datos con la aplicación *web* en donde se debe modificar el archivo *.env* que se encuentra en el proyecto

que se creó anteriormente, en la figura 3.32 donde se deberá modificar el nombre de la BDD, el *host*, el puerto, el nombre de usuario y el *password* correspondiente.

```

9      DB_CONNECTION=mysql
10     DB_HOST=127.0.0.1
11     DB_PORT=3306
12     DB_DATABASE=Proyecto_ESFOT
13     DB_USERNAME=root
14     DB_PASSWORD=

```

Figura 3.32 Archivo.env modificado y configurado.

3.10. Diagramas de la base de datos

Se diseñó el modelo físico de la base de datos en donde se encuentran las entidades con sus respectivos atributos, las entidades se transforman en tablas y los atributos se convierten en columnas, también se especifican las claves primarias y se mantiene su relación entre las tablas como se muestra en la figura 3.33.

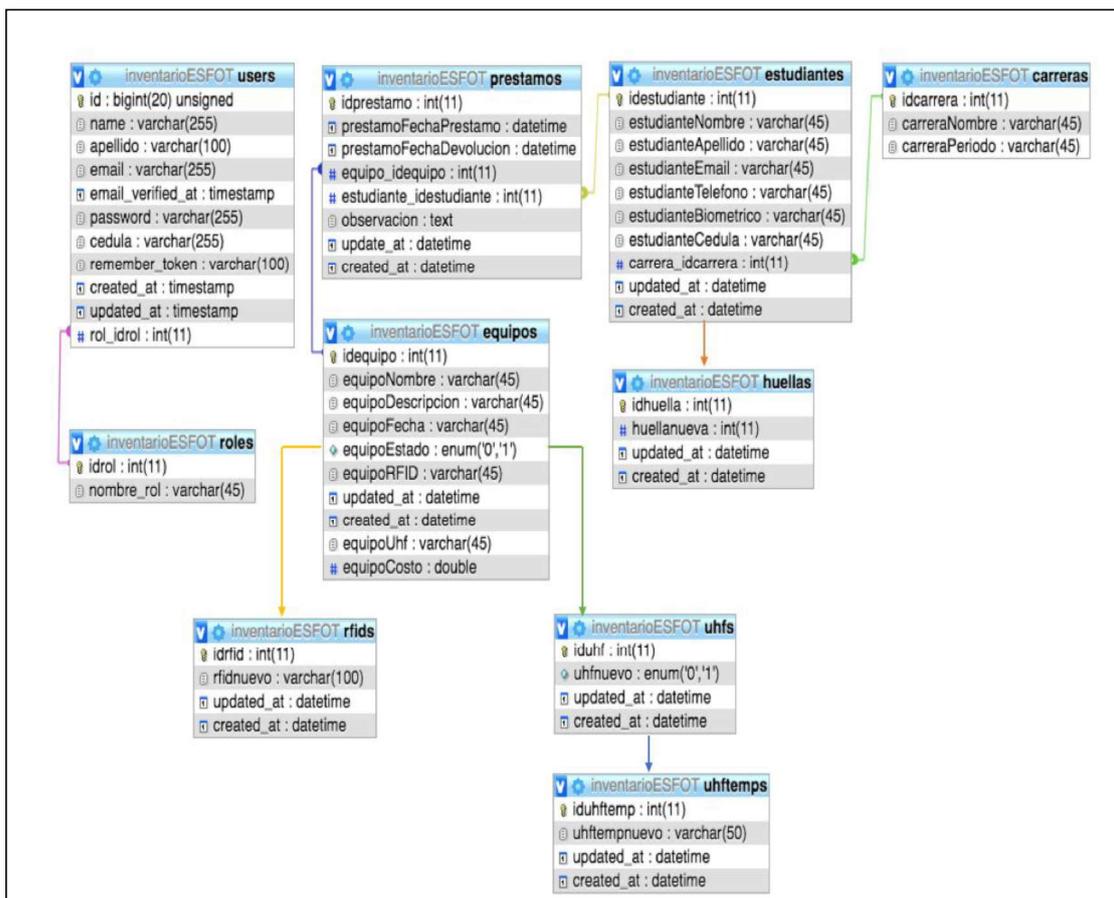


Figura 3.33 Modelo físico de la base de datos.

3.11. Desarrollo de la aplicación web

Creación del Login

Para el ingreso al sistema web se creó el siguiente Login de la figura 3.34 para el control de los roles de ingreso como son: administrador y las personas encargadas del laboratorio.

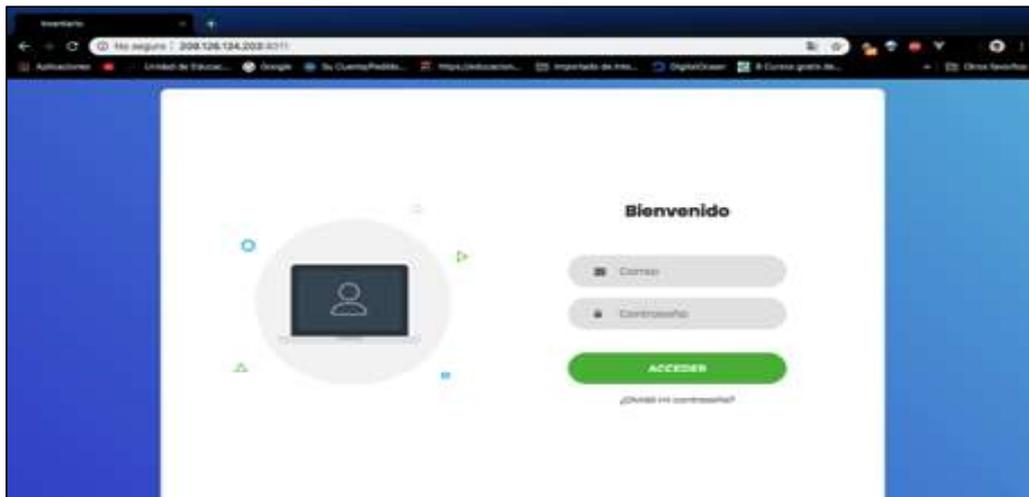


Figura 3.34 Login del sistema web.

Creación y desarrollo del módulo Estudiante

En esta sesión se puede registrar o eliminar estudiantes, a la vez existe un buscador que permite filtrar los datos por nombre como se observa en la figura 3.35.

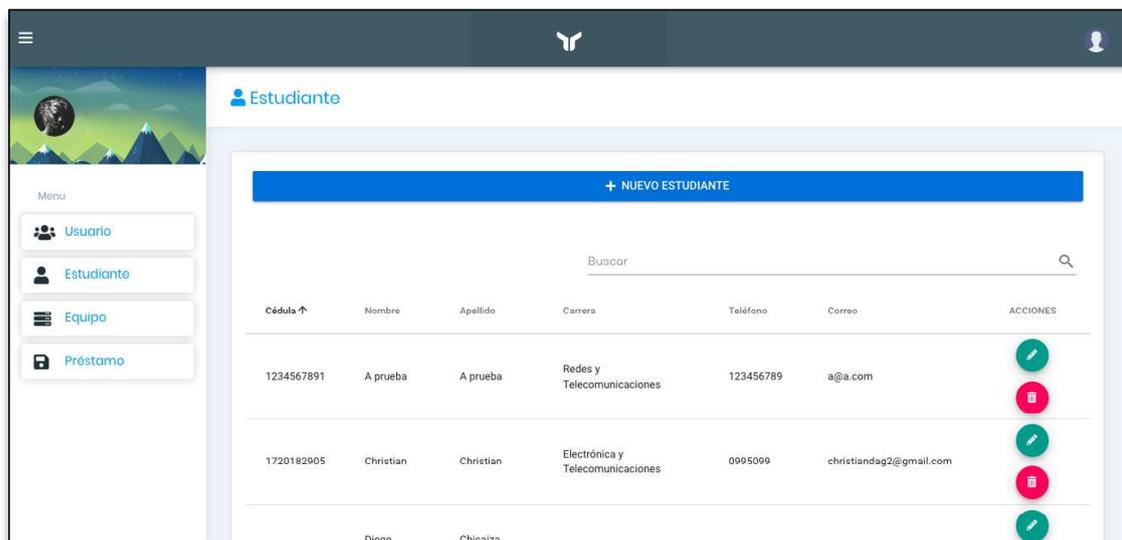
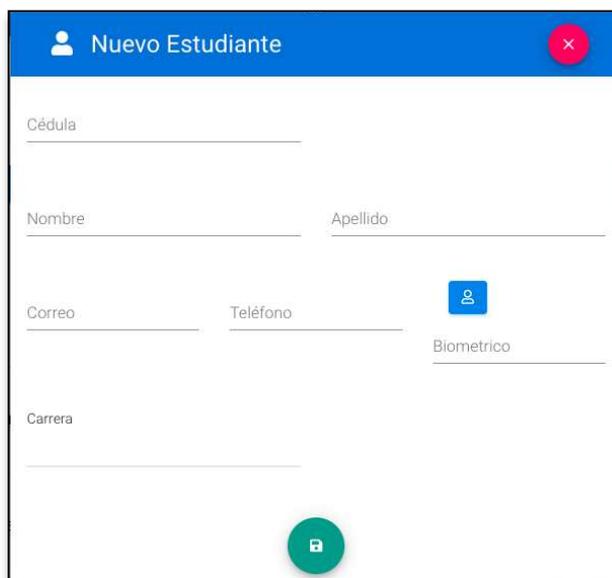


Figura 3.35 Módulo de Estudiante.

Al registrar un nuevo estudiante la aplicación *web* solicitará: número de identificación, nombres, apellidos, correo personal o institucional, teléfono, especialidad y la huella dactilar, lo cual servirá para las prestaciones de los equipos que se encuentren en el laboratorio, véase figura 3.36.



Modal de registro de nuevo estudiante. El formulario contiene los siguientes campos:

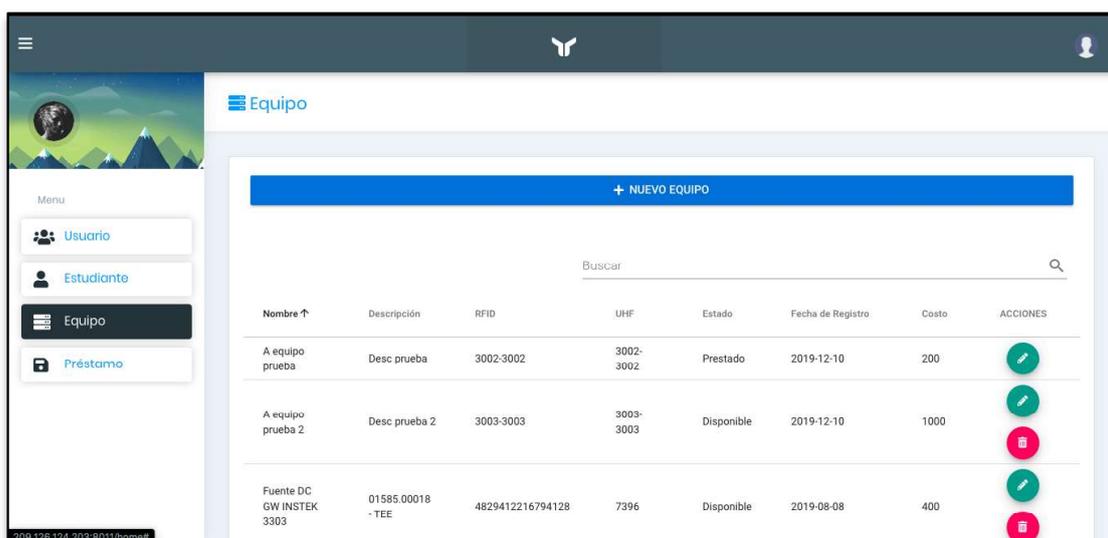
- Cédula
- Nombre
- Apellido
- Correo
- Teléfono
- Biometrico (con ícono de huella)
- Carrera

El modal tiene un botón de cerrar (X) en la esquina superior derecha y un botón de guardar (con ícono de disco) en la parte inferior central.

Figura 3.36 Modal de registro de nuevo estudiante.

Creación y desarrollo del módulo Equipo

En esta sesión, se puede registrar o eliminar equipos; a la vez, existe un buscador que permite filtrar los datos por nombre de equipo como se ve en la figura 3.37.



Módulo Equipo. Interfaz de usuario que muestra un menú lateral y una tabla de equipos.

Menú:

- Usuario
- Estudiante
- Equipo**
- Préstamo

Tabla de Equipos:

Nombre ↑	Descripción	RFID	UHF	Estado	Fecha de Registro	Costo	ACCIONES
A equipo prueba	Desc prueba	3002-3002	3002-3002	Prestado	2019-12-10	200	
A equipo prueba 2	Desc prueba 2	3003-3003	3003-3003	Disponible	2019-12-10	1000	
Fuente DC GW INSTEK 3303	01585.00018 - TEE	4829412216794128	7396	Disponible	2019-08-08	400	

Figura 3.37 Módulo Equipo.

Al registrar un nuevo equipo la aplicación *web* solicitará: nombre del equipo, descripción, costo del equipo y datos de identificación UHF y RFID como se observa en la figura 3.38.

Figura 3.38 Modal de registro de nuevo equipo.

Creación y desarrollo del módulo usuario

A esta sesión solo podrá ingresar el o los administradores autorizados y es en donde podrá registrar a las personas encargadas de los laboratorios, véase figura 3.39.

Cédula ↑	Nombre	Correo	Rol	ACCIONES
0	Admin	laboratoriofееepn@gmail.com	Administrador	
1234567890	Admin	laboratoriotееepn@gmail.com	Administrador	
1724174345	Angela	angela.lagua6b@hotmail.com	Encargado del laboratorio	

Figura 3.39 Módulo de usuario.

En esta sesión se registró los siguientes datos: número de cédula, nombres, apellidos, correo y se le asignará un rol (administrador o encargado del laboratorio) como se observa en la figura 3.40.

Figura 3.40 Modal de registro de usuario.

Creación y desarrollo del módulo de préstamo

En este módulo se podrá realizar los préstamos y devoluciones de los equipos que se encuentran en el laboratorio de acuerdo con la figura 3.41.

Cédula ↑	Estudiante	Artículo	Fecha de Préstamo	Fecha de Devolución	Observación	Estado	ACCIONES
1720182905	Christian Christian	A equipo prueba	2019-12-12 13:48:33			✓	
1720182905	Christian Christian	A equipo prueba	2019-12-10 23:17:47	2019-12-12 13:40:06		✗	🗑️
1720182905	Christian Christian	A equipo prueba	2019-12-10 20:58:15	2019-12-10 23:17:13		✗	🗑️
1720182905	Christian Christian	A equipo prueba	2019-12-10 20:51:22	2019-12-10 20:53:17	orueba	✗	🗑️
1723451637	Diego Armando Chicaiza Chicaiza	Multimetro FLUKE 87	2019-11-26 12:06:57	2019-11-26 12:07:46	ultimaas pruebas	✗	🗑️

Figura 3.41 Módulo de préstamos y devoluciones.

Al realizar un nuevo préstamo de equipo, el prestatario deberá estar ya registrado de esa manera solo deberá colocar su huella dactilar y acercar la etiqueta RFID de corta distancia para proceder al préstamo, véase figura 3.42.

Nuevo Prestamo

Equipo

RFID

Nombre

Descripcion

Estudiante

Cedula

Nombre

Apellido

Correo

Telefono

Observacion

Salir Guardar

Figura 3.42 Modal de préstamo de equipo.

Para la devolución del equipo, la persona que realizó el préstamo deberá acercar el tag RFID de corta distancia al módulo RFID-RC522 y devolver el equipo a la persona que se encuentre en el laboratorio, ya que los datos se llenarán automáticamente, véase figura 3.43.

Devolucion

Equipo

RFID

Nombre

Descripcion

Estudiante

Cedula

Nombre

Apellido

Correo

Telefono

Observacion

Salir Guardar

Figura 3.43 Modal de devolución de equipos.

3.12. Instalación

Después de verificar el funcionamiento total de los componentes electrónicos, su interacción con la base de datos, interfaz gráfica y compatibilidad con la página *web*, se procedió a realizar las cajas en las que se encuentran los componentes electrónicos y posteriormente instalarlas en el laboratorio con el resto de los componentes y cableado que conforman el proyecto.

La caja de madera para el cliente 1 es de 11 cm de alto por 11 cm de ancho y con una profundidad de 14 cm dentro de esta se distribuyeron algunos componentes electrónicos como se puede observar en la figura 3.44. La placa Arduino se encuentra levemente separada de la madera gracias a unos pequeños cauchos, atornillada para que no se pueda mover facilitando y asegurando para la conexión de los cables de alimentación y de red. El módulo RFID-RC522 fue colocado al lado derecho del frente de la caja, se aseguró mediante dos tornillos igualmente con dos pequeños cauchos por debajo y se reforzó con silicona debido a que el módulo queda algo expuesto fuera de la caja para poder registrar las etiquetas de corta distancia en los equipos. El lector de huella digital se colocó a la izquierda del frente de la caja, se aseguró mediante cinta doble faz y un poco de silicona ya que no se pueden colocar tornillos en este lector. En la parte superior central del frente de la caja se encuentra colocado un diodo led color azul, el cual se aseguró por la parte trasera con silicona.



Figura 3.44 Caja del cliente 1 con su respectivo montaje.

La caja de madera para el cliente 2 es de 8.50 cm de alto por 11 cm de ancho y con una profundidad de 14 cm dentro de esta se distribuyeron algunos componentes electrónicos como se puede observar en la figura 3.45. La placa Arduino se encuentra atornillada levemente separada de la madera gracias a unos pequeños cauchos y sobre esta se encuentra el módulo *ethernet shield*, aseguradas para que no se puedan mover facilitando la conexión de los cables de alimentación y de red. La caja cuenta con una salida para el cableado de la antena RFID-UHF y para el zumbador electrónico.

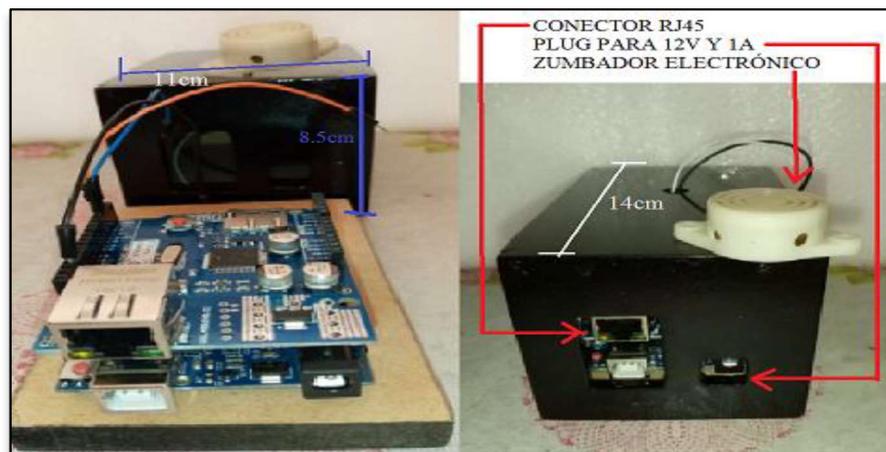


Figura 3.45 Caja del cliente 2 con su respectivo montaje.

La antena se encuentra instalada frente a la puerta principal del laboratorio, asegurada mediante 4 tornillos y sobre ella a unos 100 cm el zumbador electrónico el cual, está sujeto con 2 tornillos y su cableado sujeto con silicona como se observa en la figura 3.46.

El cableado de red, la alimentación para la caja del cliente 2 y la alimentación para la antena RFID-UHF se encuentra colocado dentro de la canaleta que va desde los tomacorrientes a lado del escritorio hasta la caja del cliente 2 como se encuentra en las figuras 3.47 y 3.48.



Figura 3.46 Instalación de antena UHF y zumbador.

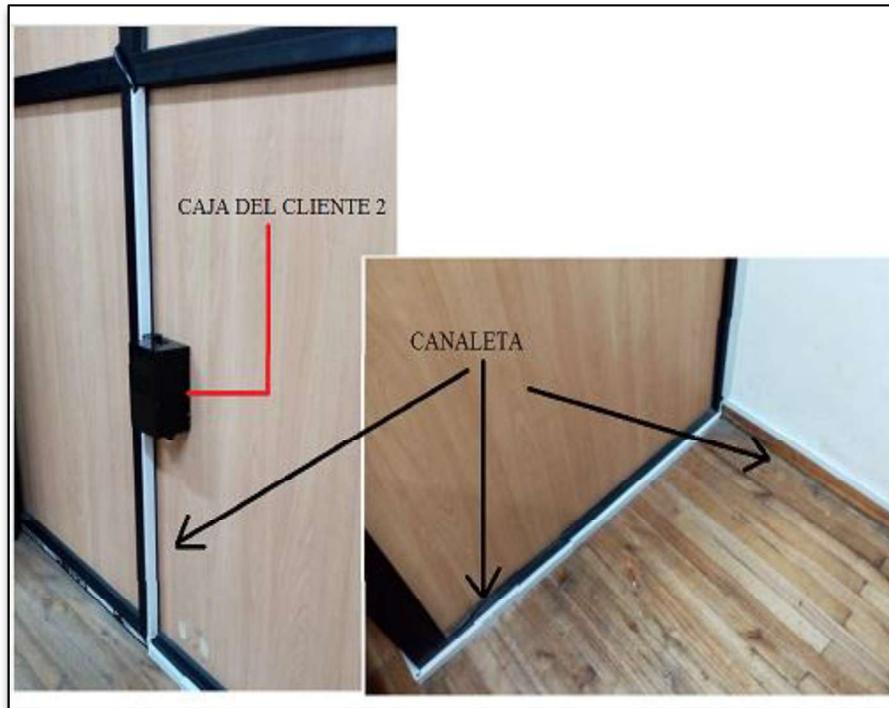


Figura 3.47 Instalación del cableado del cliente 2.



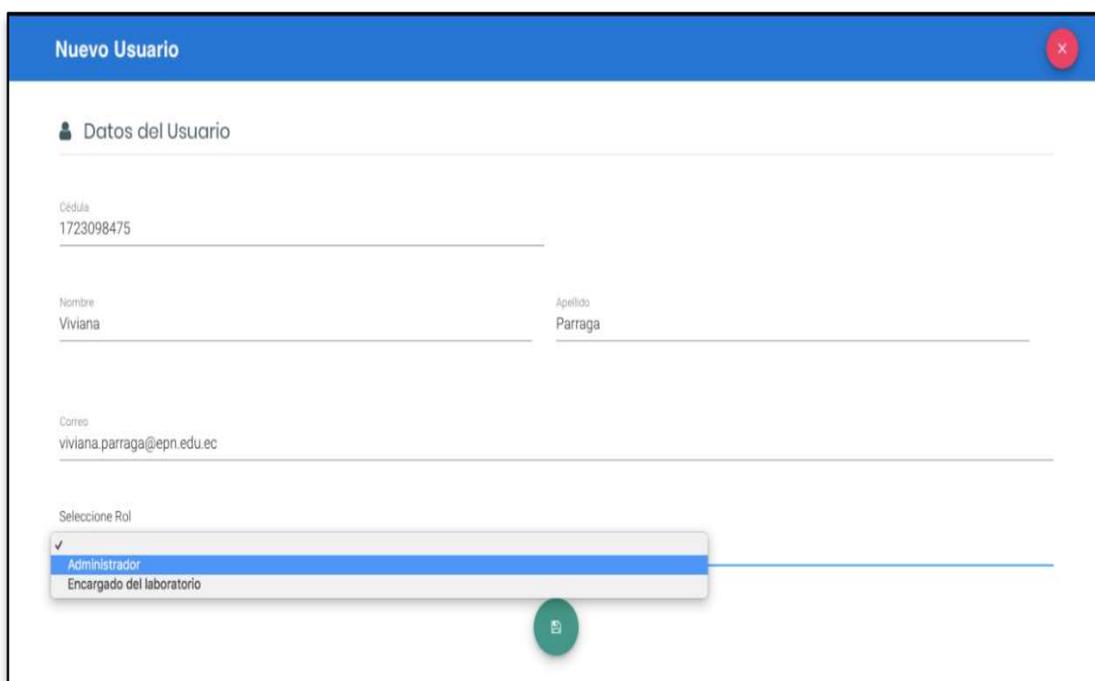
Figura 3.48 Instalación del cableado del cliente 2.

3.13. Pruebas

Para la utilización del sistema se deberá tener ingresado en la base de datos los siguientes *ítems*: administrador o encargado de laboratorio, estudiantes y equipos, los cuales se detallarán a continuación.

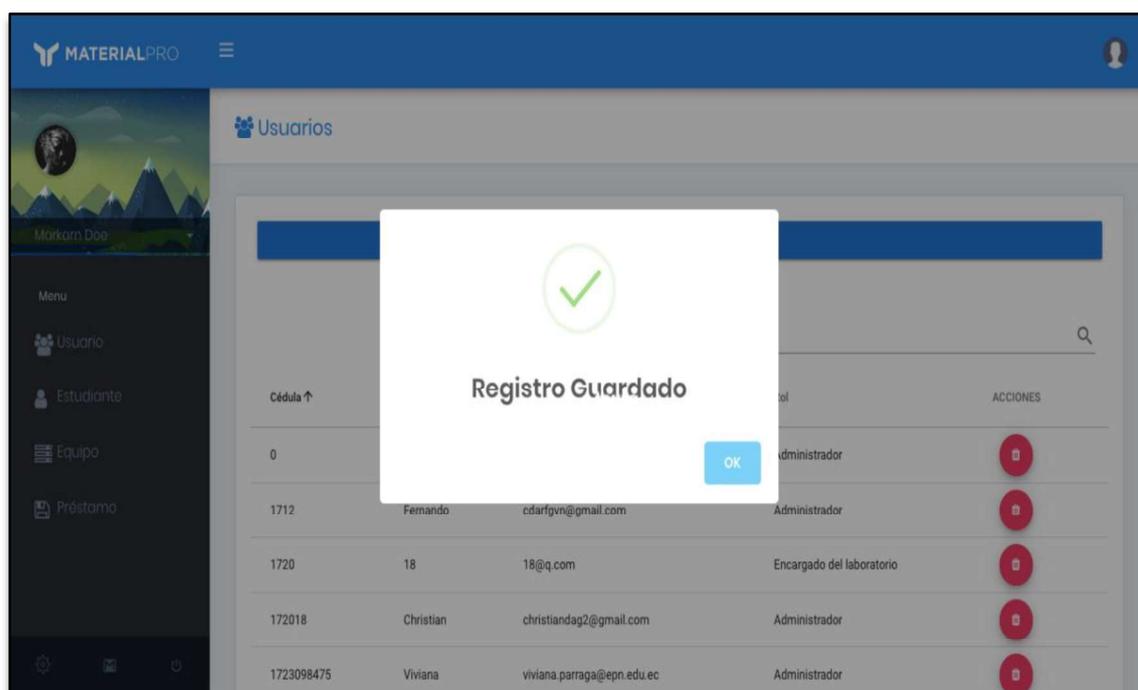
Registro de administrador

En este módulo se deberá registrar los datos de la persona que va a ser un nuevo administrador como indica la figura 3.49, después de guardar los datos como se observa en la figura 3.50 se podrá ingresar al sistema con sus módulos completos, véase figura 3.51.



The screenshot shows a form titled "Nuevo Usuario" with a close button in the top right corner. The form is divided into sections for user data. The "Datos del Usuario" section includes fields for "Cédula" (ID number) with the value "1723098475", "Nombre" (Name) with the value "Viviana", and "Apellido" (Last name) with the value "Parraga". Below these is a "Correo" (Email) field with the value "viviana.parraga@epn.edu.ec". A "Seleccione Rol" (Select Role) dropdown menu is open, showing two options: "Administrador" (selected) and "Encargado del laboratorio". A green circular button with a document icon is located at the bottom center of the form.

Figura 3.49 Registro de usuario (administrador).



The screenshot shows the "Usuarios" page in the MATERIALPRO system. A white confirmation dialog box is centered on the screen, displaying a green checkmark icon and the text "Registro Guardado" (Registration Saved), with an "OK" button. In the background, a table lists the registered users. The table has columns for "Cédula" (ID), "Nombre" (Name), "Apellido" (Last name), "Correo" (Email), "Rol" (Role), and "ACCIONES" (Actions). The user "Viviana Parraga" is listed at the bottom of the table with the role "Administrador".

Cédula ↑	Nombre	Apellido	Correo	Rol	ACCIONES
0				Administrador	
1712	Fernando		cdarfyn@gmail.com	Administrador	
1720	18		18@q.com	Encargado del laboratorio	
172018	Christian		christiandag2@gmail.com	Administrador	
1723098475	Viviana		viviana.parraga@epn.edu.ec	Administrador	

Figura 3.50 Usuario ya registrado.

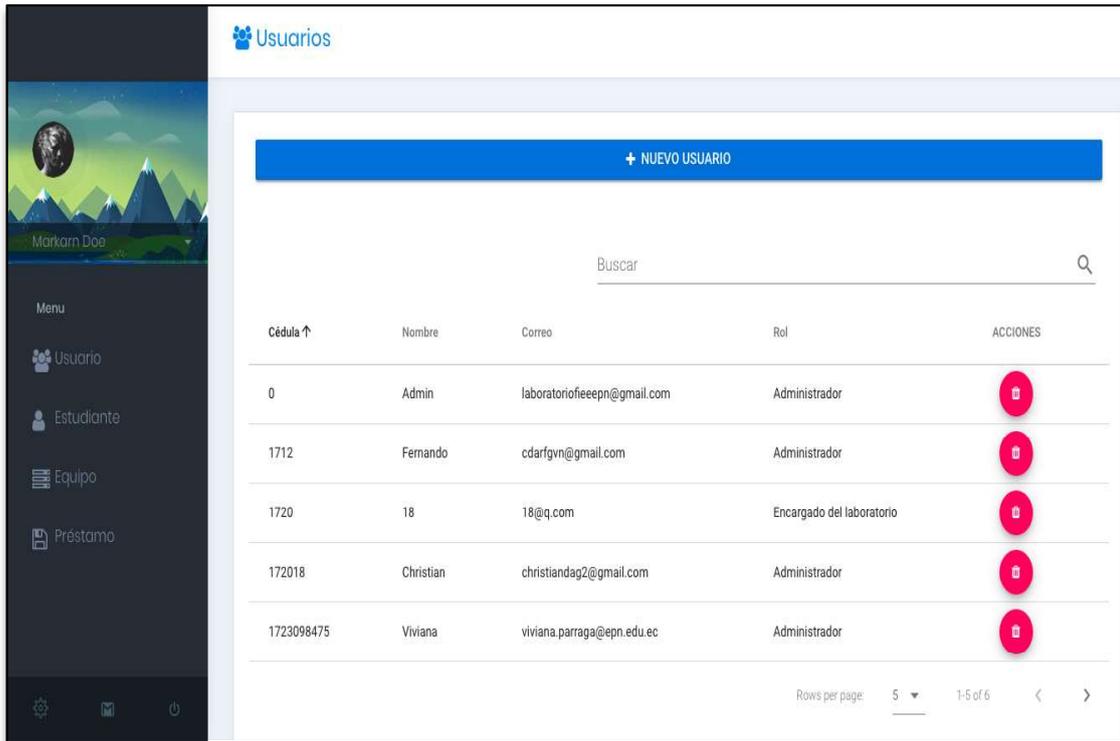


Figura 3.51 Visualización de usuario ya registrado en la base de datos.

Registro para personas encargadas del laboratorio

Al igual que el administrador el encargado del laboratorio deberá ser registrado en el sistema para realizar el préstamo como se observa en las figuras 3.52 y 3.53.

The screenshot shows a registration form titled 'Datos del Usuario'. The form contains the following fields and options:

- Cédula:** 1723087571
- Nombre:** Maritza
- Apellido:** Rosero
- Correo:** maritza.rosero@epn.edu.ec
- Selección de Rol:** A dropdown menu with two options: 'Administrador' (selected with a checkmark) and 'Encargado del laboratorio'.

A green circular button with a plus sign is located at the bottom center of the form.

Figura 3.52 Registro del usuario (encargado del laboratorio).

Cédula ↑	Nombre	Correo	Rol	ACCIONES
0	Admin	laboratoriofiecepn@gmail.com	Administrador	
1712	Fernando	cdarfyn@gmail.com	Administrador	
1720	18	18@iq.com	Encargado del laboratorio	
172018	Christian	christiandag2@gmail.com	Administrador	
1723087571	Maritza	maritza.rosero@epn.edu.ec	Encargado del laboratorio	

Figura 3.53 Usuario encargado de laboratorio registrado en la base de datos.

Registro de estudiantes

Para este módulo se deberá ingresar los datos del estudiante (figuras 3.54 y 3.55.) para que pueda realizar los futuros préstamos.

Nuevo Estudiante

Cedula
1724174345

Nombre: Angela Apellido: Laguna

Correo: angela.lagua6b@hotmail.com Telefono: 0984271629 Biometrico: 4

Carrera: Electrónica

Salir Guardar

Figura 3.54 Datos de un nuevo estudiante.

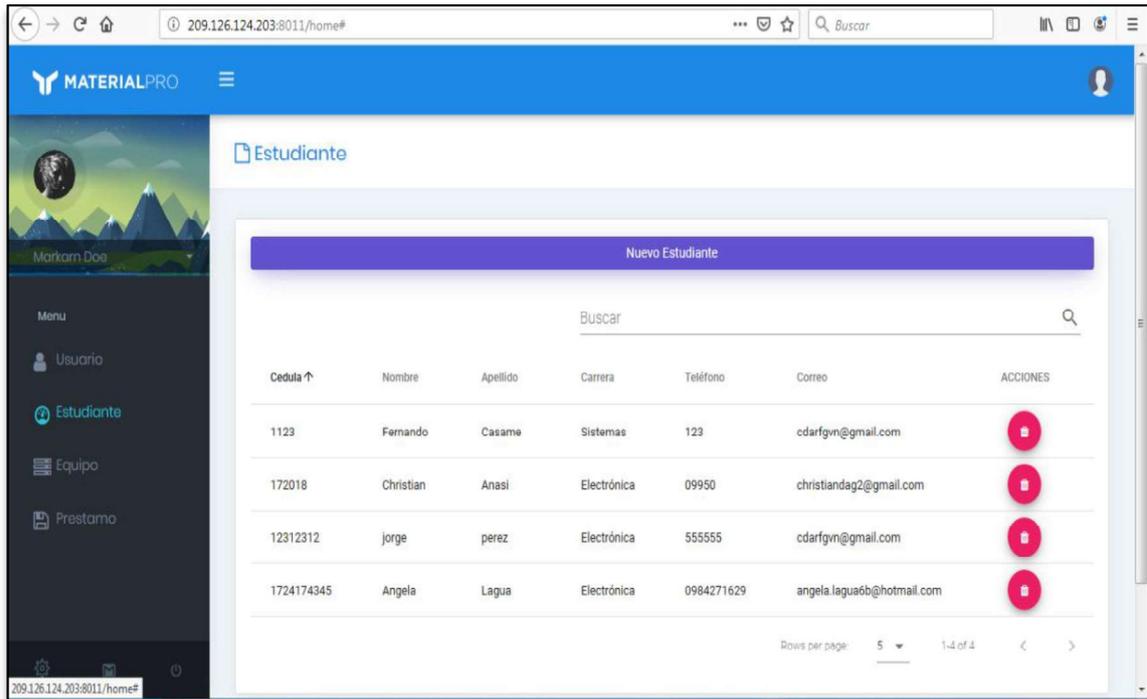


Figura 3.55 Estudiante ya registrado en la base de datos.

Registro de Equipos

En este módulo se deberá ingresar los datos principales de los equipos (véase figuras 3.56 y 3.57.) para los futuros préstamos.

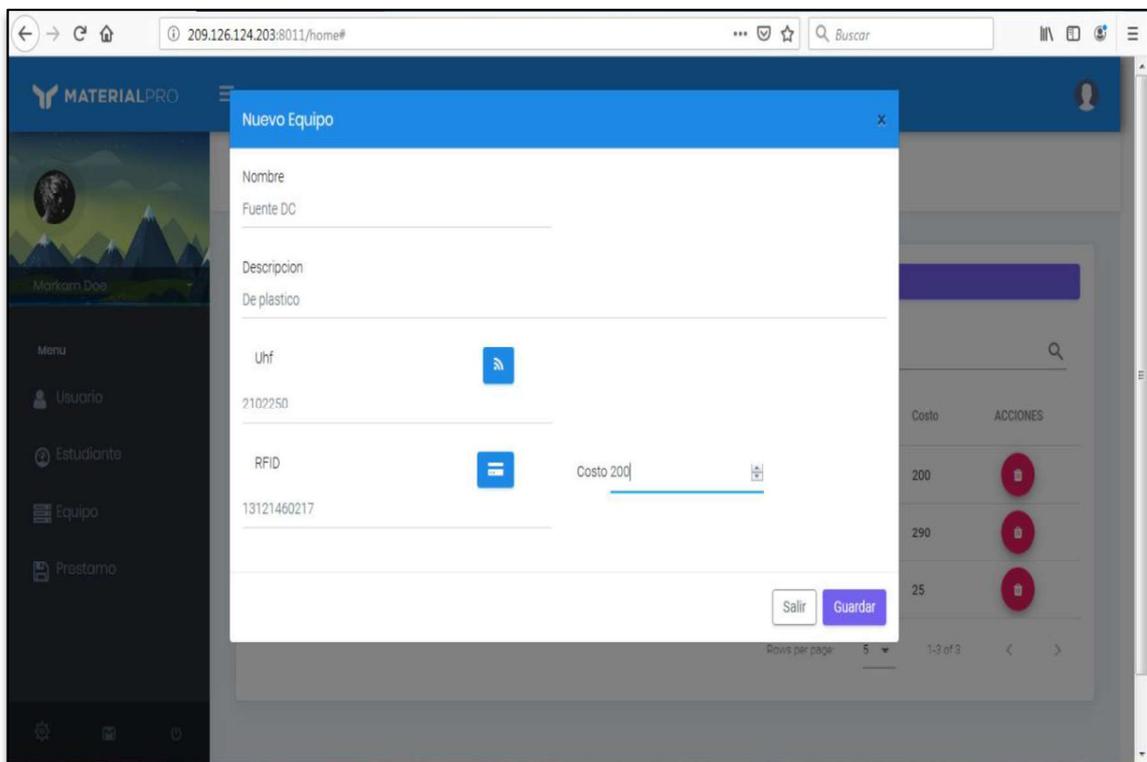


Figura 3.56 Ingreso de datos de un nuevo equipo.

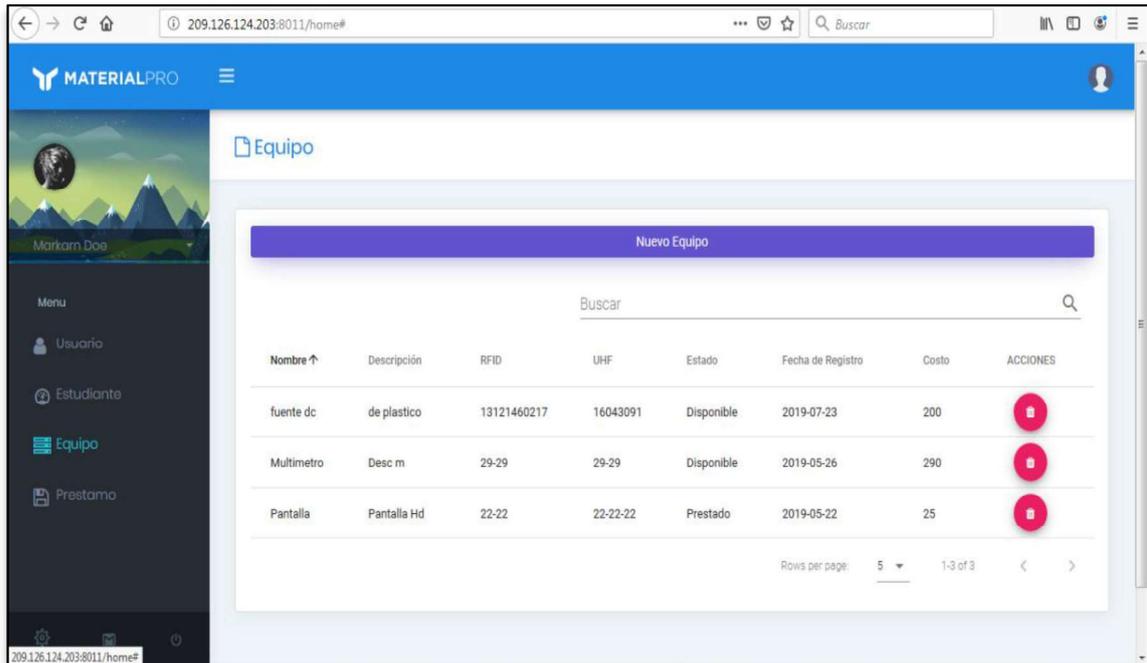


Figura 3.57 Equipo registrado en la base de datos.

Si se desea ingresar el equipo con la misma etiqueta el programa emitira una alerta como muestra la figura A.

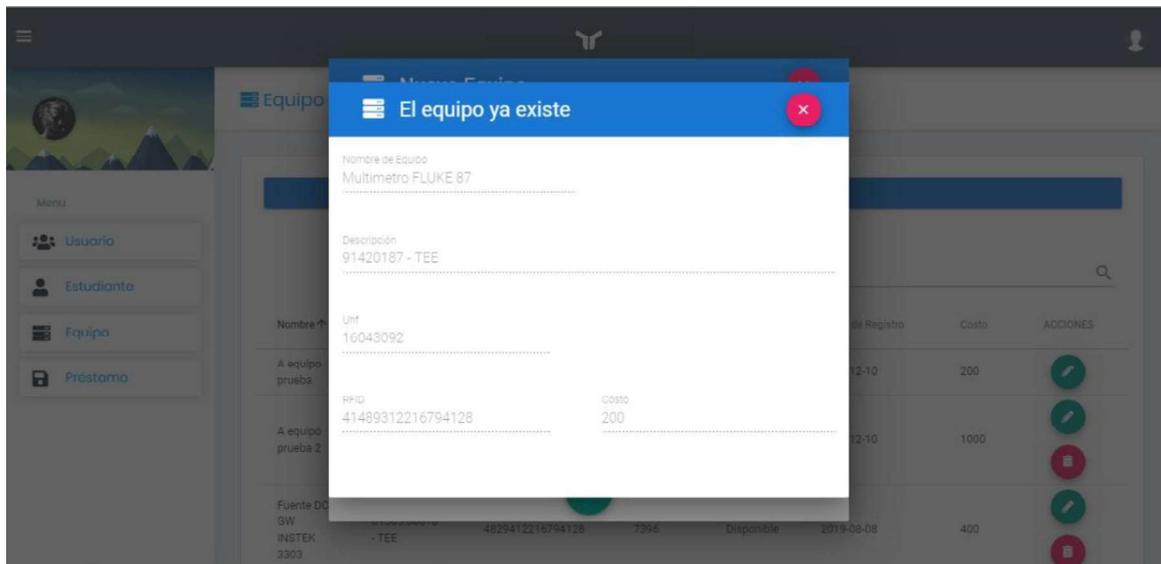


Figura A Equipo ya registrado en el sistema

Préstamo del equipo

Para los préstamos de equipos deben estar registrados los estudiantes y los equipos, el estudiante deberá colocar su huella dactilar y acercar el tag de corta distancia para que los datos se llenen, véase figura 3.58 y 3.59, inmediatamente el estudiante recibirá un

correo de notificación al momento de realizar el préstamo del equipo como se indica en la figura 3.60.

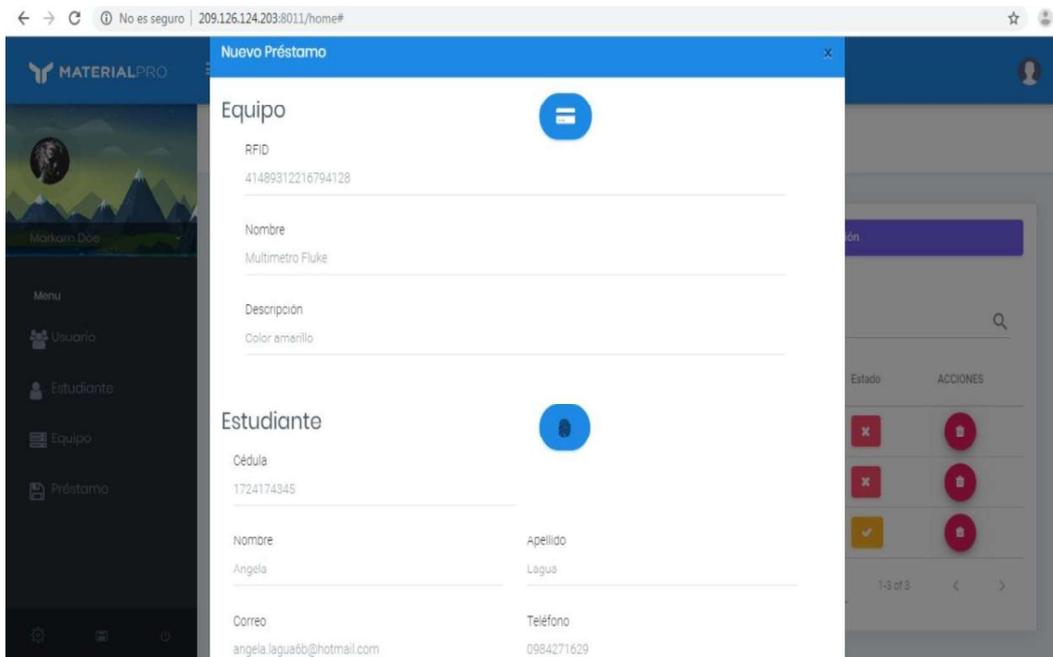


Figura 3.58 Préstamo del equipo.

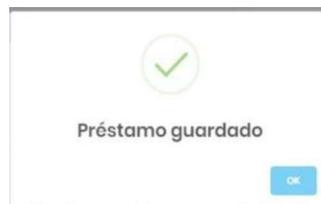


Figura 3.59 Mensaje de actividad guardada.



Figura 3.60 Correo de notificación de préstamo.

Devolución del equipo

Para la devolución del equipo el estudiante deberá acercarse a la caja del cliente 1, pasar el tag de corta distancia para registrar en el sistema la devolución (Figura 3.61.) y de igual forma recibirá un correo de notificación indicando el respaldo de la devolución como se observa en la figura 3.62.

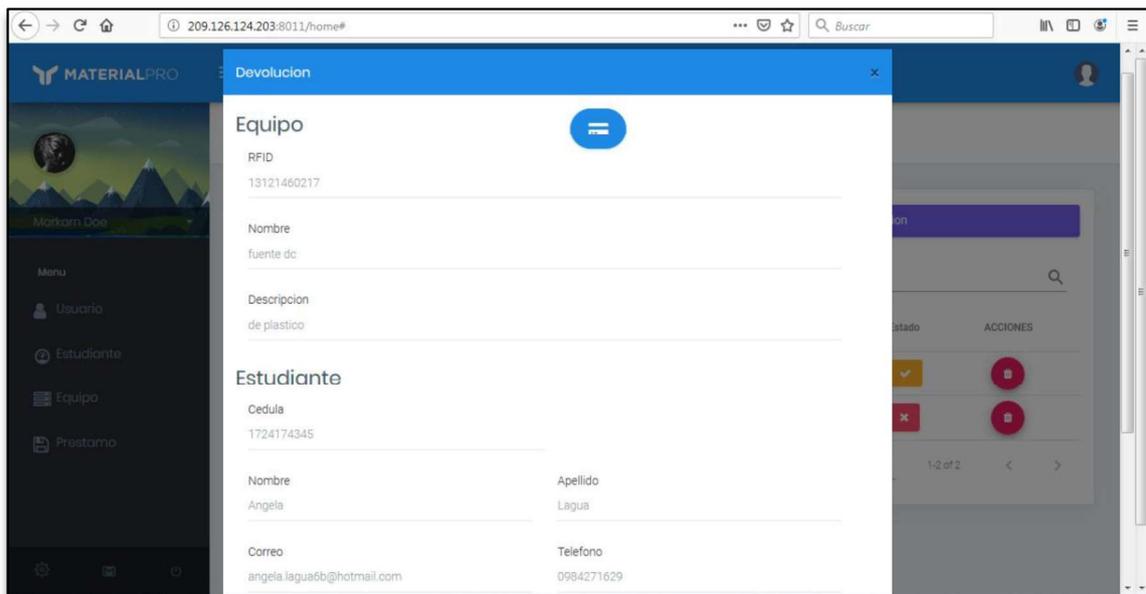


Figura 3.61 Devolución del equipo.



Figura 3.62 Correo de notificación de devolución del equipo.

Sistema de seguridad

En caso de no ser registrado el préstamo del equipo correctamente (Figura 3.63), el sistema cuenta con una alarma acústica la cual está compuesta de un zumbador que

suenan en intervalos de tiempo de 30 milisegundos alertando al personal del laboratorio de acuerdo con la figura 3.64.



Figura 3.63 Préstamo no autorizado.

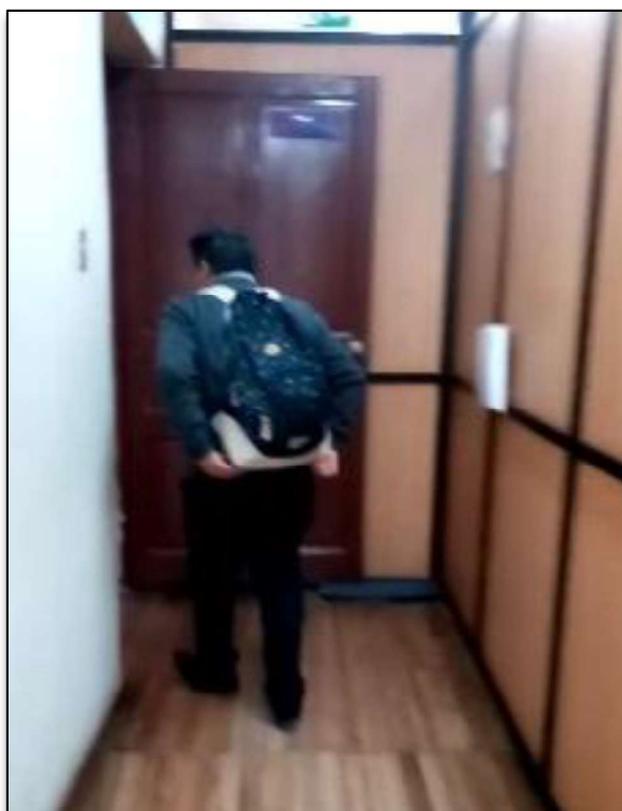


Figura 3.64 Activación del zumbador por equipo no prestado.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Se implementó un sistema de inventario para el laboratorio de Tecnología Eléctrica y Electrónica, aprovechando la tecnología RFID la cual permite la identificación de *tags* de forma inalámbrica, así como la utilización de otros componentes electrónicos con mejores características. Además, la interfaz gráfica posee funcionalidades fáciles de utilizar, debido a ello, se permitió desarrollar este proyecto para brindar a los estudiantes, docentes y personal encargado del laboratorio las facilidades al momento que requiera algún equipo y en el registro de estos ahorrar tiempo generando también seguridad de los activos.

Para el diseño físico y electrónico del proyecto se tuvo que analizar los requerimientos que se obtuvieron por la encargada de laboratorio. Además de, las características que proporciona la placa Arduino Uno, el módulo *Ethernet Shield*, el módulo RFID-RC522, el lector de huella dactilar y la antena RFID-UHF. Se analizaron los puertos y métodos que permitieron la interconexión entre ellos. El código se desarrolló en un lenguaje de alto nivel para su comunicación y con todo esto se diseñó el proyecto que cumplió con las características físicas y funcionales de tal manera que resultó un diseño adaptable y de fácil manejo.

Para el desarrollo de la aplicación *web* se utilizaron algunas herramientas de *software* libre como Laravel que es un *framework* para el desarrollo de PHP y a su vez existe una extensa documentación ya sea en grupos, foros e incluso la propia página de Laravel.

Para la implementación de la aplicación *web* se tuvieron que realizar varias pruebas con el servidor para poder tener una excelente conexión con el equipo Arduino situado en el cliente 1 y cliente 2. Al momento de hacer las peticiones de las antenas y el lector hacia el servidor se encontraron dificultades de conexiones. La institución tiene deshabilitados los puertos por los cuales se deseaba realizar las pruebas correspondientes, motivo por el cual se tuvo que gestionar permisos administrativos que fueron solucionados de manera concreta y así se pudo realizar las pruebas respectivas.

Se realizaron pruebas del sistema de inventario con los equipos que ya cuentan con las etiquetas y la aplicación *web*. Se verificó el registro de nuevos estudiantes, usuarios, administradores y equipos, además pruebas de préstamos, devoluciones, pruebas de alcance, interferencias y verificaciones de la alerta acústica que solo se accione en el caso de un préstamo no autorizado para de esa manera cumplir con las especificaciones

recomendadas inicialmente. Las pruebas arrojaron resultados satisfactorios confirmando que las conexiones de *hardware* y *software* son correctas.

El sistema de inventario realizado ayudará a la responsable del laboratorio a mejorar los préstamos de los equipos que reposan en el laboratorio, a su vez llevar un control adecuado de los equipos que son prestados y el control sobre los usuarios que realizan los préstamos de éstos.

En la aplicación *web* que se encuentra alojada en un computador interno en la ESFOT se tuvo que realizar los ajustes necesarios para obtener una excelente conexión con los equipos Arduino y de esa manera evitar errores de almacenamiento de información en la base de datos que tiene el servidor.

El proyecto permite tener la facilidad de saber si algún equipo requerido se encuentra prestado o si está guardado, sin la necesidad de ir a observar dentro del laboratorio, ya que, mediante la aplicación *web* se puede chequear la lista de equipos desde un computador que tenga acceso a internet.

El proyecto a su vez resultó un complemento con el sistema de cámaras que se encuentran instalados en los pasillos de la ESFOT, ya que, si existiera algún hurto de los equipos aparte de contar con una alarma acústica se podría observar quién es la persona que sale con el equipo al instante que ésta se activa, brindando al laboratorio mayor seguridad.

El hardware utilizado fue seleccionado gracias a las características que brinda como son sus bajos costos, fáciles conexiones, baja alimentación de los equipos que pudo ser solventada por los mismos equipos y compatibilidad entre el módulo Arduino del cliente 1, del cliente 2, del *Shield Ethernet*, del módulo RFID-RC522, la antena UHF, el zumbador electrónico y lector de huella digital.

El cableado utilizado fue seleccionado gracias a los conocimientos inculcados en el paso de los semestres tomando en cuenta el tipo de red, a que se tiene una corta distancia entre el equipo y el punto de acceso a la red, que el conector compatible para el módulo *shield ethernet* es un RJ45 por lo que se utilizó la norma EIA/TIA 568B que es la más utilizada y un cable UTP categoría 5e que es la que cumple con los requerimientos tanto para la red y el proyecto.

La antena RFID de gama UHF fue esencial para el proyecto ya que permitió el reconocimiento de las etiquetas colocadas en los equipos gracias a la polarización circular utilizada, a que no requiere línea de vista como otros equipos y a que la

comunicación *wiegand* es utilizada con los módulos Arduino y *shield ethernet* facilitando el flujo de la información hacia la base de datos.

4.2. Recomendaciones

En caso de que se pretenda realizar modificaciones o ampliaciones al presente proyecto para trabajos futuros, se recomienda la utilización de *tags* para metales los cuales solo se consiguen por cajas y con un costo elevado, por ende, se aplicaría si fuera utilizado para una mayor cantidad de equipos, pero que vale la pena invertir ya que se obtendría una mayor seguridad de todos los activos porque estos tags son más rápidos de detectar y no interfieren con las estructuras metálicas.

Se debe tener precaución con las etiquetas que se encuentran en los equipos porque son de papel y estos pueden despegarse o dañarse por una mala manipulación, el encargado deberá revisar al momento del préstamo y de la devolución.

También se puede implementar en otras áreas de la ESFOT como para el préstamo de los proyectores ya que es el mismo caso que se está aplicando a los equipos del laboratorio de Tecnología Eléctrica y Electrónica.

Para obtener un proyecto más seguro y confiable se recomienda la utilización de otra antena paralela para obtener un sistema redundante y así evitar que en algún momento los tags no sean leídos debido a alguna falla de lectura de las antenas.

Para economizar costos e incrementar nuevos proyectos se recomienda tener un servidor con todas las prestaciones dentro de la ESFOT para que alojen proyectos como este y no alquilar espacios en otros servidores remotos ya que generarían pérdidas o a su vez instalar estos proyectos de forma local, pero se deberá tener en cuenta que no contaría con las mismas prestaciones o características que se tienen como es el poder ingresar a la información desde cualquier sitio que tenga Internet.

5. BIBLIOGRAFÍA

- [1] Scrum.org, «¿Qué es Scrum?,» [En línea]. Available: <https://www.scrum.org/resources/blog/que-es-scrum>. [Último acceso: 29 Julio 2019].
- [2] Platzi.com, «Qué es SCRUM y los roles en SCRUM,» [En línea]. Available: <https://platzi.com/blog/que-es-scrum-y-los-roles-en-scrum/>. [Último acceso: 29 julio 2019].
- [3] Proyectosagiles.org, «Cliente (Product Owner),» [En línea]. Available: <https://proyectosagiles.org/cliente-product-owner/>. [Último acceso: 29 julio 2019].
- [4] Ceolevel.com, «SCRUM MASTER: ¿Qué es y qué no es?,» [En línea]. Available: <http://www.ceolevel.com/scrum-master-que-es-y-que-no-es>.
- [5] Proyectosagiles.org, «Equipo de desarrollo (Development Team),» [En línea]. Available: <https://proyectosagiles.org/equipo-team/>. [Último acceso: 29 Julio 2019].
- [6] Scrum.org, «¿Qué es un Sprint Backlog?,» [En línea]. Available: <https://www.scrum.org/resources/what-is-a-sprint-backlog>. [Último acceso: 29 julio 2019].
- [7] Blogdelinformatico-reizer.blogspot.com, «QUE ES XAMPP?,» noviembre 2105. [En línea]. Available: <http://blogdelinformatico-reizer.blogspot.com/2015/11/que-es-xampp.html>. [Último acceso: 29 julio 2019].
- [8] Neoattack.com, «¿Qué es el programa MySQL?,» [En línea]. Available: <https://neoattack.com/neowiki/mysql/>. [Último acceso: 29 julio 2019].
- [9] Arsys.es, «¿Qué es Laravel?,» [En línea]. Available: <https://www.arsys.es/blog/programacion/que-es-laravel/>. [Último acceso: 27 julio 2019].
- [10] Tutorialesprogramacionya.com, «¿Qué es Bootstrap?,» [En línea]. Available: <https://www.tutorialesprogramacionya.com/cssya/bootstrapya/detalleconcepto.php?punto=1&codigo=116&inicio=0>. [Último acceso: 20 junio 2019].
- [11] Desarrolloweb.com, «Qué es HTML,» [En línea]. Available:

- <https://desarrolloweb.com/articulos/que-es-html.html>. [Último acceso: 17 junio 2019].
- [12] Desarrolloweb.com, «Qué es CSS,» [En línea]. Available: <https://desarrolloweb.com/articulos/26.php>. [Último acceso: 12 junio 2019].
- [13] Maestrosdelweb.com, «¿Qué es Javascript?,» [En línea]. Available: <http://www.maestrosdelweb.com/que-es-javascript/>. [Último acceso: 28 julio 2019].
- [14] Dipolerfid.es, «Tecnología RFID,» [En línea]. Available: <https://www.dipolerfid.es/es/tecnologia-RFID>. [Último acceso: 17 marzo 2019].
- [15] Blog.atlasrfidstore.com, «Mapa vial,» [En línea]. Available: <https://blog.atlasrfidstore.com/learn-uhf-rfid>. [Último acceso: 15 Junio 2019].
- [16] Evaluandoerp.com, «Comparación entre tecnologías de RFID y código de barras,» 2018. [En línea]. Available: <https://www.evaluandoerp.com/comparacion-tecnologias-rfid-codigo-barras/?fbclid=IwAR14kTCsWvuBjmb5KzdyYVmNamFLXOWucBKx-drVsKhdbOnvz9txB5z-ovM>. [Último acceso: 20 junio 2019].
- [17] www.blog.330ohms.com, «¿Cuál es la diferencia entre Arduino UNO y Arduino Leonardo?,» [En línea]. Available: <https://blog.330ohms.com/2016/02/03/cual-es-la-diferencia-entre-arduino-uno-y-arduino-leonardo/>. [Último acceso: 14 agosto 2019].
- [18] Hetpro-store.com, «Módulo RFID-RC522 RF con Arduino UNO SPI,» [En línea]. Available: <https://hetpro-store.com/TUTORIALES/módulo-lector-rfid-rc522-rf-con-arduino/>. [Último acceso: 20 junio 2019].
- [19] Panamahitek.com, «¿Cómo funciona el protocolo SPI?,» [En línea]. Available: <http://panamahitek.com/como-funciona-el-protocolo-spi/>. [Último acceso: 20 junio 2019].
- [20] Tarjetashid-mifare-rfid.com, «Tecnología Wiegand,» [En línea]. Available: <http://www.tarjetashid-mifare-rfid.com/wiegand.html>. [Último acceso: 21 junio 2019].
- [21] Aprendiendoarduino.wordpress.com, «Ethernet Shield,» [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2016/07/04/ethernet-shield/>.
- [22] Aprendiendoarduino.wordpress.com, «Aprendiendo Arduino,» 04 julio 2016. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2016/07/04/ethernet->

shield/. [Último acceso: 21 junio 2019].

- [23] Panamahitek.com, «MIFARE RC522 + Arduino,» [En línea]. Available: <http://panamahitek.com/mifare-rc522-arduino/>. [Último acceso: 3 junio 2019].
- [24] Community.createlabz.com, «Sensor Óptico De Huella Digital Con Arduino,» [En línea]. Available: <https://community.createlabz.com/knowledgebase/r307-optical-fingerprint-sensor/>. [Último acceso: 4 junio 2019].
- [25] Monkeyboard, «Wiegand-Protocol-Library-for-Arduino,» [En línea]. Available: <https://github.com/monkeyboard/Wiegand-Protocol-Library-for-Arduino?fbclid=IwAR36HlxcYuK776dQlxW-VNgrttwINtv1jsalpps19VYulwZXgVRLDRB8zMA>. [Último acceso: 5 abril 2019].
- [26] Aprendiendoarduino.wordpress.com/, «Librería Ethernet,» [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2016/07/05/libreria-ethernet/>. [Último acceso: 8 junio 2019].
- [27] Infootec.net, «Arduino Uno R3,» [En línea]. Available: https://www.infootec.net/arduino/?fbclid=IwAR2pmjOMzIJvhruPMQhbKxJRZr3XPA6QIQI3MXJXmk9CHRU6d_GCDIooAUg. [Último acceso: 23 marzo 2019].
- [28] Instructables.com, «Lector De Huella Digital Arduino,» [En línea]. Available: <https://www.instructables.com/id/Lector-De-Huella-Digital-Arduino/>. [Último acceso: 28 mayo 2019].
- [29] Aprendiendoarduino.wordpress.com, «Aprendiendo Arduino,» <https://aprendiendoarduino.wordpress.com/category/bus-spi/>, [En línea]. Available: Aprendiendo Arduino. [Último acceso: 2 junio 2019].

ANEXOS

ÍDICE DE ANEXOS

Anexo I Historias de usuario.....	64
Anexo II Diagrama completo de la conexión de los dos clientes.....	67
Anexo III Código completo del cliente 1.....	68
Anexo IV Código completo del cliente 2.....	76
Anexo V Manual de instalación.....	79
Anexo VI Manual de usuario.....	85

6. ANEXOS

Anexo I Historias de usuario

Tabla 6.1 Módulo estudiante.

Historia de Usuario	
Identificador (ID): HU-001	Usuario: Administrador
Nombre de Historia: Módulo de Estudiante Registro y eliminación	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Baja
Programador(es) Responsable(es): Diego Chicaiza	
<p>Descripción: Para el registro del estudiante se deberá contar con número de cédula, nombres, apellidos, correo electrónico, teléfono, carrera y huella dactilar.</p> <p>Se podrá eliminar al estudiante acorde al requerimiento del administrador</p>	
<p>Observaciones: Al momento de registrar un estudiante es obligatorio la huella dactilar del estudiante</p>	

Tabla 6.2 Módulo equipo.

Historia de Usuario	
Identificador (ID): HU-002	Usuario: Administrador
Nombre de Historia: Módulo de Equipos Registro y eliminación	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Baja
Programador(es) Responsable(es): Diego Chicaiza	
<p>Descripción: Para el registro de equipos deberán contar con el nombre, descripción corta, código UHF, código RFID y un costo aproximado del equipo</p> <p>Se podrá eliminar el equipo acorde a la conveniencia del administrador.</p>	
<p>Observaciones: Al momento de registrar un equipo es obligatorio los códigos UHF y RFID</p>	

Tabla 6.3 Módulo usuarios.

Historia de Usuario	
Identificador (ID): HU-003	Usuario: Administrador
Nombre de Historia: Módulo de usuarios registro y eliminación	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Baja
Programador(es) Responsable(es): Diego Chicaiza	
<p>Descripción: Para el registro los usuarios deberán contar con número de cédula, nombres, apellidos, correo electrónico y un rol (administrador o encargado del laboratorio)</p> <p>Se podrá eliminar al usuario acorde a la conveniencia del administrador</p>	
<p>Observaciones: Al momento de registrar un usuario es obligatorio tener un rol</p>	

Tabla 6.4 Préstamo equipos.

Historia de Usuario	
Identificador (ID): HU-003	Usuario: Administrador/ Encargado
Nombre de Historia: Préstamo de equipos	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Baja
Programador(es) Responsable(es): Diego Chicaiza	
<p>Descripción: Para poder realizar el préstamo del equipo el estudiante debe colocar su huella en el lector biométrico y como ya se encuentra registrado se llenarán los campos automáticamente.</p> <p>De la misma forma el equipo deberá ser registrado en el lector con su respectivo <i>tag</i> de RFID y se llenará automáticamente los datos del equipo.</p>	
<p>Observaciones: Para poder realizar los préstamos debe estar registrado el estudiante y el equipo.</p>	

Tabla 6.5 Devolución del equipo.

Historia de Usuario	
Identificador (ID): HU-003	Usuario: Administrador/ Encargado
Nombre de Historia: Devolución de equipos	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Baja
Programador(es) Responsable(es): Diego Chicaiza	
Descripción: Para la devolución de los equipos el responsable del laboratorio deberá pasar el equipo por el lector de RFID llenando de manera automática los datos del equipo y del usuario.	
Observaciones: El equipo debe estar con todas sus etiquetas para la devolución	

Anexo III Código completo del cliente 1

```
1. #include <SPI.h>
2. #include <Ethernet.h>
3. #include <MFRC522.h>
4. #include <Adafruit_Fingerprint.h>
5. #define SS_PIN 8
6. #define RST_PIN 9
7. MFRC522 rfid(SS_PIN, RST_PIN);
8. MFRC522::MIFARE_Key key;
9. String mensaje = "";
10. int positivo = 0;
11. byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
12. IPAddress ip(172, 31, 116, 232);
13. IPAddress subnet(255, 255, 0, 0);
14. IPAddress gateway(172, 31, 116, 193);
15. IPAddress server(209,126,124,203);
16. EthernetClient client;
17. SoftwareSerial mySerial(2, 3);
18. Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
19. int id_finding;
20. int id;
21. int temp;
22. char resp;
23. void setup() {
24.   Serial.begin(9600);
25.   while (!Serial);
26.   SPI.begin();
27.   rfid.PCD_Init();
28.   for (byte i = 0; i < 6; i++) {
29.     key.keyByte[i] = 0xFF;
30.   }
31.   Serial.print(F("RFID inicializado, key:"));
32.   Serial.println();
33.   delay(1000);
34.   Ethernet.begin(mac, ip);
```

```

35. Serial.print("Ethernet inicializado en la direccion: ");
36. Serial.println(Ethernet.localIP());
37. delay(1000);
38. finger.begin(57600);
39. if (finger.verifyPassword()) {
40. Serial.println("Sensor biometrico encontrado!");
41. } else {
42. Serial.println("No se encontro el sensor biometrico :(");
43. while (1) { delay(1); }
44. }
45. delay(1000);
46. pinMode(5,OUTPUT);
47. pinMode(6,OUTPUT);
48. pinMode(7,OUTPUT);
49. digitalWrite(7,LOW);
50. }
51. void loop() {
52. id_finding = getFingerprintIDez();
53. if(id_finding==0){
54. Serial.println("Estudiante no registrado");
55. mensaje=String(id_finding);
56. mensaje = "/registrar?registrar=none" + mensaje;
57. digitalWrite(5,HIGH);
58. delay(300);
59. digitalWrite(5,LOW);
60. delay(300);
61. digitalWrite(5,HIGH);
62. delay(300);
63. digitalWrite(5,LOW);
64. delay(300);
65. digitalWrite(5,HIGH);
66. delay(300);
67. digitalWrite(5,LOW);
68. resp = enviar_http(mensaje);
69. Serial.print("Respuesta no registrado: ");
70. Serial.println(resp);

```

```

71. if (resp=='1'){
72. finger.getTemplateCount();
73. id = finger.templateCount +1;
74. Serial.print("Resgistrando en el ID: ");
75. Serial.println(id);
76. if(getFingerprintEnroll(id)==1){
77. digitalWrite(5,HIGH);
78. delay(500);
79. digitalWrite(5,LOW);
80. }
81. }
82. if(resp=='2'){
83. finger.emptyDatabase();
84. Serial.println("Borradas todas las huellas");
85. digitalWrite(5,HIGH);
86. delay(5000);
87. digitalWrite(5,LOW);
88. }
89. }else if(id_finding>0){
90. digitalWrite(5,HIGH);
91. delay(1000);
92. digitalWrite(5,LOW);
93. mensaje = String(id_finding);
94. mensaje = "/huella?huella="+mensaje;
95. Serial.println("ID del estudiante: "+
96. mensaje);
97. resp = enviar_http(mensaje);
98. Serial.print("Respuesta: ");
99. Serial.println(resp);
100.     if(resp=='2'){
101.         finger.emptyDatabase();
102.         Serial.println("Borradas todas las huellas");
103.         digitalWrite(5,HIGH);
104.         delay(3000);
105.         digitalWrite(5,LOW);
106.     }

```

```

107.     }
108.     delay(50);
109.     if ( ! rfid.PICC_IsNewCardPresent())
110.         return;
111.     if ( ! rfid.PICC_ReadCardSerial())
112.         return;
113.     digitalWrite(5,HIGH);
114.     delay(1000);
115.     digitalWrite(5,LOW);
116.     Serial.print(F("PICC type: "));
117.     MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
118.     Serial.println(rfid.PICC_GetTypeName(piccType));
119.     mensaje = leer_rfid(rfid.uid.uidByte, rfid.uid.size);
120.     mensaje = "/rfid?rfid="+mensaje;
121.     Serial.print("Lectura rfid: ");
122.     Serial.println(mensaje);
123.     delay(1000);
124.     resp = enviar_http(mensaje);
125.     Serial.print("Respuesta: ");
126.     Serial.println(resp);
127.     if(resp=='1'){
128.         Serial.println("Respuesta positiva");
129.     }
130.     rfid.PICC_HaltA();
131.     rfid.PCD_StopCrypto1();
132.     }
133.     char enviar_http(String mensaje_eth){
134.         char c;
135.         positivo = 0;
136.         delay(1000);
137.         Serial.print("Conectando al servidor: ");
138.         Serial.print(server);
139.         Serial.println("...");
140.         if (client.connect(server, 8011)) {
141.             Serial.println(mensaje_eth);
142.             Serial.print("Enviando: ");

```

```

143.     mensaje_eth="GET "+mensaje+" HTTP/1.1";
144.     Serial.println(mensaje_eth);
145.     client.println(mensaje_eth);
146.     client.println("Host: www.google.com");
147.     client.println("Connection: close");
148.     client.println();
149.     delay(2000);
150.     while (client.available()) {
151.         c = client.read();
152.         if (c==':'){
153.             positivo++;
154.         }else if(c!=':'){
155.             positivo = 0;
156.         }
157.         if (positivo==2){
158.             c = client.read();
159.             break;
160.         }
161.     }
162.     delay(2000);
163.     client.stop();
164.     } else {
165.     Serial.println("conexion fallida");
166.     client.stop();
167.     }
168.     return c;
169.     }
170.     String leer_rfid(byte *buffer, byte bufferSize) {
171.     String lectura="";
172.     for (byte i = 0; i < bufferSize; i++) {
173.     lectura = lectura + buffer[i];
174.     }
175.     return lectura;
176.     }
177.     int getFingerprintIDez() {
178.     uint8_t p = finger.getImage();

```

```

179.     if (p != FINGERPRINT_OK) return -1;
180.     p = finger.image2Tz();
181.     if (p != FINGERPRINT_OK) return -1;
182.     p = finger.fingerFastSearch();
183.     if (p != FINGERPRINT_OK) return 0;
184.     return finger.fingerID;
185.     }
186.     uint8_t getFingerprintEnroll(int id_enroll) {
187.     int p = -1;
188.     Serial.println("Coloque su dedo");
189.     digitalWrite(5,HIGH);
190.     temp=0;
191.     while(temp<5000){
192.     temp++;
193.     p = finger.getImage();
194.     switch (p) {
195.     case FINGERPRINT_OK:
196.     Serial.println("Huella leida");
197.     digitalWrite(5,LOW);
198.     break;
199.     default:
200.     break;
201.     }
202.     if (p == FINGERPRINT_OK) {
203.     break;
204.     }
205.     }
206.     if (temp>=5000){
207.     Serial.println("Termino el tiempo");
208.     digitalWrite(5,LOW);
209.     return 0;
210.     }
211.     p = finger.image2Tz(1);
212.     switch (p) {
213.     case FINGERPRINT_OK:
214.     Serial.println("Imagen convertida");

```

```

215.     break;
216.     default:
217.         Serial.println("Error convirtiendo imagen");
218.         return p;
219.     }
220.     Serial.println("Retire el dedo");
221.     delay(2000);
222.     Serial.println("Coloque el dedo nuevamente");
223.     digitalWrite(5,HIGH);
224.     temp = 0;
225.     while (temp<5000) {
226.         temp++;
227.         p = finger.getImage();
228.         if(p == FINGERPRINT_OK){
229.             Serial.println("Huella leida");
230.             digitalWrite(5,LOW);
231.             break;
232.         }
233.     }
234.     if (temp==5000){
235.         Serial.println("Termino el tiempo");
236.         return 0;
237.     }
238.     p = finger.image2Tz(2);
239.     switch (p) {
240.         case FINGERPRINT_OK:
241.             Serial.println("Imagen convertida");
242.             break;
243.         default:
244.             Serial.println("Error convirtiendo imagen");
245.             return p;
246.     }
247.     Serial.print("Creando modelo para el ID: ");
248.     Serial.println(id_enroll);
249.     p = finger.createModel();
250.     if (p == FINGERPRINT_OK) {

```

```
251.     Serial.println("Modelo correcto!");
252.     } else {
253.     Serial.println("Error creando modelo");
254.     return p;
255.     }
256.     p = finger.storeModel(id_enroll);
257.     if (p == FINGERPRINT_OK) {
258.     Serial.println("Guardado!");
259.     digitalWrite(5,HIGH);
260.     delay(500);
261.     digitalWrite(5,LOW);
262.     } else {
263.     Serial.println("Error guardando");
264.     return p;
265.     }
266.     return 1;
267.     }
```

Anexo IV Código completo del cliente 2

```
1. #include <Wiegand.h>
2. #include <SPI.h>
3. #include <Ethernet.h>
4. WIEGAND wg;
5. String mensaje = "";
6. String mensaje2="";
7. byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
8. IPAddress ip(172, 31, 116, 229);
9. IPAddress gateway(172, 31,4, 2);
10. IPAddress myDns(172, 31,4, 2);
11. IPAddress server(209,126,124,203);
12. EthernetClient client;
13. char resp;
14. int positivo = 0;
15. void setup() {
16. Serial.begin(9600);
17. while (!Serial);
18. wg.begin();
19. Serial.println("Lector UFH inicializado");
20. delay(1000);
21. Ethernet.begin(mac, ip, myDns, gateway);
22. Serial.println("Ethernet inicializado");
23. delay(100);
24. }
25. void loop() {
26. if(wg.available())
27. {
28. mensaje = wg.getCode();
29. Serial.print("Code: ");
30. Serial.print(mensaje);
31. Serial.print(", Type: ");
32. Serial.println(wg.getWiegandType());
33. mensaje2 = "/uhf?uhf="+mensaje;
34. resp = enviar_http(mensaje2);
```

```

35. Serial.print("Respuesta: ");
36. Serial.println(resp);
37. if(resp=='0'){
38. alarma();
39. Serial.println("Alarma activada");
40. }else if((resp=='1')){
41. Serial.println("Salida permitida");
42. }else{
43. delay(2000);
44. mensaje2 = "/uhftemp?uhftemp="+mensaje;
45. resp = enviar_http(mensaje2);
46. }
47. }
48. delay(100);
49. }
50. char enviar_http(String mensaje_eth){
51. char c;
52. Serial.print("Conectando al servidor: ");
53. Serial.print(server);
54. Serial.println("...");
55. if (client.connect(server, 8011)) {
56. Serial.println(mensaje_eth);
57. Serial.print("Enviando: ");
58. mensaje_eth="GET "+mensaje_eth+" HTTP/1.1";
59. Serial.println(mensaje_eth);
60. client.println(mensaje_eth);
61. client.println("Connection: close");
62. client.println();
63. delay(500);
64. while (client.available()) {
65. c = client.read();
66. if (c==:){
67. positivo++;
68. }else if(c!=:){
69. positivo = 0;
70. }

```

```
71. if (positivo==2){
72. c = client.read();
73. break;
74. }
75. }
76. Serial.println("Desconectado del servidor");
77. client.stop();
78. } else {
79. Serial.println("conexion fallida");
80. client.stop();
81. }
82. return c;
83. }
84. void alarma(){
85. int nota = 1319;
86. tone(8, nota, 300);
87. delay(300);
88. tone(8, nota, 300);
89. delay(300);
90. tone(8, nota, 300);
91. delay(300);
92. tone(8, nota, 300);
93. delay(300);
94. tone(8, nota, 300);
95. delay(300);
96. noTone(8);
97. }
```

Anexo V Manual de instalación

Instalación del programa localmente

- **Requerimientos de Hardware**

OIS: Windows 7, Windows 10, Mac OS X 10.13 (High Sierra), Mac OS X 10.11 (El Capitán).

RAM: 1GB en adelante.

Procesador: Dual Core en adelante.

Disco Duro: 2GB de espacio para el programa y 500MB de espacio libre para almacenamiento de los datos.

- **Requerimientos de Software**

Para poder ejecutar los comandos de instalación del software se debe instalar previamente GIT que es una herramienta que funciona en la terminal o línea de comando tanto para sistemas operativos Windows e IOS como se observa en la figura 6.2. A continuación, el enlace de la descarga de la herramienta GIT: "<https://git-scm.com/downloads>".



Figura 6.2 Página de descarga de la herramienta GIT.

Para la base de datos se deberá instalar XAMPP que se encuentra disponible para sistemas operativos Windows e IOS, también debe contar con la versión 7.0 de PHP en adelante como se observa en la figura 6.3. A continuación, el enlace de la descarga: "<https://www.apachefriends.org/es/download.html>".



Figura 6.3 Página de descarga de la herramienta XAMPP.

- **Instalación del *software* del proyecto**

Para poder hacer uso de la aplicación *web* se debe principalmente copiar la carpeta de *software* del proyecto en una unidad de disco que puede ser (C:) o (D:) o de preferencia en un directorio que no pueda ser manipulado fácilmente (véase figura 6.4).

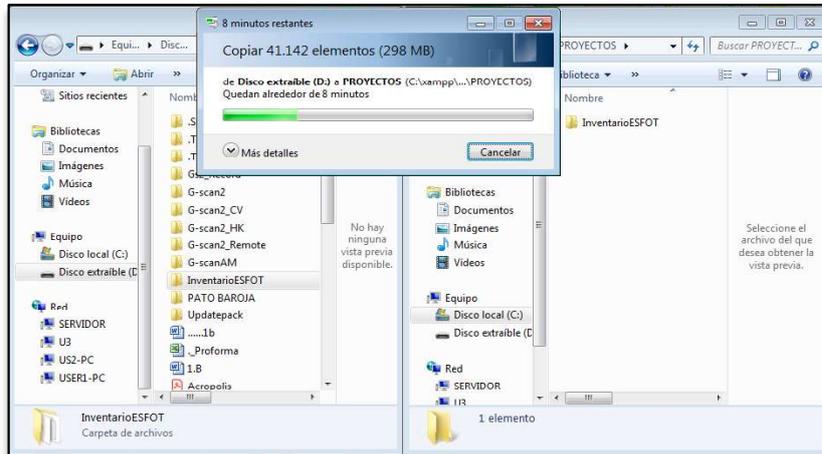


Figura 6.4 Instalación de la carpeta en una unidad de disco.

- **Configuración del programa**

Para hacer uso de la aplicación *web* se debe tener previamente instalado las herramientas GIT, XAMPP y tener copiado la carpeta del *software* del proyecto en la unidad de disco de preferencia del usuario, con ello se precederá a los siguientes pasos.

Instalación y ejecución de la base de datos

Para la instalación de la base de datos se debe ingresar al panel de XAMPP y activar la casilla MySQL como se observa en la Figura 6.5.

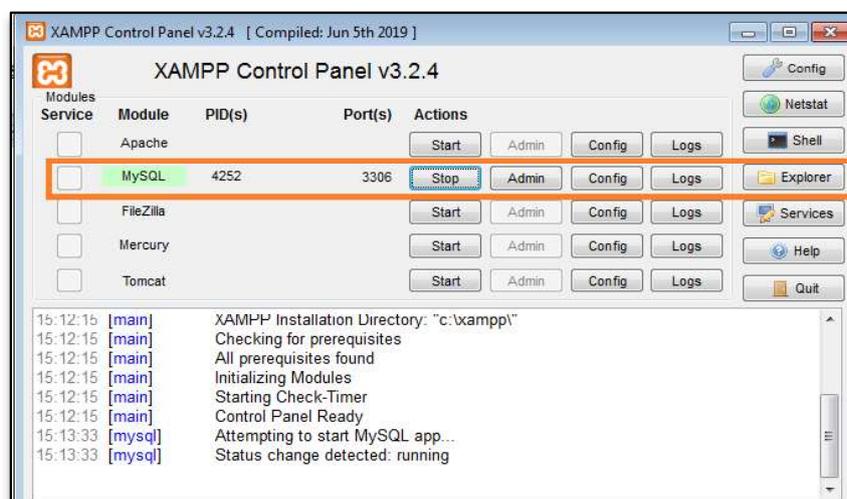


Figura 6.5 Activación del módulo MySQL.

Activada la casilla de MySQL se deberá abrir el navegador de preferencia del usuario e ingresar la siguiente dirección `http://localhost/phpmyadmin/` y luego dar clic en la opción nuevo para poder crear la base de datos como se puede ver en la figura 6.6.

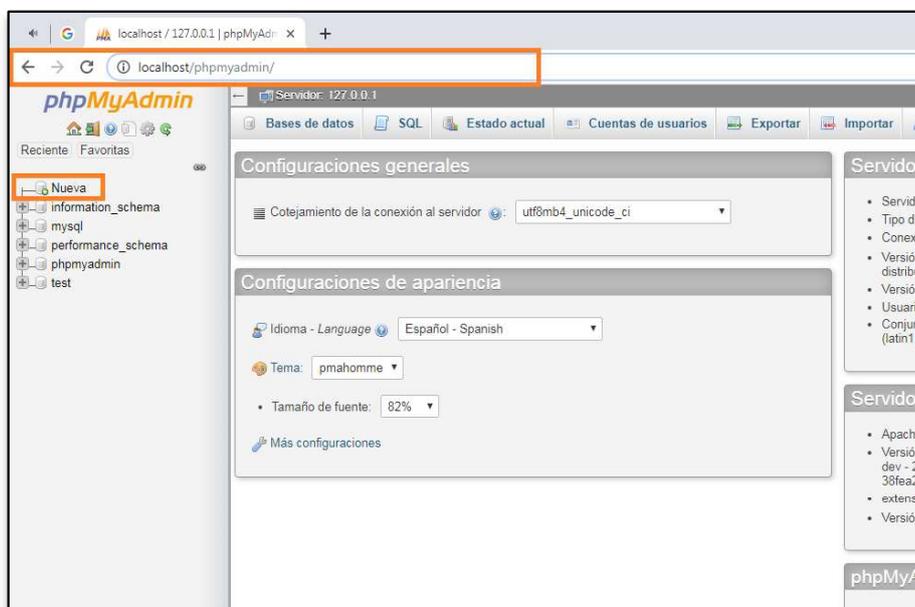


Figura 6.6 Creación de una nueva base de datos.

Después se colocará inventarioepn para el nombre de la base de datos y seguidamente se dará clic en la opción importar, véase figura 6.7.

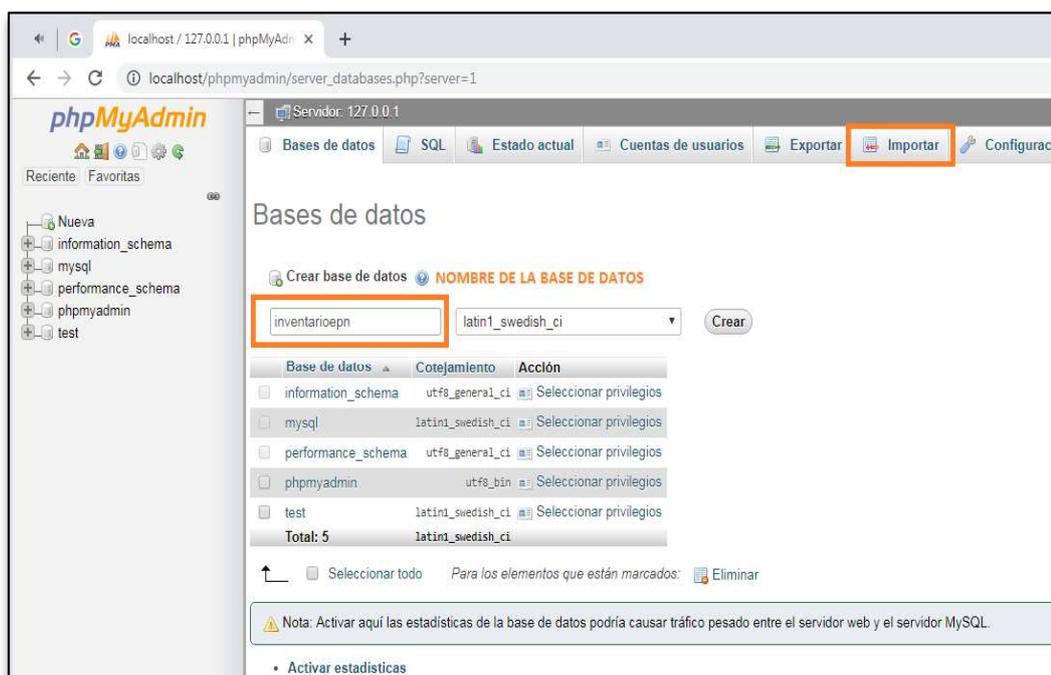


Figura 6.7 Importación de la base de datos.

Luego se deberá seleccionar el archivo con extensión SQL que se encuentra en la carpeta del *software* que tiene el nombre de *inventarioepn.sql* y luego se dará clic en abrir de acuerdo con la figura 6.8.

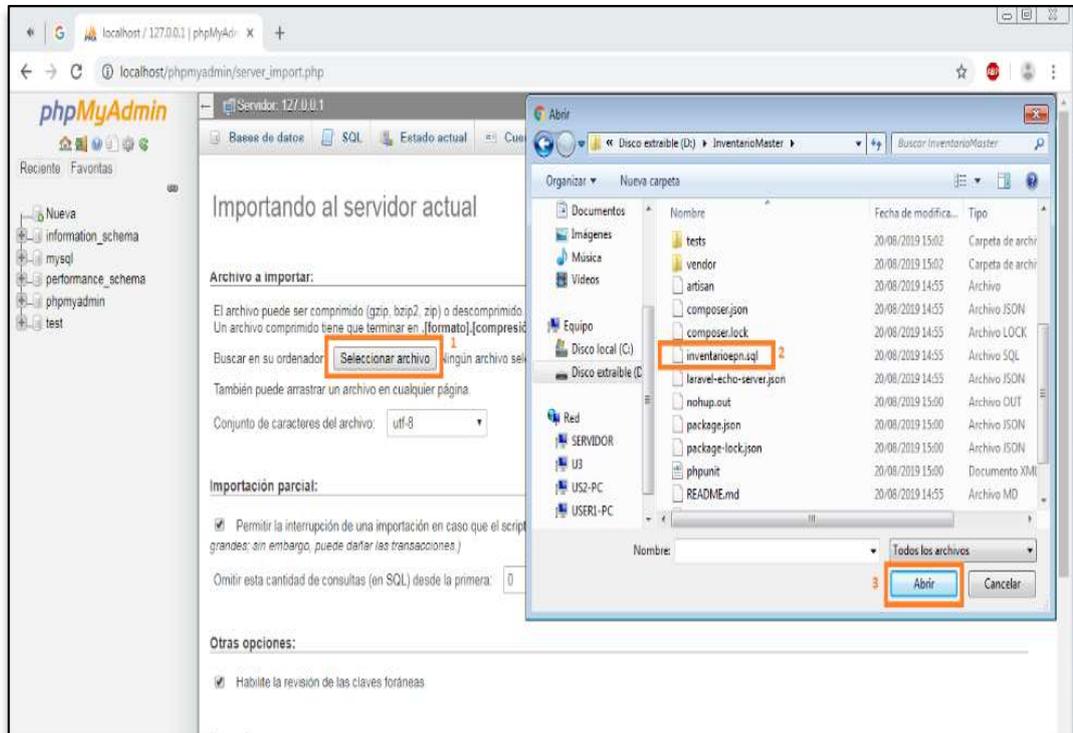


Figura 6.8 Selección del archivo inventarioepn.sql.

Para finalizar se dará clic en continuar quedando cargada la base de datos como se observa en la figura 6.9.

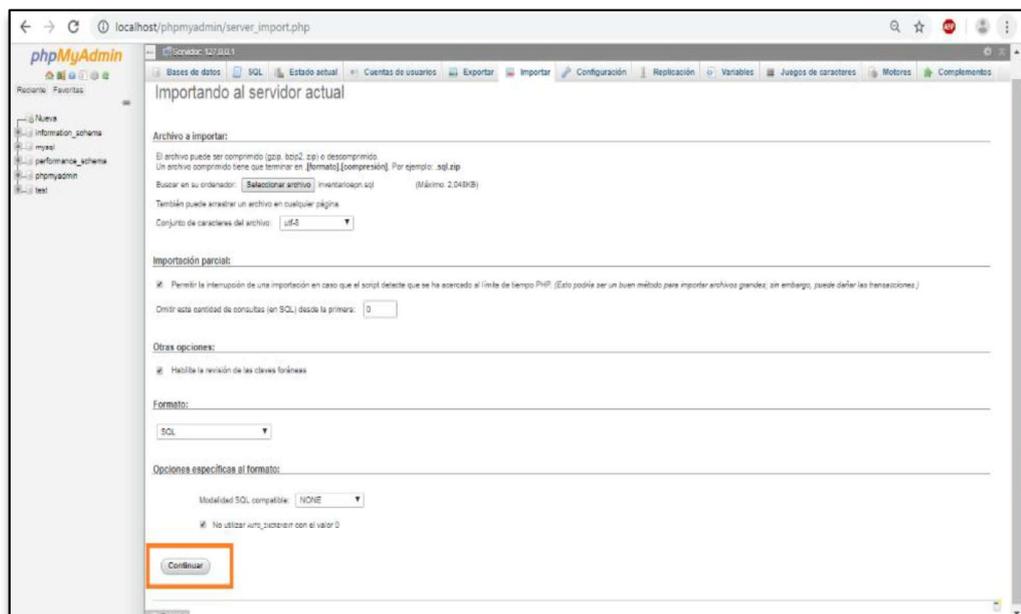


Figura 6.9 Base de datos precargada.

Luego se tendrá la base de datos creada para poder usar con la aplicación web, véase figura 6.10.

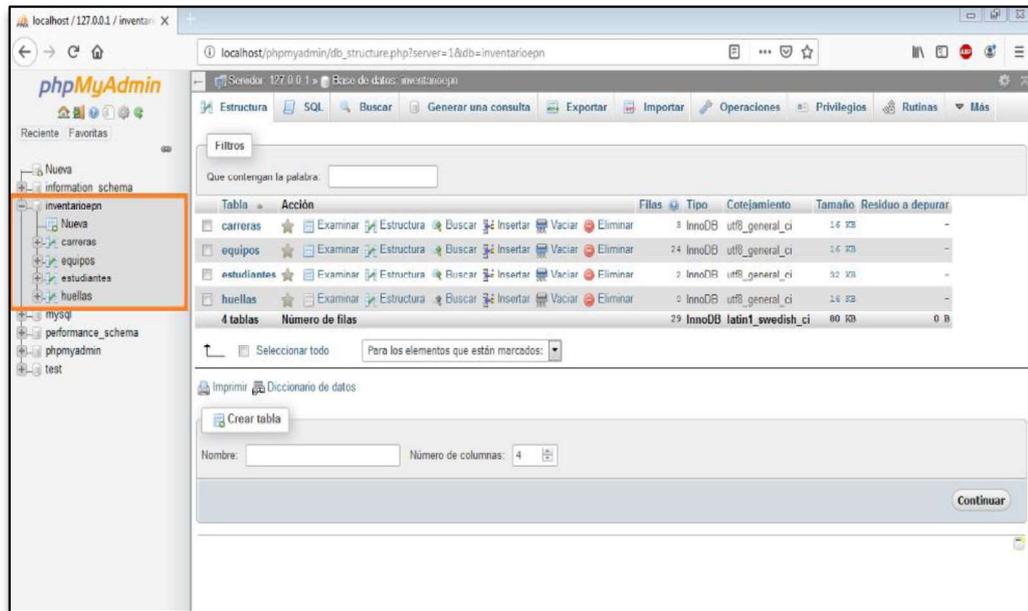


Figura 6.10 Base de datos de la aplicación web.

Ejecución y configuración de la aplicación web

Primero se debe abrir la aplicación GIT para poder arrancar el programa mediante línea de comando como se observa en la figura 6.11.

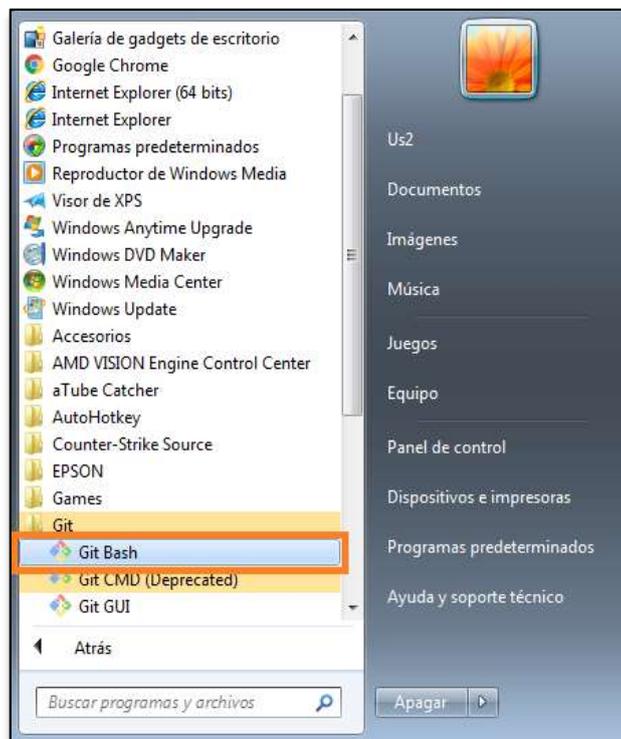
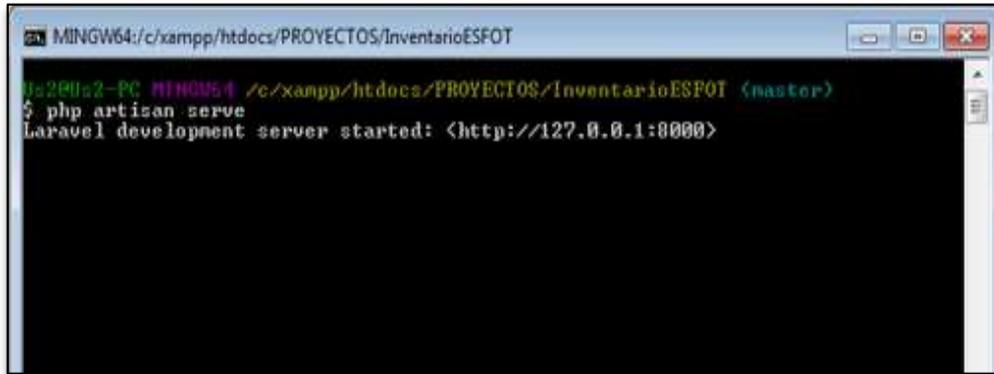


Figura 6.11 Abriendo la herramienta GIT.

Dentro de la ventana de GIT se colocará la dirección de la carpeta que contiene el *software* en donde se encuentra ubicada la aplicación *web* y seguidamente se ingresará el comando **php artisan serve** y se dará enter para así iniciar la ejecución del programa de acuerdo con la figura 6.12.



```
MINGW64: c:/xampp/htdocs/PROYECTOS/InventarioESPOT
C:\xampp\htdocs\PROYECTOS\InventarioESPOT (master)
$ php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

Figura 6.12 Ventana de línea de comando GIT.

Finalmente se deberá abrir un navegador de preferencia del usuario e ingresar la dirección `1xx.xxx.xxx:xxx:8xxx` y se podrá dar uso a la aplicación *web* como se observa en la figura 6.13.

El sistema funcionará correctamente siempre y cuando se haya quemado el programa modificado en los módulos Arduino para el cliente1 y cliente 2 con sus respectivas configuraciones de las direcciones IP y con su respectivo puerto, las que dependerán de la red que se vaya a usar para el servidor y los puntos de conexión de cada cliente con un puerto 8xxx que es del servidor local.

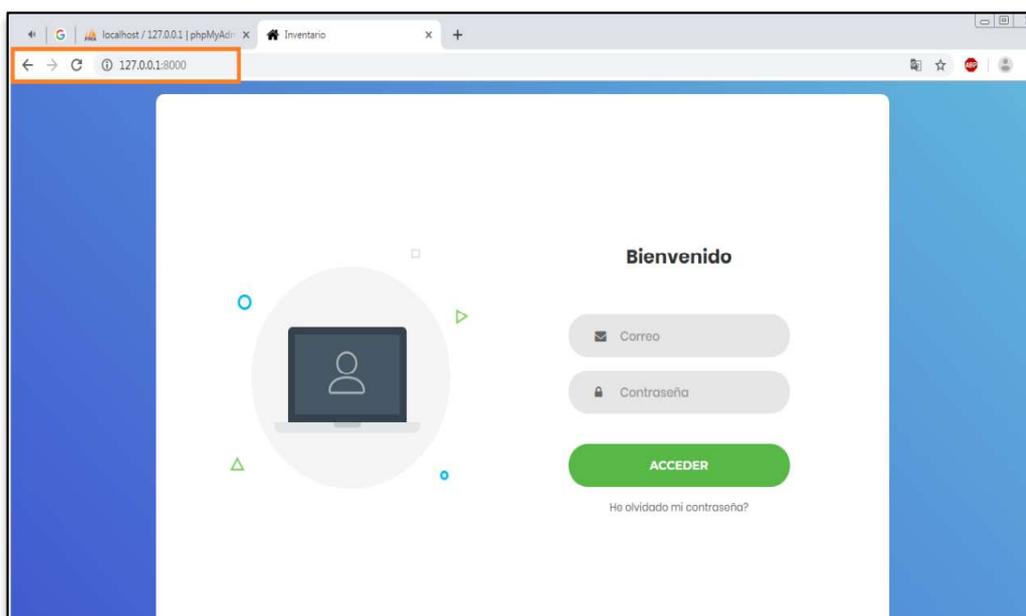


Figura 6.13 Aplicación *web* lista para ingresar.

Anexo VI Manual de usuario

1. Abrir un navegador, digitar la dirección IP en la que se encuentra remotamente subido el proyecto seguido de dos puntos y el número de puerto por el cual el servidor HTTP escucha la petición del cliente (xxx.xxx.xxx.xxx:8xxx) y dar enter, haciendo que se visualice el login principal del sistema como se observa en la figura 6.14.

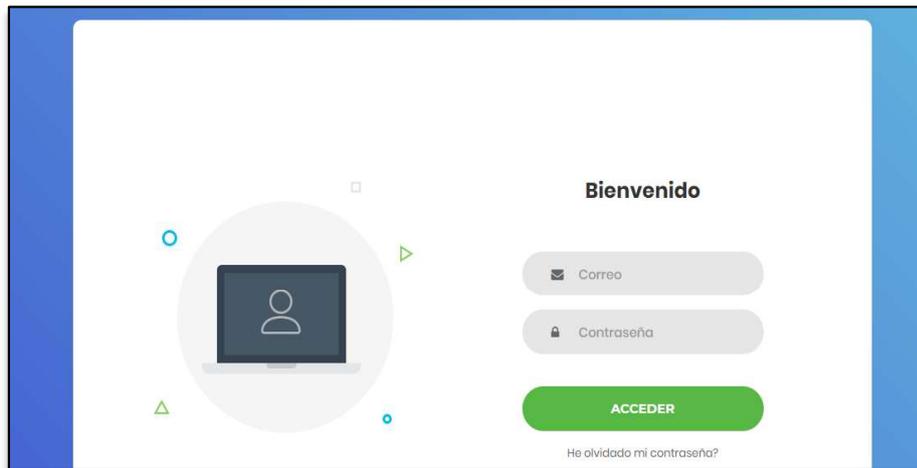


Figura 6.14 Navegador con la dirección 2xx.xxx.xxx.xxx:80xx.

2. Ingresar las credenciales de usuario y su respectiva contraseña. Si los datos ingresados son correctos pasará a la siguiente página que permitirá ver las pestañas del menú como son usuario, estudiante, equipo y préstamo como se observa en la figura 6.15.

Nota: la pestaña usuario solo se puede visualizar y manipular si se ingresó como administrador.

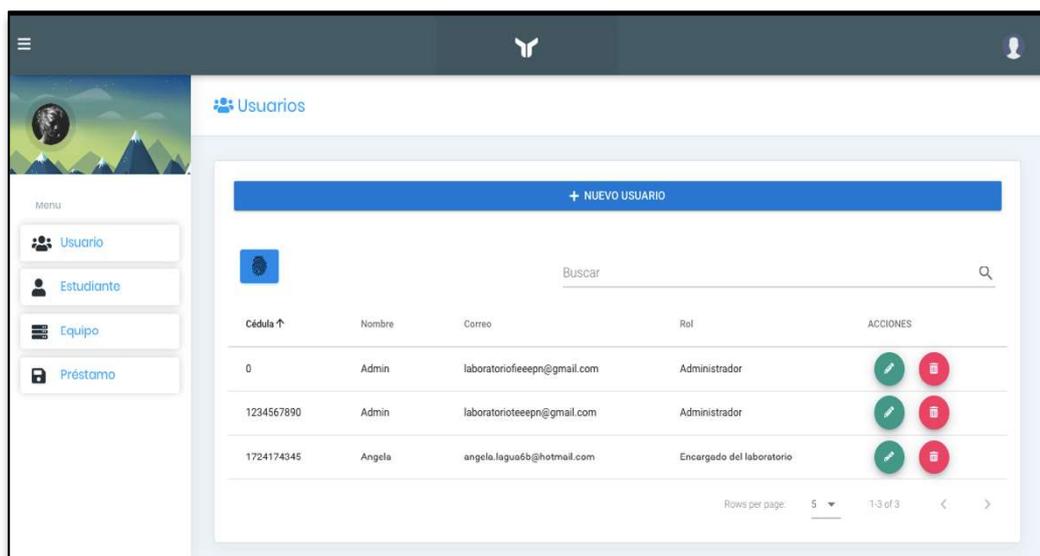


Figura 6.15 Menú del sistema y pestaña de usuario.

3. En la pestaña usuario se puede ver a las personas que se encuentran como administradores o encargados de laboratorio y a su vez eliminarlas o añadir a alguien más. En esta pestaña se cuenta con dos botones que se describen a continuación y se pueden observar en la Figura 6.15.

a) Eliminar huellas

El botón Limpiar huella, permite borrar todas las huellas con un ID asignado que se encuentren en la base de datos, por lo que, se sugiere solo realizar cuando sea necesario y para que el lector de huellas vuelva a asignar los ID desde el número 1.

b) Agregar un administrador o encargado de laboratorio

Al dar clic en la barra azul que dice +nuevo usuario, seguidamente se abre una ventana para ingresar los datos (figura 6.16), después de llenar todos los campos se procede a dar clic en el ícono de guardar y si esta actividad se realizó correctamente sale un mensaje de registro guardado al cual solo deberá dar clic en OK para que desaparezca, véase figura 6.17.

Nota: Recordar el número de cédula y el correo electrónico ingresado porque estos serán utilizados como su usuario y contraseña.

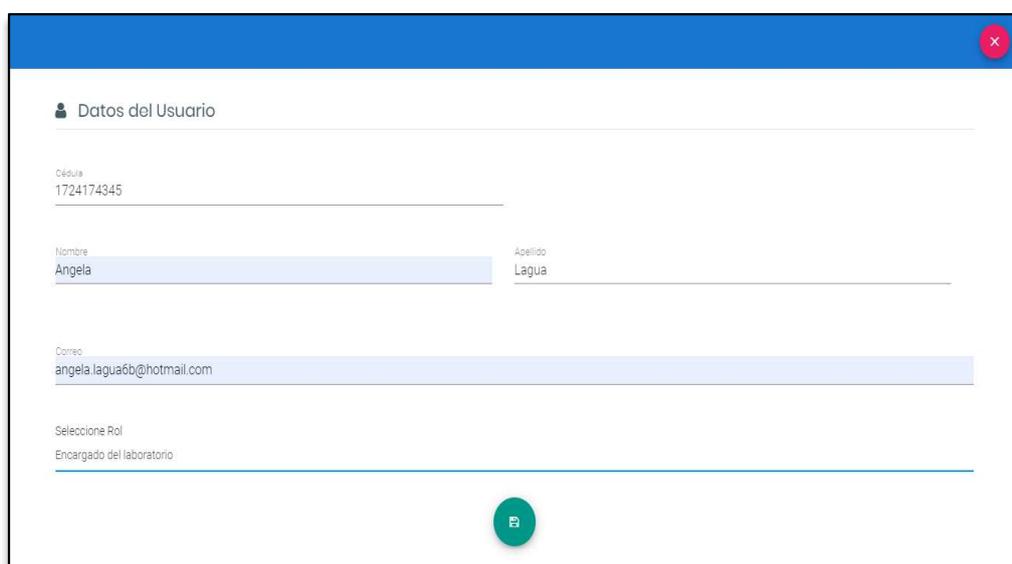
The image shows a web form titled "Datos del Usuario" with a blue header bar and a red close button in the top right corner. The form contains several input fields: "Cédula" with the value "1724174345", "Nombre" with "Angela", "Apellido" with "Lagua", and "Correo" with "angela.lagua6b@hotmail.com". At the bottom, there is a "Seleccione Rol" dropdown menu with "Encargado del laboratorio" selected. A green circular button with a white document icon is centered at the bottom of the form.

Figura 6.16 Ventana para nuevo usuario.

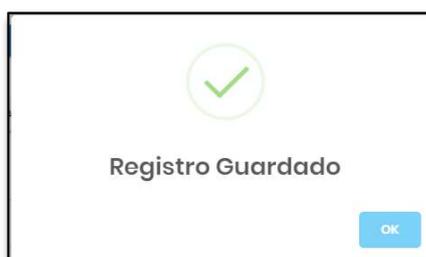


Figura 6.17 Mensaje de registro correcto.

4. En la pestaña estudiante se puede observar a las personas que se encuentran registradas con sus datos personales, quienes son las que pueden realizar préstamos o devoluciones y a su vez estas pueden ser eliminadas o añadir a alguien más con un botón azul que se describe a continuación en la figura 6.18.

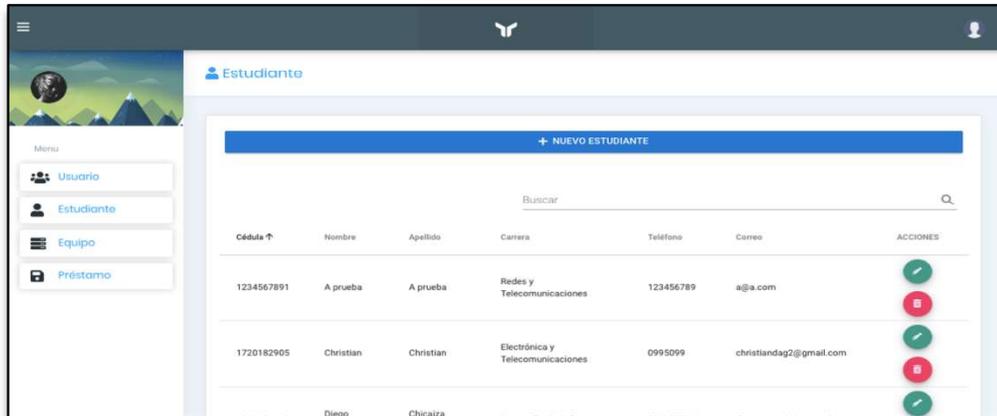


Figura 6.18 Pestaña Estudiante.

a) Registrar un nuevo estudiante

Al dar clic en la barra azul que dice nuevo estudiante, seguidamente se abre una ventana para ingresar los datos personales, después se debe dar clic a este ícono  y colocar el dedo pulgar de la persona a registrarse sobre el lector de huella digital hasta que el led se encienda, de ahí retirar por 2 segundos y volver a colocar el dedo, esta actividad de colocar el dedo y retirar se debe realizar por 4 veces para que se asigne un nuevo número de ID, seguidamente se observar el número de ID asignado como se puede ver en la figura 6.19 o caso contrario deberá volver a realizar este procedimiento, luego se dará clic en guardar y si todo es correcto sale un mensaje de registro guardado al cual solo deberá dar clic en OK para que desaparezca.

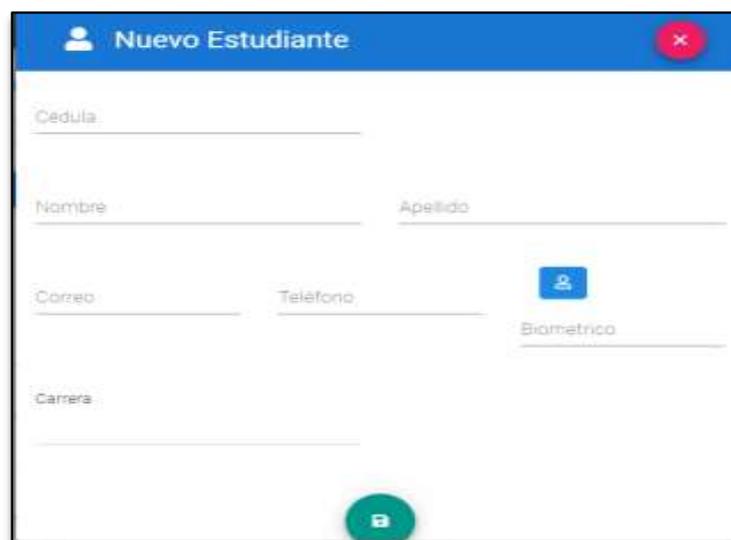


Figura 6.19 Ventana para nuevo estudiante.

5. En la pestaña equipo se puede observar los equipos que se encuentran en el inventario con sus respectivos datos, estado, precio, etiquetas y fecha. Estos equipos son los que se pueden prestar si su estado es disponible, se pueden devolver si su estado es prestado y a su vez se pueden eliminar o añadir con un botón azul que se describe a continuación, véase figura 6.20.

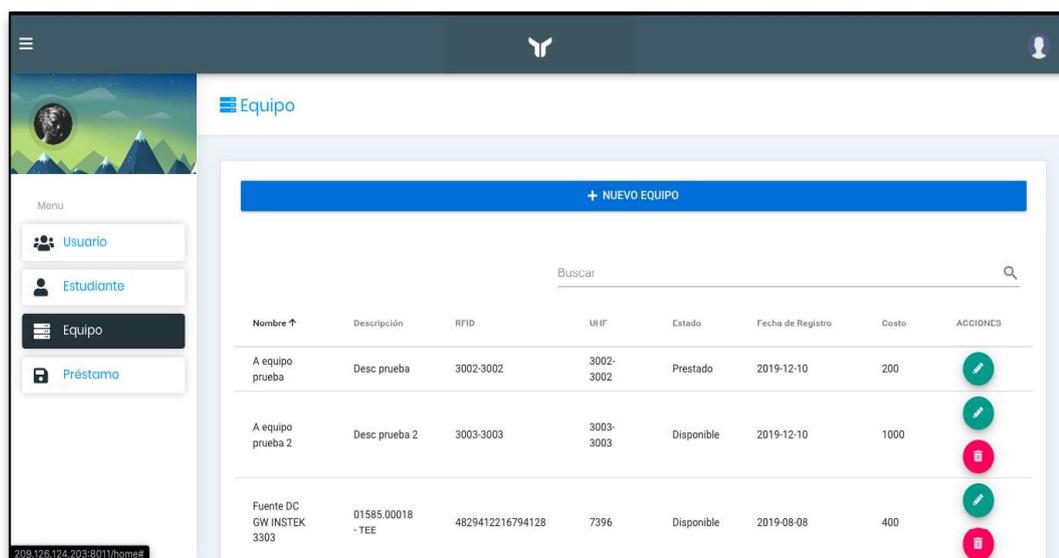


Figura 6.20 Pestaña equipo.

a) Registrar un nuevo equipo

Dar clic en la barra azul que dice nuevo equipo, seguidamente se abre una ventana (figura 6.23) para ingresar los datos del equipo, después se debe pasar el *tag* de la Figura 6.21 muy cerca del módulo RFID-RC522 de la caja del cliente 1 observando que se encienda el led caso contrario pasar nuevamente el tag y después de 5 segundos bajo este ícono  se mostrará el código del equipo a prestar. Luego pasar el tag de la figura 6.22 por la antena RFID-UHF colocada cerca de la puerta asegurándose que se emitió un leve sonido de registro, caso contrario pasar nuevamente el tag y después de 5 segundos bajo este ícono  se observará el ID del tag. Finalmente con todos los datos llenos y extraído el ID de los tags se dará clic en guardar y si todo es correcto sale un mensaje al cual solo deberá dar clic en **OK** para que desaparezca.



Figura 6.21 Tag de corta distancia.



Figura 6.22 Tag RFID-UHF.

A screenshot of a software application window titled 'Nuevo Equipo'. The window has a blue header bar with a close button (X) on the right. Below the header, there are several input fields and checkboxes. The 'Nombre' field is empty. The 'Fuente DC' field contains the text 'De plastico'. The 'Descripcion' field is empty. The 'Uhf' checkbox is checked, and the 'RFID' checkbox is unchecked. The 'Uhf' field contains the value '2102250'. The 'RFID' field contains the value '13121460217'. The 'Costo' field contains the value '200'. At the bottom right of the window, there are two buttons: 'Salir' and 'Guardar'.

Figura 6.23 Ventana nuevo equipo.

6. En la pestaña préstamo se puede observar los equipos prestados y los devueltos con sus respectivos datos de las personas que lo adquirieron y la fecha que se realiza dicha acción, a su vez estas actividades se pueden eliminar (figura 6.24). Esta pestaña cuenta con dos botones que se describen a continuación:

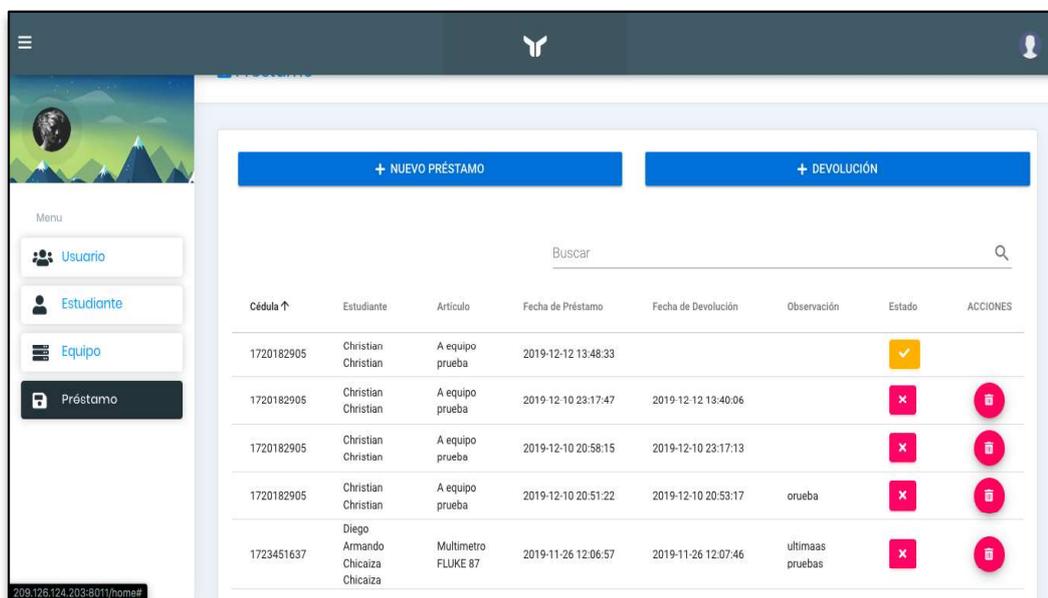


Figura 6.24 Pestaña préstamo.

a) Préstamo

Al dar clic en nuevo préstamo se desprende una ventana (figura 6.25) la cual llena sus datos automáticamente al dar clic en el ícono,  pero después de 5 segundos de haber pasado el *tag* de corta distancia de la figura 6.21 muy cerca del módulo RFID-RC522, percatarse que el led se encendió indicando que la información fue leída y sino intentarlo nuevamente. Seguidamente se debe colocar el dedo pulgar de la persona que va a realizar el préstamo en el lector de huella digital de igual manera verificar que el led se encienda o intentarlo de nuevo y después de 5 segundos dar  clic en el ícono para que se llene la información en la ventana del préstamo y finalmente al dar clic en guardar se enviará automáticamente una notificación de correo electrónico indicando el préstamo, la fecha, el equipo y la persona que realizó esta actividad.

Nota: Si al colocar la huella dactilar el led parpadea, esto quiere decir que el usuario no se encuentra registrado y que para poder realizar algún préstamo deberá realizar el proceso de registrar un nuevo estudiante.

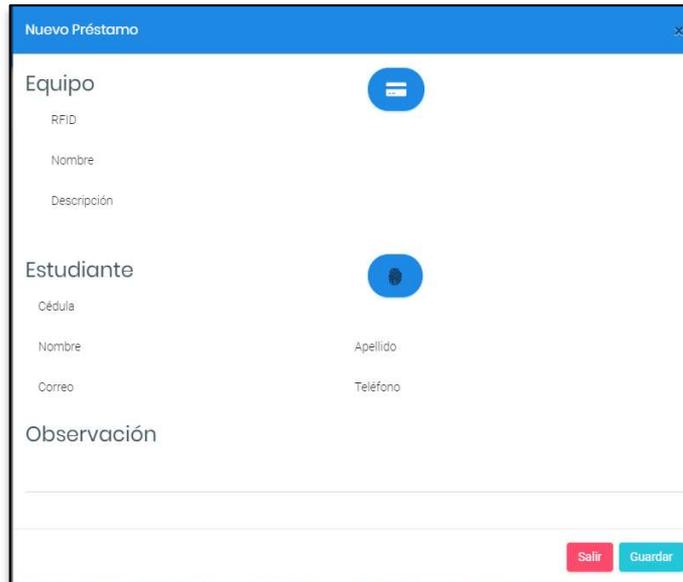


Figura 6.25 Ventana nuevo préstamo.

b) Devolución

Al dar clic en devolución se desprende una ventana (figura 6.26) la cual llena sus datos automáticamente al dar clic en el ícono,  pero después de 5 segundos de haber pasado el tag de corta distancia de la figura 6.21 muy cerca del módulo RFID-RC522, percatarse que el led se encendió indicando que la información fue leída y sino intentarlo nuevamente. Finalmente, al dar clic en guardar se enviará automáticamente una notificación de correo electrónico indicando la devolución, la fecha, el equipo y el nombre de la persona que inicialmente realizó el préstamo.

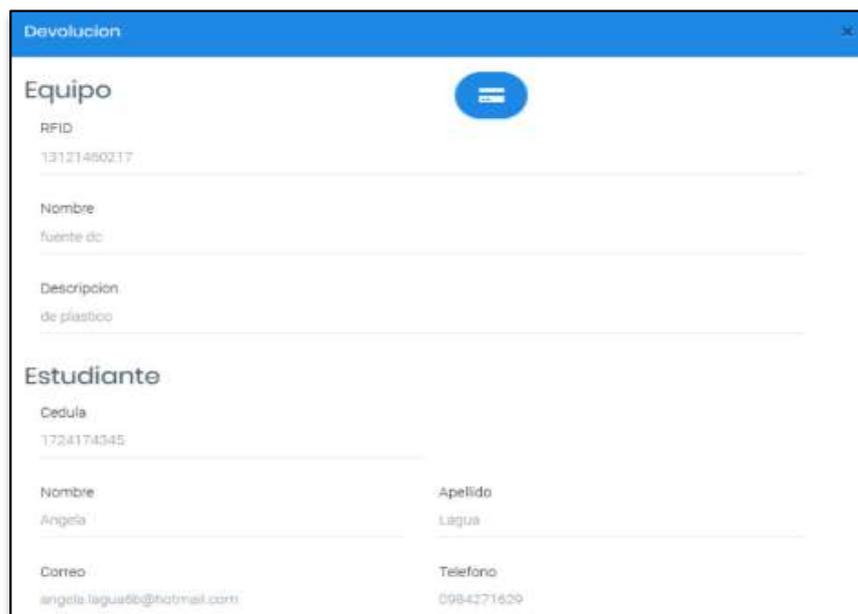


Figura 6.26 Ventana de devolución.

7. Simbología general

-  Registrar una nueva huella dactilar.
-  Extraer ID de RFID de corta distancia.
-  Extraer ID de RFID de larga distancia UHF.
-  Extrae datos de la persona que colocó la huella dactilar.
-  Equipo prestado.
-  Equipo devuelto.
-  Eliminar.

Rows per page: Número de filas por página.

 Filtrar búsqueda.

 Visualizar usuario, permite cerrar sesión.