



# UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y  
LOS RECURSOS NATURALES  
NO RENOVABLES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y  
TELECOMUNICACIONES

**“DESARROLLO DE UN PROTOTIPO DE REGISTRO PARA EL  
PRÉSTAMO DE COMPUTADORAS Y ESPACIO EN LA BIBLIOTECA  
DEL AEIRNNR DE LA UNL, CON ARDUINO Y UN LECTOR RFID”**

TESIS DE GRADO PREVIO A OPTAR POR  
EL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y TELECOMUNICACIONES

**AUTOR:**

Andreina Verónica Chunchu Román

**DIRECTOR:**

Ing. Andy Fabricio Vega León, Mg. Sc.

LOJA-ECUADOR

2016



## **CERTIFICACIÓN**

Ing. Andy Fabricio Vega León, Mg. Sc.

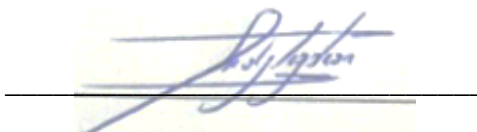
**DIRECTOR DEL TRABAJO DE TESIS**

### **CERTIFICA:**

Que el presente proyecto fin de carrera elaborado previo a la obtención del Título de Ingeniería en Electrónica y Telecomunicaciones, titulado: **“DESARROLLO DE UN PROTOTIPO DE REGISTRO PARA EL PRÉSTAMO DE COMPUTADORAS Y ESPACIO EN LA BIBLIOTECA DEL AEIRNNR DE LA UNL, CON ARDUINO Y UN LECTOR RFID.”**, realizado por la estudiante Egresada **Andreina Verónica Chuncho Román**, cumple con los requisitos establecidos por las normas generales para la graduación en la Universidad Nacional de Loja, tanto en aspecto de forma como de contenido.

Por lo tanto, autorizo proseguir los trámites legales para su presentación y defensa.

Loja, 10 de noviembre del 2016

A handwritten signature in blue ink, appearing to read 'Andy Fabricio Vega León', is written over a horizontal line.

Ing. Andy Fabricio Vega León, Mg. Sc

**DIRECTOR DEL TRABAJO DE TESIS.**

## **AUTORÍA**

Yo, **ANDREÍNA VERÓNICA CHUNCHO ROMÁN**, declaro ser autora del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional- Biblioteca Virtual

**Firma:** .....

A handwritten signature in blue ink, reading "ANDREINA CHUNCHO ROMAN", is written over a horizontal dotted line.

**Cédula:** 1105681710

**Fecha:** 12 de diciembre de 2016


**CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.**

Yo, **ANDREÍNA VERÓNICA CHUNCHO ROMÁN**, declaro ser autora de la tesis titulada: **“DESARROLLO DE UN PROTOTIPO DE REGISTRO PARA EL PRÉSTAMO DE COMPUTADORAS Y ESPACIO EN LA BIBLIOTECA DEL AEIRNNR DE LA UNL, CON ARDUINO Y UN LECTOR RFID”** como requisito para optar por grado de: **INGENIERA EN ELECTRÓNICA Y TELECOMUNICACIONES**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los doce días del mes de diciembre del dos mil dieciséis.

**Firma:**  .....

**Autor:** Andreína Verónica Chunchu Roman.

**Cédula:** 1105681710

**Dirección:** Catamayo (Av. Catamayo y Alonso de Mercadillo).

**Correo Electrónico:** anvercita.ac@gmail.com

**Teléfono:** 2677689 **Celular:** 0999755406.

**DATOS COMPLEMENTARIOS**

**Director de Tesis:** Ing. Andy Fabricio Vega León, Mg. Sc.

**Tribunal de Grado:** Ing. Juan Manuel Galindo, Mg. Sc.

Ing. Luis Enrique Chuquimarca Jiménez, M.I.

Ing. Diego Fernando Carrera Moreno Mg. Sc.

## DEDICATORIA

*Al supremo hacedor, único Dios, principio y fin, el que me ha dado fortaleza y me ha restaurado tantas veces; por ello con todo el amor, y humildad que de mi corazón emana, la presente tesis está dedicada principalmente a Dios, a mis padres, a mis hermanos, a mi familia, a mis amigos; pilares fundamentales en mi desarrollo continuo.*

*A mis maestros por su esfuerzo diario, por sembrar en nosotros la chispa de la curiosidad, investigación y aquello acerca del poder que posee el conocimiento.*

## AGRADECIMIENTO

*Un corazón agradecido es un corazón lleno de amor, la gratitud engrandece por ello agradezco principalmente al ser más maravilloso que existe y existirá por todos los siglos, que ha sido mi mejor amigo, mi mayor ejemplo, mi guardián, agradezco por todo lo que me ha dado, por su compañía y su apoyo incondicional, por su eterno amor, por haber puesto todo lo necesario en mi camino para lograr el presente objetivo, por todos los días maravillosos que te hacen sentir viva.*

*A Hugo por confiar en mí en todo momento y por su ayuda incondicional; a Jordy que fué y será mi mayor ejemplo de lucha, siempre estarás en mi corazón; a mis papis; mamis; hermanos; familia; amigos, son los mejores regalos que me pudieron dar, por todo el apoyo que he recibido, por darme ánimos en todo momento y por su paciencia.*

*Agradezco a la Universidad Nacional de Loja y excelentes docentes que ayudaron con mi formación, por todas las alegrías, aprendizaje, por los maravillosos momentos que viví en aquellas aulas que no se podrán olvidar fácilmente y que ayudarán en mi vida profesional, especialmente a mi director de Tesis, por la ayuda brindada para la finalización del mismo.*

## ÍNDICE

CERTIFICACIÓN.....	II
AUTORÍA .....	III
CARTA DE AUTORIZACIÓN .....	IV
DEDICATORIA.....	V
AGRADECIMIENTO .....	VI
ÍNDICE.....	VII
ÍNDICE DE TABLAS.....	X
ÍNDICE DE FIGURAS .....	XI
1. TÍTULO .....	1
2. RESUMEN.....	2
2.1 Abstract.....	3
3. INTRODUCCIÓN .....	4
4. REVISIÓN DE LITERATURA.....	6
4.1 CAPÍTULO I: SISTEMAS DE IDENTIFICACIÓN AUTOMÁTICA .....	6
4.1.1 Sistemas de autoidentificación. ....	6
4.1.1.1 Sistema de código de barras. ....	6
4.1.1.2 Sistemas biométricos. ....	6
4.1.1.3 Sistemas con bandas magnéticas. ....	7
4.1.1.4 Sistemas de acceso con tarjetas RFID .....	7
4.1.1.5 Comparación entre sistemas de acceso.....	7
4.1.2 Identificación por radiofrecuencia.....	9
4.1.2.1 Elementos de un sistema RFID. ....	9
4.1.2.2 Frecuencias, regulación y normativa .....	13
4.1.2.3 Frecuencia de operación 13,56 MHz.....	16
4.1.2.4 Seguridad.....	18
4.1.2.5 Control de errores .....	19
4.1.2.6 Ventajas de RFID. ....	20
4.1.3 Codificación y Modulación .....	20
4.1.3.1 Codificación en banda Base. ....	20
4.1.3.2 Modulación.....	21
4.2 CAPÍTULO II: ALMACENAMIENTO DE INFORMACIÓN .....	21
4.2.1 Servidor de almacenamiento de información. ....	22

4.2.1.1 Tipos de servidores de alojamiento .....	22
4.2.1.1.1 Servidor web.....	22
4.2.1.1.2 Servidor de base de datos .....	23
4.2.1.1.2.1 Principales modelos de los servidores de base de datos .....	23
4.2.2 Sistema de Gestores de Bases de datos (SGBD) comerciales y libres. ....	26
4.2.3 Análisis de uso de las bases de datos.....	27
4.2.4 Ventajas de los servidores de los sistemas de gestión de base de datos (SGBD)..	29
4.2.5 MySQL .....	30
4.2.5.1 PHPMYADMIN.....	31
4.2.6 PHP .....	31
4.3 CAPÍTULO III: PORTALES CAUTIVOS .....	33
4.3.1 Portales cautivos con hotspot .....	33
4.3.2 Tipos de portales cautivos .....	35
4.3.2.1 Portales cautivos por hardware.....	35
4.3.2.2 Portales cautivos por software .....	36
4.3.2.3 Similitudes y diferencias entre portales cautivo por software y por hardware...	37
4.3.3 MikroTik.....	38
5. MATERIALES Y MÉTODOS .....	40
5.1 Descripción de los sistemas de autoidentificación. ....	40
5.1.1 Instalación de AppServ.....	40
5.1.2 Sistema SALA 1: Activación y desactivación de las computadoras de mesa, mediante el registro automático.....	47
5.1.3 Sistema SALA 2: Acceso a INTERNET mediante autenticación usando Hotspot, después del registro automático.....	54
5.2 Diseño del software. ....	62
5.2.1 Sala I, desbloqueo de máquinas de mesa después del registro.....	62
5.2.2 Sala II, préstamo de espacio para utilizar wifi en computadoras personales o PDAs.....	67
5.2.2.1 Configuración de hotspot en Mikrotik.....	69
5.2.3 Descripción del Software Utilizado.....	73
5.2.3.1 Winbox .....	73
5.2.3.2 Arduino .....	74
5.2.3.3 Netbeans .....	74



5.2.3.4 Sublime Text .....	75
5.3 Diseño del hardware .....	76
5.3.1 Diseño del lector RFID para Sala 1 .....	76
5.3.2 Diseño del lector RFID para Sala 2 .....	76
5.4 Diagramas de flujo.....	77
5.4.1 Diagrama Sala 1, software Arduino. ....	77
5.4.2 Diagrama Sala 2, software Arduino .....	79
6. RESULTADOS.....	81
6.1 Prototipo Sala 1 .....	81
6.1.1 Tabla de registro “Biblioteca” .....	81
6.1.2 Interfaces de bloqueo y desbloqueo.....	81
6.2 Prototipo Sala 2 .....	83
6.2.1 Tabla de registro “registro_wifi” .....	83
6.2.2 Pantalla de autenticación.....	84
6.3 Costos .....	85
7. DISCUSIÓN.....	86
8. CONCLUSIONES .....	89
9. RECOMENDACIONES .....	90
10. BIBLIOGRAFÍA .....	92
11. ANEXOS .....	95

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Comparación entre Tecnologías de control de Acceso (Sánchez, 2008) .....	8
<b>Tabla 2.</b> Características de las cuatro bandas de frecuencia en que puede trabajar un sistema RFID (Hernández, 2009) .....	14
<b>Tabla 3.</b> Tabla de estudiantes, con información de nombre de estudiante y número de cédula.....	25
<b>Tabla 4.</b> SGBD gratuitos. ....	26
<b>Tabla 5.</b> SGBD de paga. ....	27
<b>Tabla 6.</b> Diferencia de portales cautivos (Tapia, 2012).....	38
<b>Tabla 7.</b> Opciones de lectura de la tag RFID NFC V3 (Elechouse).....	50
<b>Tabla 8.</b> Características Técnicas Arduino uno (2016). ....	54
<b>Tabla 9.</b> Especificaciones técnicas del dispositivo RB951Ui-2HND (2016).....	61
<b>Tabla 10.</b> Conexión del módulo RFID V3 con la placa Arduino Uno, sala 1 .....	76
<b>Tabla 11.</b> Conexión del módulo RFID V1 con la placa Arduino uno, sala 2 .....	77
<b>Tabla 12.</b> Precio de los materiales usados. ....	85

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Estructura código de barras (Izard, 2005).....	6
<b>Figura 2.</b> Sistema biométrico, detección de huellas dactilares (Pérez, 2016). ....	7
<b>Figura 3.</b> Sistemas con bandas magnéticas (Spetel, 2015).....	7
<b>Figura 4.</b> Elementos de un sistema RFID.....	10
<b>Figura 5.</b> Antena con elemento inductivo (izquierda) y antena dipolar (derecha) (Casero, 2013) .....	11
<b>Figura 6.</b> Lector RFID móvil (Casero, 2013).....	12
<b>Figura 7.</b> Dependencia de la intensidad de campo con respecto a la distancia (Javier Portillo, 2008).....	17
<b>Figura 8.</b> Estructura cliente-servidor. ....	22
<b>Figura 9.</b> Portal cautivo, dispositivo Endian (Endian). ....	36
<b>Figura 10.</b> Portal cautivo en Antamedia (Antamedia).....	37
<b>Figura 11.</b> Instalación de App Server. ....	40
<b>Figura 12.</b> Creando una base de datos. ....	42
<b>Figura 13.</b> Campos requeridos para la tabla de registro de la Sala 1.....	42
<b>Figura 14.</b> Modificación de los campos de la tabla “registro”. ....	43
<b>Figura 15.</b> Tabla que incluya la lectura de las tarjetas RFID. ....	44
<b>Figura 16.</b> Tabla que incluya la lectura de las tarjetas RFID de la Sala 1.....	44
<b>Figura 17.</b> Campos requeridos para la tabla de registro de la Sala 2.....	44
<b>Figura 18.</b> Modificación de los campos de la tabla “registro_wifi”.....	44
<b>Figura 19.</b> Tabla que incluya la lectura de las tarjetas RFID de la Sala 2.....	45
<b>Figura 20.</b> Campos de la tabla “tarjetas” .....	45
<b>Figura 21.</b> Campos de la tabla “tarjetas” .....	46
<b>Figura 22.</b> Página oficial del servidor web.....	46
<b>Figura 23.</b> Creación de BD en el servidor web. ....	47
<b>Figura 24.</b> Sistema Sala 1. ....	48
<b>Figura 25.</b> Tag RFID 13,56 MHz (software, 2016).....	50
<b>Figura 26.</b> Placa RFID Elechouse v3 (Elechouse). ....	51
<b>Figura 27.</b> Placa Arduino Uno (2016). ....	52
<b>Figura 28.</b> Sistema Sala 2, internet inalámbrico.....	55
<b>Figura 29.</b> Módulo RFID, Elechouse V1.1 (Elechouse, 2016). ....	57
<b>Figura 30.</b> Shield Ethernet de Arduino (2016). ....	59
<b>Figura 31.</b> Mikrotik RB951Ui-2HnD (Routerboard). ....	60
<b>Figura 32.</b> Interfaz gráfica JFrameGui. ....	63
<b>Figura 33.</b> Interfaz gráfica para el registro, usando el número de cédula. ....	64
<b>Figura 34.</b> Interfaz gráfica de la pantalla de bloqueo. ....	65
<b>Figura 35.</b> Interfaz gráfica al desbloquear la pantalla. ....	66
<b>Figura 36.</b> Conexiones WAN, LAN Y WLAN al routerboard.....	69
<b>Figura 37.</b> Lista de direcciones IP, para las interfaces de red. ....	70

<b>Figura 38.</b> Enmascaramiento de las IP's parte 1. ....	70
<b>Figura 39.</b> Enmascaramiento de las IP's parte 2. ....	71
<b>Figura 40.</b> Lista de rutas y su puerta de enlace. ....	71
<b>Figura 41.</b> Creación del hotspot en Mikrotik.....	72
<b>Figura 42.</b> Usuarios creados para Hotspot.....	72
<b>Figura 43 .</b> Interfaz de autenticación del Hotspot. ....	73
<b>Figura 44.</b> Winbox.....	74
<b>Figura 45.</b> Arduino .....	74
<b>Figura 46.</b> Netbeans.....	75
<b>Figura 47.</b> Sublime text. ....	75
<b>Figura 48.</b> Arduino Uno más Shield Ethernet (2016).....	76
<b>Figura 49.</b> Diagrama de flujo del programa en Arduino, para la lectura serial sala 1... 77	77
<b>Figura 50.</b> Diagrama general de flujo de la sala 1.....	78
<b>Figura 51.</b> Diagrama de flujo del programa en Arduino, para la lectura serial sala 2... 79	79
<b>Figura 52.</b> Diagrama general de flujo de la sala 2.....	80
<b>Figura 53.</b> Registro de los usuarios para el uso de las computadoras de escritorio. ....	81
<b>Figura 54.</b> Computadoras bloqueadas, sala 1. ....	81
<b>Figura 55.</b> Máquina servidor principal. ....	82
<b>Figura 56.</b> Máquina desbloqueada, después del registro. ....	82
<b>Figura 57.</b> Máquina principal al pasar el identificador RFID por el lector. ....	83
<b>Figura 58.</b> Registro de los usuarios para el uso de la Sala 2. ....	83
<b>Figura 59.</b> Usuarios activos Hotspot. ....	84
<b>Figura 60.</b> Página de autenticación Hotspot, sala 2. ....	84

## **1. TÍTULO**

**“Desarrollo de un prototipo de registro para el préstamo de computadoras y espacio en la biblioteca del AEIRNNR de la UNL, con Arduino y un lector RFID”**

## **2. RESUMEN**

En este trabajo de tesis se presenta un prototipo de registro automático usando tecnología RFID, solución al registro manual de usuarios que actualmente se usa en la biblioteca del AEIRNNR, se usa específicamente placas de desarrollo Arduino, Arduino Ethernet Shield, Módulos RFID elechouse compatibles con Arduino, tarjetas RFID a 13,56MHz y un equipo router Wireless de MikroTik.

La biblioteca se compone de la sala 1 correspondiente al préstamo de computadoras de escritorio y la sala 2 al préstamo del espacio para usar el internet inalámbrico en computadoras personales, tablets y Smartphones.

Al diseñar los sistemas, los lectores interactúan con un router MikroTik y con java, para lograr la activación de usuarios Hotspot, y el desbloqueo de máquinas de mesa respectivamente y en los dos casos la creación de una tabla en MySQL, para almacenar el registro de los usuarios.

Se indicará el diseño del prototipo de registro automático, cuyo funcionamiento será probado a menor escala (10 usuarios), con la finalidad de mostrar la utilidad del sistema para que en un futuro pueda ser implementado en la biblioteca correspondiente a una de las áreas de la Universidad Nacional de Loja que debe estar a la vanguardia con la tecnología e innovación.

## 2.1 Abstract

In the work of thesis is presented a prototype of automatic registered using technology RFID, the solution is registered as manual of users that is used manually at library of AEIRNNR, It uses specifically Arduino development plates, Arduino Ethernet Shield, RFID *Modules*, elechouse compatibles with Arduino, RFIS targets to 13, 56 MHz and MiKroTik Wireless equipment.

The library consists of room first corresponding to the loan of desktop computers and room second to the loan of the space to use the wireless internet in personal computers, tablets and Smartphones.

Designing readers systems that interact with a MikroTik router and java, to achieve the activation of users Hotspot, and the unblocking of table machines respectively and in both cases the creation of a table in MySQL, to store the registry of the Users. The design of the prototype of automatic registration will be indicated, whose operation will be tested in smaller scale (10 users), in order to show the utility of the system so that in the future it can be implemented in the library corresponding to one of the areas of the National University of Loja, that must be at the forefront with technology and innovation.

### 3. INTRODUCCIÓN

La biblioteca del Área de Energía, las Industrias y los Recursos Naturales no Renovables (AEIRNNR) de la Universidad Nacional de Loja (UNL) que es encargada de brindar su servicio a las carreras de Electrónica y Telecomunicaciones, Electromecánica, Sistemas, Geología Ambiental y Ordenamiento Territorial, y a otras carreras que deseen acceder a los servicios de biblioteca ofrecidos; estando las carreras del AEIRNNR en su totalidad vinculadas a promover la investigación científica y la creación de nuevas tecnologías para la solución de múltiples problemas, y siendo que el sistema bibliotecario utiliza un sistema de registro totalmente manual, invirtiendo tiempo, dinero, y recursos naturales que pueden ser aprovechados de una forma diferente.

Por otra parte, los usuarios que no ingresan a la biblioteca, sin llevar el proceso de registro que se encuentran cercanos a la misma, tienen libre acceso de usar el internet inalámbrico brindado, produciendo una disminución en velocidad de navegación de los estudiantes que si se registran, se pretende solucionar dicho problema utilizando un lector Identificación por Radio Frecuencia y un portal cautivo con *Hotspot*, para que cualquier usuario que se encuentre registrado y desee ingresar a internet sea redireccionado a la página de autenticación del portal cautivo, el cual comprobará que el nombre de usuario y contraseña estén correctos, es decir que existan en la base de datos del registro con etiquetas RFID.

La idea del presente proyecto consiste en realizar un prototipo de registro automático, cuyo funcionamiento será probado a menor escala (10 usuarios), con la finalidad de mostrar la utilidad del sistema para que en un futuro pueda ser implementado.

Para la sala 1 que corresponde al préstamo de las computadoras que poseen internet cableado, la situación es casi similar, se diseña un lector RFID para obtener los datos de los usuarios que requieran el préstamo de las mismas, dado que los datos del préstamo de computadoras de mesa no pueden estar unidos a los de préstamo del espacio de la sala 2, es decir se necesita dos sistemas separados, los únicos requerimientos adicionales de esta sala son dos, el primero es que se necesita indicar la disponibilidad de las máquinas, el segundo requerimiento es que se debe controlar el tiempo de uso de las



computadoras de mesa, pues el reglamento interno de la biblioteca especifica que solo se las puede utilizar durante un tiempo determinado.

Para llevar a cabo el presente diseño del prototipo se requiere de placas Arduino, Shield Arduino Ethernet, router Mikrotik, lectores RFID a 13,56MHz y algunas etiquetas de radiofrecuencia, estos materiales no representan una gran inversión y son de fácil adquisición.

De igual manera el software utilizado es libre, se desarrollará el mecanismo de almacenamiento en MySQL mediante PHP y Java, por tanto, no se requiere inversión económica, por el uso de licencias.

Se prevé que el sistema ayudará a que los usuarios siempre lleven el registro y a los bibliotecarios a tener toda la información sin desechar ningún dato, para que más usuarios accedan a la biblioteca, fomentando el uso de espacios dedicados para realizar tareas y consultas de manera eficiente, y a la vez se convertirá en la primera biblioteca de la UNL en comenzar a implementar un registro de este tipo.

La información que se encuentra en la etiqueta (*tag*) será interceptada por el lector cada vez que se encuentre cerca del mismo, una vez obtenida dicha información se almacena en una base de datos local; al tener el registro diario, al bibliotecario le va a tomar unos minutos importarlo y subirlo a un servidor web, al realizar el almacenamiento web se asegura la información, es decir si ocurre algún problema en el servidor la información no va a ser afectada, este respaldo de información se encontrará disponible en la web para cierto personal autorizado.

Para el desarrollo del prototipo se plantearon los siguientes objetivos: diseñar un sistema RFID utilizando Arduino Uno, módulo RFID; integrar el sistema a una base de datos libre MySQL y *PHP*; desarrollar un portal cautivo con *hotspot* para limitar la señal de internet inalámbrico solamente para las personas que llevan a cabo el registro.

Según la estructura del proyecto, en el inciso (4) se indicará el marco de referencia de los sistemas de identificación, el almacenamiento de información y portales cautivos; el inciso (5) trata de materiales y métodos donde se describirá el software y hardware del prototipo, en el inciso (6) se muestran los resultados, en el (7) la discusión, las

conclusiones, recomendaciones, bibliografía y anexos, se indican en los incisos 8, 9, 10, 11 correspondientemente.

## **4. REVISIÓN DE LITERATURA**

### **4.1 CAPÍTULO I: SISTEMAS DE IDENTIFICACIÓN AUTOMÁTICA**

#### **4.1.1 Sistemas de autoidentificación.**

Los sistemas autoidentificación o identificación automática van evolucionando y mejorando con el pasar de los años, se emplean para el manejo de información que corresponde a personas y objetos, logrando procesos ágiles, un margen de error menor, una pérdida menor de tiempo para los usuarios en comparación con los procesos de registro manuales [1].

Entre los múltiples sistemas autoidentificativos, daremos a conocer los más comunes:

##### **4.1.1.1 Sistema de código de barras.**

Sistema muy popular, usado en almacenes, empresas, centros distributivos, etc., por su bajo precio, que permite la lectura mediante un procedimiento óptico, necesitando línea de vista, la figura 1 nos indica la estructura de un código de barras [1].



**Figura 1.** Estructura código de barras [2].

##### **4.1.1.2 Sistemas biométricos.**

Implican características físicas de los usuarios, por ejemplo, detector de huellas (véase la Figura 2) con esto se evitaría llevar algún identificativo para el registro, es un buen sistema de control de acceso, sin embargo, es bastante costoso [3].



**Figura 2.** Sistema biométrico, detección de huellas dactilares [4].

#### **4.1.1.3 Sistemas con bandas magnéticas.**

Estos sistemas usan señales electromagnéticas para guardar información en la banda magnética, no es fácil su falsificación, presentan agilidad para un registro o acceso, tienen bajo costo, sin embargo, su deterioro es continuo de acuerdo al uso, son sistemas de contacto (véase la figura 3); y si existiese una fuente electromagnética fuerte podría modificar la información que se encuentra en la tarjeta [3].



**Figura 3.** Sistemas con bandas magnéticas [5]

#### **4.1.1.4 Sistemas de acceso con tarjetas RFID**

En este sistema de acceso intervienen un lector, una etiqueta o tarjeta de Identificación por radiofrecuencia (RFID, por sus siglas en inglés) y un sistema de procesamiento que almacena los datos, los datos se transmiten por radiofrecuencia entre lector y etiqueta, característica que ofrece menor desgaste, y por utilizar ondas de radio no se requiere línea de vista, ni contacto físico [3].

Las tarjetas RFID ofrecen la posibilidad de lectura y escritura, convirtiéndose en un sistema con múltiples aplicaciones [3].

#### **4.1.1.5 Comparación entre sistemas de acceso**

Para escoger una tecnología de acceso es necesario una previa comparación, no se pretende recriminar un sistema con respecto a otro, más bien su elección está íntimamente ligada con la aplicación, tomando en cuenta factores como durabilidad, rapidez, seguridad, capacidad, etc. En la Tabla 1 se muestra una comparación de las características de los sistemas de acceso [3].

**Tabla 1.** Comparación entre Tecnologías de control de Acceso [3]

<b>Características</b>	<b>Sistemas biométricos.</b>	<b>Sistemas con bandas magnéticas</b>	<b>Sistema de código de barras.</b>	<b>Sistema <i>RFID</i></b>
PRECIO	Alto	Medio-Bajo	Bajo	Medio
CICLO DE VIDA	Alto	Medio	Corto	Indefinido
DISTANCIA DE LECTURA	Depende del sistema biométrico	Requiere contacto	Línea de vista hasta 1.5m	Sin contacto ni línea de vista, hasta 10m
INTERFERENCIA POTENCIAL	Bloqueo del contacto o línea de vista, les afecta el ruido.	Bloqueo del contacto	Presenta problemas si son modificadas las barras de las etiquetas o tarjetas.	Campos que afecten a la transmisión de las ondas de radio
CAPACIDAD DE ALMACENAMIENTO DE DATOS	No aplica	Hasta 128 bytes	8 – 7200 caracteres.	Hasta 64KB
SEGURIDAD DE DATOS	Alta	Media	Mínima	De baja a alta
MODIFICACIÓN DE LA INFORMACIÓN	No modificable	Modificable	No modificable	Modificable

**Fuente:** Elaboración propia.

De acuerdo a la Tabla 1 se puede observar que cada tecnología presenta tanto ventajas como desventajas, para un sistema de registro automático en la biblioteca se escogió la tecnología RFID, la misma que no necesita línea de vista ni contacto físico para realizar un control de acceso como lo requieren el sistema de código de barras y el magnético, cada usuario va a tener un código único, es una tecnología de lectura y escritura, se pueden hacer lecturas simultáneas, su alcance varía desde un par de centímetros hasta varios metros dependiendo de la frecuencia de los equipos, presentan mayor durabilidad y menor desgaste porque no tienen fricción y son reescribibles [3].

#### **4.1.2 Identificación por radiofrecuencia**

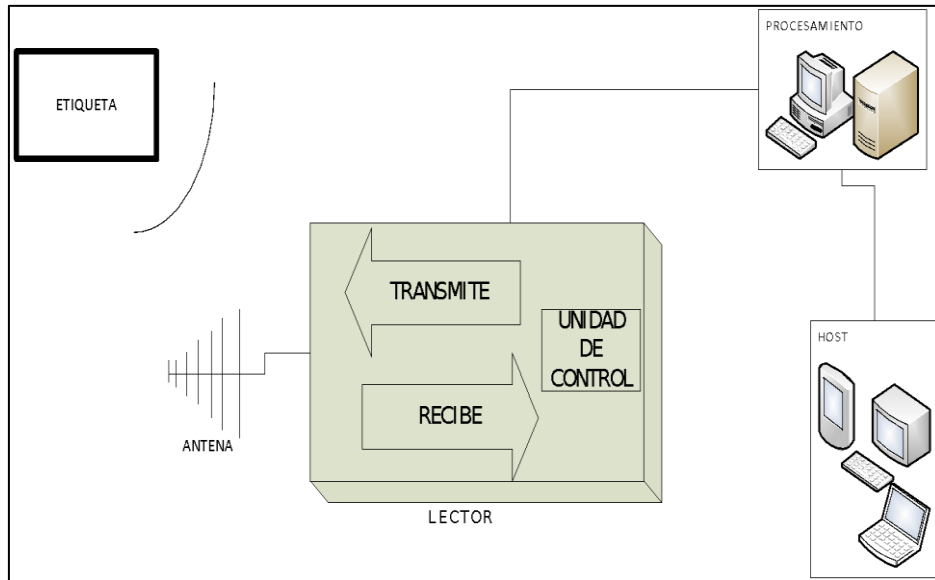
Como se mencionó en los puntos anteriores una de las formas de lograr una identificación automática es mediante el uso de tecnología de identificación por radio frecuencia (RFID, por sus siglas en inglés Radio Frequency Identification) [6].

Con sistemas RFID se puede realizar identificaciones relativas de personas, animales, objetos y capturar datos [6].

Las tarjetas, etiquetas y llaveros RFID tienen incluido un autoindentificador (ID), que es un código único, tanto la tarjeta como el lector RFID deben trabajar en la misma frecuencia para lograr la captura o escritura de datos en la tarjeta, empleando ondas de radio para la captura e identificación en forma automática [6].

##### **4.1.2.1 Elementos de un sistema *RFID*.**

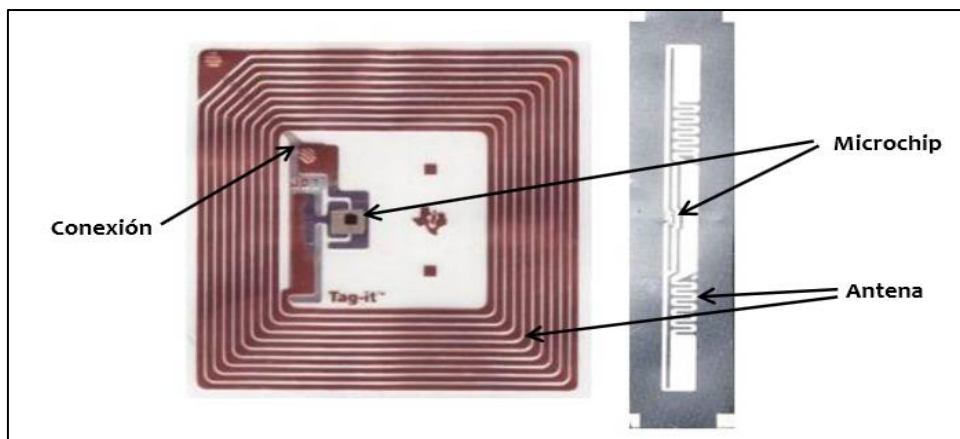
Un sistema RFID está formado por los siguientes elementos:



**Figura 4.** Elementos de un sistema *RFID*.

**Fuente:** Elaboración propia.

1. **Etiquetas (*tags*):** O transpondedores (transmisión/recepción solo cuando el lector se lo pida), aquí se almacena la información, las *tags* en su interior contienen un pequeño *chip* de silicio (receptor y transmisor) y una antena [7]. La antena transmite la información que contiene el *microchip* cuando detecte las ondas de radiofrecuencia que emite el lector, las antenas pueden ser de dos tipos, un elemento inductivo (bobina) y dipolo, así como se indica la figura 2 [7, 6]. La antena de la etiqueta y su carga electromagnética producida es menor a la quinta parte que producen los teléfonos móviles, por tanto, no son perjudiciales para la salud [7].



**Figura 5.** Antena con elemento inductivo (izquierda) y antena dipolar (derecha) [6]

Según el modo de alimentación las etiquetas pueden ser activas, pasivas y semi-activas, en las activas la fuente de alimentación viene incluida en la tarjeta, su alcance es mayor y puede almacenar mayor cantidad de datos, son más costosas; en cambio las pasivas no necesitan una batería siendo más pequeñas y económicas sin embargo para que funcionen deben estar dentro del campo que genera el lector y su rango de lectura es limitado; el costo de las semi-activas está entre las pasivas y las activas, estas poseen una fuente de alimentación que solamente se utiliza para mantenerse activada y no para transmitir datos, por lo que las hace útiles para objetos en movimiento y que su rango de lectura sea mayor que una pasiva [6, 3].

Los niveles de energía que las tarjetas necesitan para activarse y enviar información está en el orden de los micro a milivatios [6, 8].

Así mismo se clasifican dependiendo de la capacidad de almacenamiento, las tarjetas con mayor capacidad de almacenamiento son más costosas y la velocidad de lectura para una alta capacidad de almacenamiento llevará más tiempo [6].

El código EPC (EPC, por sus siglas en inglés Electronic Product Code) de las etiquetas varía de 64 a 96 *bits* y es un identificativo único de la tarjeta que viene encriptado en el chip de silicio [7].

Las tarjetas RFID tienen las medidas estandarizadas de 85.72 mm x 54.03 mm x 0.76 mm  $\pm$  tolerancias [8].

Los patrones de radiación de una tarjeta RFID pueden ser: isotrópico, omnidireccional y directivo [8].

La información que se envía desde la tarjeta al lector puede ser *half-dúplex*, *full-dúplex* y secuencial, la secuencial es cuando el campo del lector no permanece siempre prendido, este se enciende a una secuencia cuadrática, por tanto, necesita una alimentación externa del lector [8].

2. **Lectores:** O también conocido como transceptor (transmisor-receptor), le envía pulsos de energía a la tarjeta que responde con información que en ella se encuentra, luego el lector envía los datos a un computador para su procesamiento [6, 7].

En los lectores se realiza la modulación y demodulación de la información [6].

Existen lectores fijos y móviles, los fijos tienen mayor cobertura que los móviles; y también con bobina simple que tienen un corto alcance, esto se debe a que la bobina transmite tanto energía como datos, y los lectores con bobina doble que hacen la misma tarea que los de bobina simple, pero de forma más eficaz, presentando mayor alcance [6].



**Figura 6.** Lector RFID móvil [6].

3. **Hardware y procesamiento:** Son ordenadores que reciben la información de uno o varios lectores, el cual convierte las ondas radiales emitidas por la etiqueta RFID en información digital, para luego realizar alguna otra tarea como por ejemplo encender un foco, activar una computadora, abrir una puerta, etc., en este computador generalmente corre una base de datos [3].

La comunicación de un sistema RFID es bidireccional, asíncrona y transmiten esa información basándose en sistemas de acoplamiento, estos sistemas pueden ser [9]:

- Acoplamiento electromagnético: Usado por sistemas RFID a Ultra Alta Frecuencia (UHF, por sus siglas en inglés Ultra High Frequency), refleja y envía la señal al origen a largo alcance, las antenas son pequeñas [9].
- Acoplamiento inductivo: Usado por sistemas RFID a alta y baja frecuencia, la energía que necesitan las etiquetas para entrar en funcionamiento es dada por el lector mediante inducción, el lector crea un campo magnético, en este campo pueden entrar en funcionamiento las etiquetas y entra en funcionamiento emitiendo información, para ampliar la distancia de este campo se debe elevar la potencia de la antena del lector [9].



La transferencia de datos en un sistema inductivo puede ser de tres formas:

1. *Load modulation*: Representa a la activación de la tarjeta mediante el campo magnético que produce el lector, como una impedancia, esta impedancia debe ser variada al conectarla y desconectarla, esto es controlado por los datos, logrando variar el voltaje y enviar datos entre tarjeta y lector, la distancia entre lector y *tag* no debe ser mayor a  $0,16\lambda$  (deben estar cerca) [12].
2. *Load modulation* con portadora: Usada cuando existe un acoplamiento débil entre *tag* y lector, y es muy complicado detectar las variaciones de tensión. Para realizar este acoplamiento se debe colocar una resistencia de carga como un circuito de acople para la *tag*, esta resistencia variará a una frecuencia elevada, estas frecuencias que resultan de la variación se las denomina subportadora, hacen más fácil detectar la variación de tensión, luego se puede utilizar la modulación por desplazamiento en amplitud (ASK), por desplazamiento en frecuencia (FSK) o desplazamiento en fase (PSK) [10].
3. Subarmónicos: Se utiliza el primer subarmónico de una frecuencia ( $f_a$ ), que es la mitad de la frecuencia del lector, luego esta frecuencia será modulada y enviada al transponder,  $f_a$  suele ser de 128 kHz y el primer subarmónico es  $f_a / 2$  [12].

El prototipo RFID a 13, 56 MHz, ocupa acoplamiento inductivo, transfiriendo sus datos mediante Subarmónicos [12].

#### **4.1.2.2 Frecuencias, regulación y normativa**

Para que se establezca una comunicación entre transpondedor y transceptor se requiere que ambos tengan la misma frecuencia [7].

Un sistema RFID puede trabajar en baja frecuencia (LF, por sus siglas en inglés Low Frequency) funciona en los 125KHz, en alta frecuencia (HF, por sus siglas en inglés High Frequency) a los 13,56MHz, en ultra alta frecuencia (UHF, por sus siglas en inglés Ultra High Frequency) a los 860 a 960MHz y en microondas a 2,45 GHz y 5.8 GHz, a mayor frecuencia mayor velocidad de lectura de datos y mayor coste [7].

En los sistemas de baja frecuencia la distancia entre el lector y el transponder es apenas de unos cuantos centímetros es decir solamente pueden leer una tarjeta a la vez; en cambio los sistemas de alta frecuencia (HF , por sus siglas en inglés High Frequency) son más populares y cubren distancias de 1cm a 1.5m utilizando comúnmente tarjetas pasivas; para Frecuencias Ultra altas (UHF, por sus siglas en inglés Ultra High Frequency) las distancias cubiertas pueden llegar hasta los 4 metros dependiendo del fabricante y del ambiente es decir se debe evitar ambientes con agua o en donde se encuentren materiales conductores como el metal, sin embargo en esta frecuencia se pueden leer más de una etiqueta a la vez, los sistemas microondas en cuanto a la distancia de lectura pueden llegar hasta los 6 metros, estos sistemas son más resistentes a los fuertes campos electromagnéticos que son originados por los motores eléctricos razón de su uso en la gestión de vehículos [3].

En la Tabla 2 se dan a conocer algunas características de los sistemas RFID en las cuatro bandas de frecuencia [3].

**Tabla 2.** Características de las cuatro bandas de frecuencia en que puede trabajar un sistema RFID [8]

BANDA DE FRECUENCIA	CAPACIDAD DE DATOS	VELOCIDAD Y TIEMPO DE LECTURA DE DATOS	COBERTURA	ZONA DE LECTURA	COSTOS DEL LECTOR	APLICACIÓN
<b>BAJA FRECUENCIA</b> <b>125 KHz</b> <b>134 KHz</b>	PASIVAS:64bits ACTIVAS:2kbits	Bajas (200 bps – 1 kbps). Tiempo de lectura lento.	Pasivas 0.5m. Activas 2m	Buena en materiales no conductores. Mala en materiales conductores.	Bajo	-Control de acceso. -Identificación de animales. -Gestión de bienes. - Identificación de vehículos, contenedores, ganado
<b>13,56 MHz ALTA FRECUENCIA</b>	PASIVAS: 512bits hasta 8 kbits*. *En HF no se utilizan las etiquetas activas	Media 25Kbps – 100 Kbps, puede leer 40 etiquetas en un segundo.	Alrededor de 1 metro.	Buena penetración en materiales no conductores y líquidos. Mala en metálicos.	Medio	-Gestión de maletas en aeropuertos. -Servicios de alquiler. -Gestión de bibliotecas, almacén. -Inventario biblioteca. -Seguimiento de paquetes. -Control de acceso.
<b>ULTRA ALTA FRECUENCIA</b> <b>433MHz,</b>	64 bits- 8kbits	28 kbps, 100 etiquetas por segundo	Pasivas de 3-4 metros. Activas hasta	Buena penetración en materiales	Muy alto	-Trazabilidad. -Gestión de bienes, artículos.

860MHz, a 960MHz o 928			10 metros.	conductores y no conductores mala en líquidos		
<b>MICROONDAS</b> <b>2.45 y 5.8 GHz</b>	128 <i>bits</i> - 512 <i>Kbits</i>	100 Kbps – 1Mbps	Pasivas de 1- 2 metros. Activas 15 metros o más.	Buena penetración en materiales no conductores, es reflejado por las superficies conductoras, susceptible al ruido	Muy alto	-Peaje electrónico. –Aplicaciones logísticas terminales. – Localización. -Control de acceso para vehículos.

Fuente: Elaboración propia.

Como indica la tabla 2, los Sistemas RFID presentan múltiples aplicaciones en los campos comerciales, industriales, de transporte, salud, en cada una de sus frecuencias de trabajo, su uso también en el control de acceso y gestión de bibliotecas.

En cuanto a estándares en RFID las organizaciones más utilizadas en normar las tarjetas RFID son ISO/IEC y EPC (EPC, por sus siglas en inglés Electronic Product Code), aunque también existen otros más [11].

La Organización Internacional de Estándares (ISO, por sus siglas en inglés International Organization for Standardization) define estándares comerciales e industriales que se puede emplear en cualquier parte del mundo, y la Comisión Electrotécnica Internacional (IEC, por sus siglas en inglés International Electrotechnical Commission) promueve la cooperación internacional para la estandarización en los campos de la electrónica y las tecnologías, de estos dos organismos nace el estándar ISO/IEC, siendo utilizado como normativa en muchos países, establece normativas que se refiere a la interfaz aérea y a las múltiples aplicaciones de RFID, en altas frecuencias se utilizan los estándares [11, 12]:

- ISO/IEC 14443: Son sistemas sin contacto a unos pocos milímetros, existen dos versiones, 14443A y 1443B [11].
- ISO/IEC 15693 [11].
- ISO/IEC 1800: [11]. ISO/IEC 18000-3 Frecuencia 13.56MHz (*HF*), para variedad de aplicaciones, especifican la interfaz aérea (forma en que las etiquetas y los lectores se pueden comunicar) [13]

El código Electrónico de Productos (EPC) divide las etiquetas por clases, encargado de asignar identificadores únicos para las etiquetas [11, 13].

Otro protocolo que pertenece a EPC Global es el EPC Gen 2, se encarga de la comunicación entre lector y etiqueta, y que no exista interferencias cuando se encuentran dos lectores cerca [11, 13].

Las etiquetas de baja frecuencia (LF) y alta frecuencia (HF) trabajan dentro de la banda ISM (ISM, por sus siglas en inglés Industrial – Scientific - Medical) por tanto no requieren el uso de licencias. La modulación y codificación depende [11, 13].del estándar.

En el caso de Ecuador se rige a las normas del código electrónico de productos (EPC), su representante en Ecuador es GS1 [www.gs1ec.org](http://www.gs1ec.org) y el uso de la frecuencia es semejante a las normas FCC (FCC, por sus siglas en español Comisión Federal de Comunicaciones) de Estados Unidos [11, 13].

#### **4.1.2.3 Frecuencia de operación 13,56 MHz**

Como podemos observar en la Tabla 2 en alta frecuencia la mayoría de los sistemas son pasivos característica que permite una vida más larga del transponder al no estar sujetas a la duración de una batería, además estos sistemas usan acoplamiento inductivo, el mismo se ocupa también en bajas frecuencias y abarca el 80% de los sistemas RFID [11].

Este principio de operación consiste en que la corriente que circula por la antena del lector genera un campo magnético que al llegar a la antena del transponder induce en ésta una corriente con la que se alimenta [11].

La zona de operación es el campo creado por la antena del lector, por tanto, operan a distancias cortas, característica que beneficia en un menor efecto de sistemas adyacentes y ruido externo comparado con un sistema de ultra alta frecuencia (UHF, por sus siglas en inglés Ultra Hight Frequency) [11].

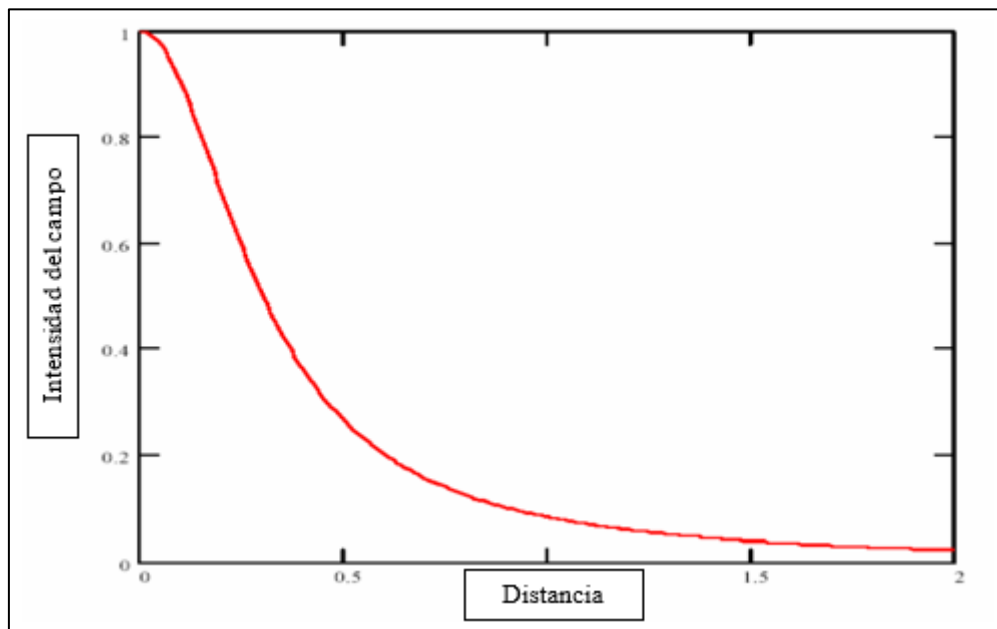
Esta frecuencia de operación se propaga fácilmente ya que no es absorbida por el agua ni por la influencia de las personas, sin embargo, es sensible a los materiales conductores como metales, y a los campos magnéticos cubre distancias de 1cm a 1.5m [11].

El campo magnético de un sistema de alta frecuencia es vectorial esto quiere decir que la orientación del transpondedor también tiene influencia dentro del sistema, la solución sería el uso de campos rotantes [11].

La tecnología Mifare trabaja en esta frecuencia y tiene incorporado un chip electrónico y antena dentro de la tarjeta, que fácilmente no son visibles, presentan ciertos parámetros que pueden ser solamente leídos y otros que pueden además de ser leídos ser cambiados, por tal razón su uso es muy común a nivel mundial por su bajo costo y rapidez [11].

El tamaño de memoria para tarjetas que funcionan en esta frecuencia va desde los 64 *bits* a varios *kilobytes*, estas pueden ser de solo lectura, solo programadas una vez y de lectura y escritura, en cuanto seguridad presenta altos niveles, útiles en el dinero electrónico, transferencias [11].

La intensidad del campo decrece con la sexta potencia de la distancia lo que quiere decir que tanto intensidad como distancia son inversamente proporcionales, como se muestra la figura 3, para un diámetro de antena de 0.8m [14].



**Figura 7.** Dependencia de la intensidad de campo con respecto a la distancia [14].

La intensidad del campo sobre la potencia de transmisor está en 42 dB $\mu$ A/m, para altas frecuencias [14].

#### 4.1.2.4 Seguridad

Es de suma importancia la seguridad en los sistemas RFID para que no sean vulnerados o atacados, creando accesos no autorizados, robo de información, fraudes [9].

Entre las formas más relevantes de violentar la seguridad de sistemas RFID, tenemos [14]:

- *Spoofing*: Ataque que consiste en enviar información falsa, que pretende ser real [14].
- *Inserción*: Ataque en el que se violenta bases de datos y se insertan otro tipo de información que le favorezca al atacante [14].
- *Replay*: Ataque en el cual se utiliza otro sistema RFID, para leer los datos del *tag*, luego estos datos son utilizados para clonar tarjetas [14].
- *Denegación de servicio (DOS, por sus siglas en inglés Denial of Service)*: Ataque que colapsa a sistemas RFID con muchos datos a tal punto que no los pueda manejar, esto puede anular la comunicación de radiofrecuencia [14].

Estos ataques se pueden evitar tomando medidas como: utilizar tarjetas de solo lectura, así se evitará la escritura no deseada, este sistema tiene que poseer alguna base de datos donde relacione el código de la tarjeta con un nombre de usuario, con su cédula de identidad, al área que pertenece, este es un ejemplo claro de un sistema de acceso, o también se pueden encriptar o codificar los datos transmitidos, la tecnología Mifare usada en el prototipo tiene protección contra ataques MIFARE [14].

La seguridad que tenga un sistema RFID es proporcional a su costo, está también depende de la aplicación que se dé al sistema, no será el mismo tipo de seguridad para aplicaciones que manejan por ejemplo transferencias de dinero, que, para aplicaciones de acceso, usualmente estos sistemas cuentan con procesos de encriptación, entre ellos los más usados son [9]:

- ✓ *Estándar de cifrado de datos (DES, por sus siglas en inglés Data Encryption Standard)*: Consiste en cifrar un texto plano en 16 pasos, se utiliza un bloque de 64 *bits*, 8 de paridad y 56 para cifrar y descifrar, esta clave es conocida con clave maestra y es generada por el lector, es la que inicia la comunicación entre lector

y etiqueta, como 56 *bits* se usan para cifrar y descifrar la clave es de ese tamaño, lo que equivale a que existan  $2^{56}$  claves diferentes también existe cifrado 3DES que es 3 veces el cifrado DES, obteniendo tres claves diferentes [15].

- ✓ Algoritmo internacional de cifrado de datos (IDEA, por sus siglas en inglés International Data Encryption Algorithm): Consiste en cifrar bloques de texto de 64 *bits*, para ello se usa una clave de 128 *bits*, esta genera 52 subclaves de 16 *bits* [15].
- ✓ Criptografía de clave pública o asimétrica: El texto se cifra, tomando una clave aleatoria de una lista de claves y se descifra con una clave diferente a la del cifrado, usualmente se da a conocer la clave de cifrado por eso su nombre de pública, pero solamente el usuario tendrá al acceso al mensaje descifrado porque jamás se da a conocer la clave de descifrado [15].
- ✓ Estándar de Cifrado avanzado (AES, por sus siglas en inglés Advanced Encryption Standard): Es un algoritmo de cifrado por bloques, tamaño de bloque de 128 *bits*, los datos encriptados se dividen en segmentos de 16 *bytes*, se genera luego una clave inicial para encriptar y desencriptar, y 10 subclaves [15].

Las tarjetas para el prototipo usan tecnología Mifare clásica, las mismas no soportan encriptación, pero en el mercado existen con tecnología Mifare/ Desfire, que soportan encriptación DES, triple DES y AES [16].

#### **4.1.2.5 Control de errores**

Se da para evitar la pérdida de información durante la transmisión, o para pedir una posible retransmisión [9].

- ✓ Control de paridad: Incluye un *bit* de paridad por cada byte transmitido, esta paridad puede ser paridad par o impar, la elección se da por una concordancia entre emisor y receptor [9].
- ✓ Método de control de redundancia longitudinal (LRC, por sus siglas en inglés Longitudinal Redundancy Checksum): De proceso lineal, los errores de transmisión se comprueban mediante sumas *OR* exclusivas de sus bytes de entrada si la suma final da 0, la transmisión se ha realizado con éxito, caso contrario han existido pérdidas, es un método fácil, aunque no muy eficaz cuando se trata de datos grandes mayores a 32 bytes [9].

- ✓ Método de proceso cíclico; Control de Redundancia Cíclica (CRC): Realiza una suma de comprobación segura con cantidades grandes de datos y a la vez puede identificarlos, es el resultado de una división entre el dato de entrada y un polinomio generado y así se va dividiendo el siguiente dato, si un dato da como resultado cero se encontrará el error. [9].

La tecnología Mifare que uso el prototipo usa el Método de proceso cíclico para el control de errores [17].

#### **4.1.2.6 Ventajas de RFID.**

- ✓ Al ser un sistema inalámbrico lo hace menos engorroso, tiene una mejor presentación ante el público, se trabaja en tiempo real incrementando de producción [9].
- ✓ Si se trabaja con sistemas pasivos, su duración es mucho mayor [9].
- ✓ No presentan desgaste como los sistemas magnéticos al no requerir de contacto ni línea de vista entre *tag* y lector, solamente deben estar dentro del rango de lectura [9].
- ✓ Al poseer sistemas anticolidión, se pueden leer más de dos tarjetas a la vez lo que los lleva a ser una tecnología eficiente [9].
- ✓ Se pueden escribir gran cantidad de veces, son reescribibles [9].
- ✓ Evita falsificaciones [9].
- ✓ Una antena RFID produce la quinta parte de una carga electromagnética de un celular, por tanto, no es perjudicial para la salud, la energía que consume el sistema RFID 13, 56MHz está por el 1mA [11].

#### **4.1.3 Codificación y Modulación**

##### **4.1.3.1 Codificación en banda Base.**

La codificación es una forma de adecuar la señal para darle seguridad al sistema, así mismo se debe realizar la codificación y decodificación. Los unos y ceros pueden ser representados por varios códigos lineales, los sistemas RFID a alta frecuencia con tecnología Mifare (tecnología usada en el prototipo) usan normalmente la siguiente codificación [10]:



**Código Manchester:** El uno es representado por la transición negativa en la mitad del periodo del *bit*, luego el pulso sigue la señal original medio *bit* más, vuelve a verificar y se da cuenta que por ejemplo está en una transición positiva, el *bit* de salida corresponderá a cero durante esa mitad de periodo de *bit* [10].

#### 4.1.3.2 Modulación

Para enviar información del transponder al lector, la resistencia de carga presente en el transponder conectada en paralelo con la antena, se conecta y desconecta según el flujo de datos, la amplitud de esa onda reflejada desde la *tag* es lo que se modula [10].

Los tipos de modulación usados en RFID son: Modulación por desplazamiento de amplitud (ASK, por sus siglas en inglés Amplitude Shift Keying), desplazamiento de frecuencia (FSK, por sus siglas en inglés Frequency Shift Keying), y desplazamiento de fase (PSK, por sus siglas en inglés Phase Shift Keying) [10].

**ASK:** Consiste en variar la amplitud de una portadora [10].

**FSK:** Se varía la frecuencia de la señal portadora en dos subfrecuencias [10].

**PSK:** Consiste en variar en dos estados la fase de la portadora, sea en  $0^\circ$  o  $180^\circ$  [10].

**Modulaciones con subportadora:** Usada en alta frecuencia (HF), el primer paso es modular cualquier parámetro de la portadora sea en amplitud (ASK), fase (PSK) o Frecuencia (FSK), hasta aquí esta señal modulada recibe el nombre de subportadora modulada, luego de esta modulación se realiza otra modulación de la señal portadora con la subportadora modulada. Este proceso múltiple de modulación otorga la frecuencia final de transmisión [10].

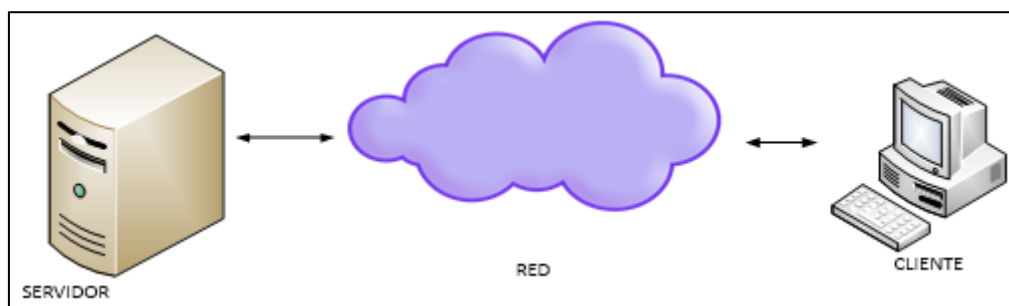
La tecnología Mifare usa modulación 100% ASK, con una subportadora de 847 KHz [17].

## 4.2 CAPÍTULO II: ALMACENAMIENTO DE INFORMACIÓN

Según el prototipo de Identificación por Radio Frecuencia (RFID, por sus siglas en inglés) que fue construido con módulos Arduino, y por las características de las tarjetas RFID, cada usuario tendrá un código único, este será leído por la tarjeta lectora RFID, luego se enviará a un sistema de almacenamiento o base de datos [18].

#### 4.2.1 Servidor de almacenamiento de información.

Ofrece servicios de alojamiento de datos, en una organización se suele encontrar más de un servidor, es el equipo físico sobre el cual se encuentra ejecutándose un software, que funciona siete días completos 24x7, lo que requiere de un hardware muy resistente preparado para problemas de calentamiento, y que permitan brindarle mantenimiento, por tanto la máquina debe tener más capacidad tanto de almacenamiento de información como de memoria principal para lograr atender a múltiples clientes, como se indica en la figura 4 [18]:



**Figura 8.** Estructura cliente-servidor.

**Fuente:** Elaboración propia

Esta es la parte del software donde después de recibir los datos por un puerto especificado, pasan a un periodo de procesamiento y luego se los guarda en una base de datos [18].

##### 4.2.1.1 Tipos de servidores de alojamiento

Debido a los múltiples requerimientos de los clientes, se pueden tener variedad de servidores, así mismo una máquina puede conectarse a múltiples servidores, entre ellos tenemos los servidores web y los de base de datos [18].

###### 4.2.1.1.1 Servidor web

Se utiliza un servidor web con una *IP* local, tiene como trabajo alojar sitios, tablas, aplicaciones, etc., los cuales estarán accesibles por los usuarios de la red a través de los navegadores [18].

Hay servidores web de paga y gratuitos, los de paga ofrecen mayores prestaciones que los gratuitos [18].

#### **4.2.1.1.2 Servidor de base de datos**

Se los utiliza para los gestores de base de datos, manejan grandes y complejos volúmenes de datos relevantes para alguna organización [18].

En la actualidad las interfaces de usuario son tan amigables que los clientes no se dan cuenta que están interactuando con una base de datos, formando parte de la vida cotidiana de manera directa o indirecta, estos datos o información de alguna forma se almacenan en un medio físico, ejemplos de ellos tenemos: información para clientes, cuentas, reservaciones de viajes, para guardar un registro de las llamadas realizadas, para almacenar información de redes de comunicaciones, transacciones bancarias, listas de empleados, búsquedas en sitios web, compras en línea, comercio en línea, inventarios, y otro sin número de aplicaciones que son atendidas mediante base de datos de gran importancia en estos sistemas, en casi todas las empresas actuales, entidades educativas, etc., [19].

Los servidores de base de datos tienen múltiples aplicaciones, hoy en día se usan en la mayoría de microempresas, empresas medianas y sobre todo en grandes empresas con la finalidad de mantener un registro de las actividades realizadas, de los productos con los que cuentan, de las ganancias, etc., [19].

Los sistemas de base de datos proporcionan información ordenada, de tal forma que acceder a estos, sea fácil, y por tanto también se evitará la redundancia de los datos, también ayuda a guardar datos de forma simultánea, sin alterar otra tabla u otra columna [19].

Es tanto su uso e importancia que son usadas en bancos, en instituciones de educación básica, superior, para información de los estudiantes, cursos, en las finanzas, ventas, producción, recursos humanos, comercio en línea, etc., [19].

##### **4.2.1.1.2.1 Principales modelos de los servidores de base de datos**

Los modelos de los servidores, se refieren a la forma en que se encuentran diseñadas las bases de datos, tanto en la parte física que indica cómo se realiza el almacenamiento de la información, es decir qué estructura se ocupa; en la parte lógica describe qué clase de datos se almacenan y la relación entre ellos y de vistas que indica interacción del usuario con la base de datos [19].

Los modelos de datos se encuentran clasificados en 4 clases:

- **Modelo entidad-relación:** Una entidad es una cosa u objeto del mundo real que es inconfundible con otros objetos, este modelo relaciona ese tipo de entidades [19].
- **Modelo de datos orientado a objetos:** En una extensión del modelo entidad-relación con el plus de encapsulación [19].
- **Modelo de datos semiestructurado:** En este modelo se hace uso de atributos para relacionarlos con un conjunto de datos [19].
- **Modelo Relacional:** Usa tablas para representar datos con sus relaciones, cada tabla está formada por columnas y a cada columna se le coloca un nombre o *ID* único, es el modelo de base de datos más utilizado y emplea el lenguaje de consulta estructurado ( *SQL*, por sus siglas en inglés *Structured Query Language*), la ventaja con respecto a los otros modelos es que este presenta una forma sencilla de representación de datos, y también otorga facilidad para realizar incluso hasta consultas demasiado complejas [19].

- **Lenguaje de consulta estructurado (*SQL*, por sus siglas en inglés *Structured Query Language*)**

Es un lenguaje que permite realizar consultas, definir la estructura de los datos, modificar los datos de la base de datos, y darle seguridad al sistema de una forma muy cómoda ante el usuario, es el lenguaje estándar para las bases de datos relacionales, como en el modelo relacional cada base de datos es un conjunto de relaciones que poseen tablas con filas y columnas, esto hace que el lenguaje *SQL* sea sencillo y elegante, de tal manera que usuarios primerizos puedan comprender el contenido de la base de datos [19].

*SQL* no es tan potente, para realizar cálculos, dentro de una misma base, como por ejemplo incrementar un valor o sumarlo con otro, para ello tiene compatibilidad con

otros programas que pueden hacer estas tareas, como, por ejemplo: java, lenguaje C, C++ y Cobol [19].

El lenguaje SQL soporta cadenas de caracteres de longitud fija (*char*), de longitud variable (*varchar*), enteros (*int*), enteros pequeños (*smallint*), números de coma fija (*numeric*), números de coma flotante (*double*), números de coma flotante con n precisión (*float*) [19].

Dentro de *SQL* tenemos dos partes fundamentales que son: el Lenguaje de definición de datos (DDL, por sus siglas en inglés Data Definition Language) y el Lenguaje de Manipulación de datos (DML, por sus siglas en inglés Data Manipulation Language). El DDL indica cómo van a ser los datos, si un dato por ejemplo es entero, char, flotante, etc. Un ejemplo se muestra en la tabla 1, donde las columnas representan los nombres de los estudiantes que corresponde a una cadena de texto es decir a datos tipo char, y las filas la información referente a su número de cédula de identidad personal que corresponden a datos enteros [19].

**Tabla 3.** Tabla de estudiantes, con información de nombre de estudiante y número de cédula

Nombre	Número de cédula
Andreina	1105681710
María	1105681716
Maribel	1105699243

**Fuente:** Elaboración propia

En cambio, la parte DML, permite la manipulación de la información, DML maneja unas ciertas funciones como [20]:

- a) **Select:** permite realizar solicitudes para que se devuelva un dato o serie de datos [20].
- b) **Insert:** Permite insertar más filas de información en una tabla [20].
- c) **Update:** Permite editar y cambiar una o varias filas que ya han sido creadas [20].
- d) **Delete:** Permite eliminar las filas o conjunto de filas creadas anteriormente [20].

Las bases de datos tendrán un cambio constante conforme se realicen cambios de inserción y borrado de datos [20].

Todo el conjunto de información de la base de datos recibe el nombre de ejemplar y el diseño de la base de datos, es decir columnas, filas, se denomina esquema, el esquema no suele tener cambios [19].


#### 4.2.2 Sistema de Gestores de Bases de datos (SGBD) comerciales y libres.

Un SGBD es el software que permite la administración, manipulación y gestión de cualquier tipo de base de datos, entre ellos tenemos el modelo relacional que es el más usado en la actualidad, por su flexibilidad y sencillez, esta herramienta sirve de interfaz entre el usuario y la base de datos [21].

Existen SGBD libre y de paga, los libres son una alternativa técnicamente viable y económicamente sostenible al software comercial, los sistemas libres suministran constantemente documentación de ayuda, que hace que su uso cada vez incremente por su bajísimo precio, pero existe el riesgo de que dicho sistema deje de estar disponible en forma gratuita y se vuelva comercial, sin embargo la elección de estos sistemas sean libres o de paga se debe realizar tomando en cuenta el soporte ofrecido, la documentación, seguridad, control de acceso a los recursos, complejidad de migración de los datos, todos estos factores dependen siempre de las restricciones particulares de la empresa u organización esto se refiere a los requerimientos que esta solicite, el personal que dispone, los recursos económicos, etc., [21].

En la alternativa SGBD libres, podemos citar como más importantes (véase Tabla 4) [20].

**Tabla 4.** SGBD gratuitos.






NOMBRE DE BASE DE DATOS	LOGOTIPO	CARACTERÍSTICAS
PostgreSQL		Es de 2 a 3 veces más lento que mysql, sistema de mayor nivel comparado con Oracle y Sybase

DB2-Express C		No se pueden aplicar parches ni la corrección de errores, no es de código abierto.
MySQL (edición gratuita)		Es de código abierto, buen rendimiento y velocidad, no se pierde información ni se corrompe los datos, se integra con PHP.

Fuente: Elaboración propia.

En la alternativa SGBD comerciales, podemos citar como más importantes [20] (véase Tabla 5).

**Tabla 5.** SGBD de paga.

NOMBRE DE BASE DE DATOS	LOGOTIPO	CARACTERÍSTICAS
MySQL (ediciones de pago)		Tiene las mismas características del MySQL libre pero con soporte técnico.
Oracle		Fácil de aprender, robusto, requiere más hardware, rendimiento elevado, escalable, multiplataforma.
Sybase		Escalable, de alto rendimiento, seguro.
Microsoft SQL SERVER		Rendimiento medio, solo trabaja en Windows.
IBM Informix		Escalable, alto rendimiento, multiplataforma.

Fuente: Elaboración propia.

#### 4.2.3 Análisis de uso de las bases de datos

La base de datos que mayor uso tiene de las anteriores nombradas es Oracle, usada en pequeñas y grandes empresas desde 1979, por su robustez, por contar con buenas herramientas muy potentes y efectivas, con muchas aplicaciones y por cumplir de la mejor manera con el estándar de Lenguaje de Consulta Estructurada (SQL, por sus

siglas en inglés) y además por ser una buena opción al aprender a programar en base de datos [20] [21] [22].

Así como Oracle tienes sus pros también presenta contras si las empresas u organizaciones que lo utilizan no cuentan con buenos recursos económicos ya que este sistema al ser bastante pesado requiere más hardware, para los servidores se usan ordenadores sumamente grandes, y además es un sistema de pago, lo que hace que para pequeñas empresas sea una alternativa poco rentable, pero como alternativa a Oracle aparece Microsoft SQLServer que es una solución de pago con rendimiento medio usado por empresas con un volumen de información elevado pero no tan elevado como Oracle. En cambio, para organizaciones o empresas que no tengan un volumen de información muy elevado, es decir para pequeñas a medianas empresas aparecen los gestores de base de datos gratuitos, los más utilizados hoy en día son PostgreSQL y MySQL [20] [21] [22].

Elegir entre PostgreSQL y MySQL parece ser una tarea bastante complicada, ya que PostgreSQL presenta ventajas comparadas a Oracle, posee una gran escalabilidad, implementa el uso de subconsultas y transacciones, la desventaja es que consume gran cantidad de recursos y es de 2 a 3 veces más lento que MySQL [20] [21] [22].

En cambio MySQL es el más usado en la actualidad, en cuanto a la parte técnica presenta debilidades comparado con PostgreSQL, pero en cambio este es más rápido, convirtiéndose en un gestor de gran rendimiento, tiene un consumo bajo en cuanto a recursos, en cuanto configuración e instalación es de gran facilidad, con respecto a otros, también la probabilidad de no entregar datos confiables es muy reducida, además el conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet, en cuanto a sus contras no tiene buena escalabilidad, carece de soportes para transacciones [20] [21] [22].

PostgreSQL presenta algunas características técnicas mejores que MySQL, sin embargo cada uno de estos gestores es apto para ciertas utilidades, dependiendo de la empresa y hay muchas de las veces que las falencias de MySQL no son de gran importancia ya que no se las usa o no son de relevancia en la empresa u organización y por ejemplo usar PostgreSQL en una empresa que necesitamos velocidad al



procesamiento de datos y que no cuente con muchos recursos conlleva pérdidas de productividad, es decir ninguno de los dos gestores son perfectos [20] [21] [22].

Por tanto, la base de datos MySQL fue elegida por no ocupar muchos recursos y por su velocidad por ser la base de datos más utilizada en páginas web, e incluso se considera que MySQL es mucho más sencillo de aprender que PostgreSQL [20] [21] [22].

Cabe recalcar que MySQL es un gestor que también se encuentra disponible en forma no gratuita, pero la única diferencia con el MySQL libre es en el soporte técnico que MySQL (edición de pago) ofrece, y la posibilidad de integrar este gestor en un software propietario [20] [21] [22].

#### **4.2.4 Ventajas de los servidores de los sistemas de gestión de base de datos (SGBD)**

Es importante trabajar con un sistema de gestión de base de datos sea cual sea este, ya que proporcionan las siguientes ventajas [23]:

- ✓ **Independencia con respecto a los datos:** Cada dato es independiente de otro, esto quiere decir que gracias a un SGBD podemos solamente mostrar los datos que queramos, ocultando otros que consideremos de acceso especial [23].
- ✓ **Acceso eficiente a los datos:** Además de almacenar los datos los SGBD ofrecen la capacidad de recuperarlos [23].
- ✓ **Integridad y seguridad de los datos:** En integridad verifica si la información ingresada es real, para el caso de la seguridad se bloquean de los datos a ciertas personas, permitiendo que solamente puedan ver lo que el administrador crea conveniente [23].
- ✓ **Administración de los datos:** La recuperación de la información es lo más eficiente posible, y un SGBD ofrece una buena administración haciéndola más fácil si los datos son compartidos por varios usuarios [23].

- ✓ **Acceso concurrente y recuperación en caso de falla:** Los SGBD cuentan con tal robustez que permite que múltiples usuarios puedan acceder a la información almacenada en una base de datos, ofrecen protección [23].
- ✓ **Reducción del tiempo de desarrollo de las aplicaciones:** Debido a que los SGBD soportan funciones de alta relevancia que son comunes con otras aplicaciones que tienen acceso a la base de datos y gracias a su interfaz de alto nivel con los datos, facilita el rápido desarrollo de las aplicaciones [23].

Según las ventajas mencionadas anteriormente en todo sistema que gestione gran cantidad de datos es indispensable el uso de los SGBD [23]

#### 4.2.5 MySQL

Es un sistema de administración de base de datos, es decir almacena información, perteneciente al modelo relacional es decir se basa en el uso tablas para representar datos con sus relaciones [24].

MySQL ha ganado popularidad por ofrecer una serie de atractivas características con respecto a los demás servidores de base de datos que se encuentran en el mercado, entre ellas podemos mencionar [24]:

- ✓ Como la versión que se utiliza para el prototipo es la versión gratuita, por tanto, tiene licencia libre [24].
- ✓ Está desarrollado en C y C++, altamente compatible con otras múltiples aplicaciones que manejan este lenguaje [24].
- ✓ Su descarga es gratuita mediante su página oficial [www.mysql.com](http://www.mysql.com), con licencia GPL [24].
- ✓ Utiliza el lenguaje SQL (Structure Query Language- Lenguaje de consulta estructurado) [24].
- ✓ Es un sistema cliente/servidor [24].
- ✓ Existe bastante información en línea, es posible encontrar gran cantidad de software desarrollado sobre MySQL o que soporta MySQL [24].

- ✓ Es portable, puede ser llevado a cualquier plataforma informática, se distribuyen ejecutables para más de veinte plataformas diferentes, como lo son Mac X, Unix, y Microsoft Windows [24].
- ✓ Es muy destacable su velocidad de respuesta [24].
- ✓ Su administración se basa en usuarios y privilegios [24].
- ✓ Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ' [24].
- ✓ Sus opciones de conectividad abarcan TCP/IP, *sockets* UNIX y *sockets* NT, además de soportar completamente Conectividad abierta de base de datos (ODBC, por sus siglas en inglés Open Database Connectivity) [24].
- ✓ Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas [24].
- ✓ Está optimizado para equipos de múltiples procesadores [24].

Todas las características mencionadas anteriormente convierten a MySQL como un gestor de base de datos popular tanto en pequeñas, medianas y hasta grandes corporaciones. MySQL en sí es un poco tedioso, ya que necesita código para eliminar, crear, actualizar, usuarios, tablas; por tanto, aparece phpmyadmin, que hace lo mismo que MySQL a diferencia que ofrece al usuario una interfaz gráfica para realizar todas estas tareas [24].

#### **4.2.5.1 PHPMYADMIN**

Permite una administración de MySQL, pero de forma gráfica, basado en *PHP*, es muy intuitivo y fácil de utilizar, así mismo está disponible de forma gratuita en línea [24].

Se podrá gestionar, crear, actualizar y eliminar base de datos, dentro de las mismas también se puede actualizar, crear, eliminar usuarios, consultar, manipular, tablas, crear campos como usuarios con tamaños determinados. También permite la importación y exportación de base de datos [24].

#### **4.2.6 PHP**

Por sus siglas en inglés Pre Hypertext Processor - Preprocesador de hipertexto, permite realizar una multitud de tareas útiles para el desarrollo web, es un lenguaje de

programación del lado del servidor que se caracteriza por su potencia, robustez, versatilidad, flexibilidad y modularidad, es open source, tiene como apoyo una base de datos que es MySQL, sin esta base de datos *PHP* no tendría sentido, en cambio una base de datos sin el lenguaje de programación que permita gestionarla sería algo inútil, es decir *PHP* está ligado a MySQL [24].

*PHP* es un lenguaje que por su flexibilidad resulta bastante fácil aprender y utilizar, especialmente para programadores familiarizados con el lenguaje C, Perl o Java ya que tiene bastante parecido a estos lenguajes, estable, usa menor cantidad de recursos, en comparación a otros lenguajes, es multiplataforma al interactuar con más de 20 tipos de bases de datos [24].

El código *PHP* tiene la estructura básica del Lenguaje de Marcas de Hipertexto (HTML, por sus siglas en inglés HyperText Markup Language) y es compatible con Java. Así mismo en *PHP* se puede trabajar con datos de números enteros, números en coma flotante, cadena de caracteres, arrays y objetos [24].

## **CONECTIVIDAD DESDE PHP A MYSQL**

Usar el lenguaje *PHP* conjuntamente con MySQL permite la construcción, de una manera sencilla y eficiente de sitios web dinámicos [24].

*PHP* mediante el uso de instrucciones como `mysql_connect ()`, permite la conexión con una base de datos creada en *MySQL* o si se prefiere la interfaz gráfica PHPMYADMIN, la instrucción, contiene tres parámetros [24]:

`mysql_connect (servidor, usuario, contraseña)`

Que en el código del prototipo se presenta como:

`$con = mysql_connect ($host, $user, $pw)`

Al realizar la apertura de la conexión, se coloca ubicación de la base de datos, el usuario y la contraseña para poder ingresar [24]:

`$host = "localhost";`  
`$user = "root";`  
`$pw = "telecomunicaciones";`

Luego de realizar la apertura del PHPMYAdmin, se debe seleccionar la base de datos, con la siguiente función [24]:

```
mysql_select_db ($db,$con)
```

Donde \$db corresponde al nombre de la base de datos creada y \$con a los parámetros de servidor, usuario y contraseña [24].

Cabe recalcar que cada vez que se finalice la acción requerida la conexión se cierra automáticamente, o a su vez se puede utilizar la siguiente función [24]:

```
mysql_close ($con)
```

*PHP* permite ejecutar sentencias SQL, sobre la base de datos con la que se conecta [24].

### **4.3 CAPÍTULO III: PORTALES CAUTIVOS**

#### **4.3.1 Portales cautivos con *hotspot***

Un portal cautivo es un gestor de tráfico, se ocupa *hotspot* cuando al intentar navegar normalmente por internet se muestra una ventana de autenticación en la que se pide ingresar nombre de usuario y contraseña previamente asignados, en dicha ventana también puede haber todo tipo de información, sucediendo esto en cualquier navegador a utilizar [25] [26].

Un portal cautivo muestra independencia del sistema operativo, es decir puede funcionar tanto en Windows como en Linux o Mac OS, también filtra los paquetes, por dirección *MAC*, *IP* o por *URLs*, otorgándonos una red optimizada, la misma ha de ser con poco o ningún mantenimiento por parte del usuario [25] [26].

Se puede utilizar tanto para máquinas de mesa (internet cableado), o incluso para acceso por medio inalámbrico [25] [26].

El portal cautivo no necesita instalación de software en el cliente, esté asegura el enrutamiento de los clientes a la pantalla de inicio de sesión [25] [26].

Los portales cautivos además sirven para administrar el ancho de banda es decir establecer un ancho de banda tanto de subida como de bajada para los usuarios y controlar la navegación en internet es decir el control de tráfico que viaja por la red, el tiempo de uso del servicio [25] [26].

### **Sitios donde son utilizados los *hotspot***

- *Wi-Fi HotSpot* Redes: Se puede construir una red *wifi hotspot* a partir de una red cableada y con cualquier hardware [27].
- ISP: Los Proveedores de servicio de internet (ISP, por sus siglas en inglés), crean planes de pago para los servicios de internet [27].
- Se puede incrementar los ingresos tanto en restaurante, cafeterías, centros de negocios, hoteles, resort al ofrecer internet gratuito por un tiempo limitado a la vez que se gratifica la lealtad de los clientes y se logra conseguir mayor cantidad de clientes [27].
- Para entornos educativos como: universidades, escuelas, bibliotecas, es de gran necesidad la implementación de un software de este tipo, porque permite bloquear sitios web no aptos para estudiantes, limitar el ancho de banda por usuario, seguimiento de registros de uso, y obtener estadísticas en tiempo real [27].
- Empresas, Gobierno: En el caso de las empresas, un portal cautivo permite que los administradores bloqueen ciertas páginas que de una u otra forma contribuyan a la distracción del trabajador, que no permiten su rendimiento al máximo [27].
- En la mayoría de los centros públicos se usan, permite ofrecer al turista y al mismo originario acceso a internet con una buena velocidad con un tiempo limitado para que no exista ningún tipo de congestión y lograr una correcta administración de ancho de banda [27].

### **Utilidades de los portales cautivos**

- Para controlar el número de usuarios que pueden acceder a internet, es así como *Laptops, Tablets, Smartphones*, con el fin de mejorar el ancho de banda y por consiguiente las velocidades de acceso [27].
- Manejo de ancho de banda, carga y descarga, puede poner un determinado ancho de banda, por ejemplo: un ancho de banda mayor para clientes administrativos y uno menor para estudiantes o viceversa [27].
- Ver la actividad del cliente, gracias a esto podemos hacer un análisis de qué páginas son más usadas por los estudiantes, para el caso de la Universidad

Nacional de Loja el interés radica en conocer cuánta acogida tiene el sistema bibliotecario *on-line* implantado en una base de datos que alberga gran cantidad de artículos científicos, acceso que se permite desde el mismo instituto educativo [27].

- Para limitar el tiempo de uso de la red cableada o inalámbrica [27].
- Ofrece estadísticas en tiempo real [27].
- Especificar niveles de acceso a contenidos [27].

#### **4.3.2 Tipos de portales cautivos**

Existen portales cautivos por hardware y por software:

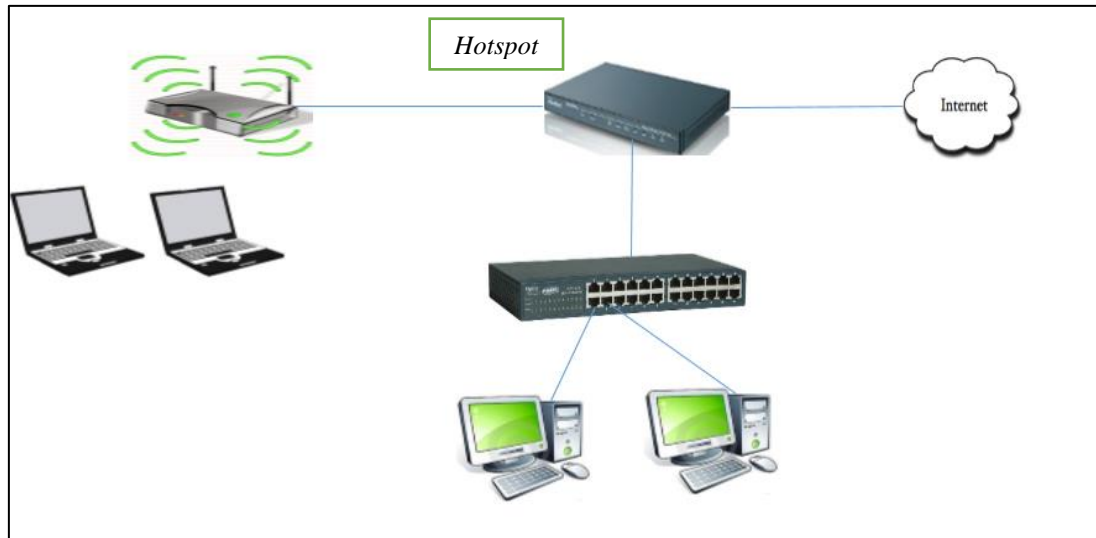
##### **4.3.2.1 Portales cautivos por hardware**

Son equipos que permiten la administración del acceso de usuarios a internet, pero se implementan sin necesidad de algún computador, ni otro equipo complementario [25] [26].

Algunos ejemplos de dispositivos que implementen este tipo de portal cautivo se citan a continuación:

- *Cisco BBSM-Hotspot* [25] [26].
- *Cisco Site Selection Gateway (SSG) / Subscriber Edge Services (SESM)* [25] [26].
- *Nomadix Gateway* [25] [26].
- *Aptilo Access Gateway* [25] [26].
- *4ipnet Hotspot Gateway* [25] [26].
- *MikroTik RouterOS* [25] [26].
- *Ovislink AirLive MW-2000S* [25] [26].
- *Endian Hotspot* [25] [26].
- Entre otros

La estructura básica de un portal cautivo por *hardware* es como lo indica la figura 5:



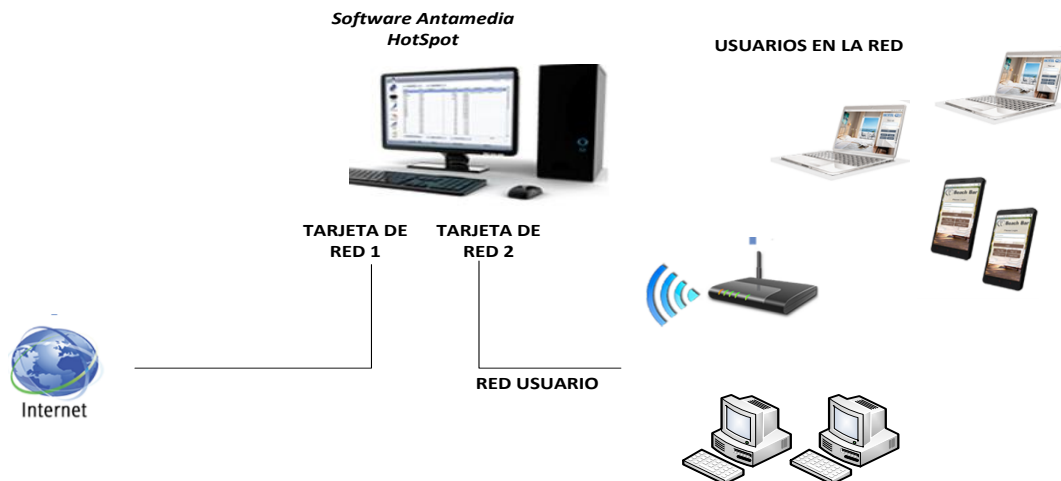
**Figura 9.** Portal cautivo, dispositivo Endian [28].

Como podemos observar en la figura 5, los portales cautivos por hardware no requieren de otro equipo adicional que contenga el gestor de tráfico, pues estos llevan dentro el software de control, dicha característica es bastante flexible, y económica.

#### **4.3.2.2 Portales cautivos por software**

Al contrario de los portales cautivos por hardware que no es necesario un computador para dar dicho servicio, un portal cautivo por software requiere de un computador con dos interfaces de red: la una para conectarse al *Access Point* y la otra a la salida de internet, donde se instala el programa que el administrador de red vea conveniente (véase figura 6) [25].





**Figura 10.** Portal cautivo en Antamedia [27].

Como se observa en la figura 6 es una red en Antamedia *Hotspot*, que puede funcionar en Linux y en Windows, de igual forma, existen variedad de programas en el mercado.

Dentro del portal cautivo por software también existe un servicio contratado, es decir existen empresas que brindan dichos servicios y realizan mantenimiento, gestión técnica, informes, etc., [26].

#### **4.3.2.3 Similitudes y diferencias entre portales cautivo por software y por hardware**

##### **☒ Similitudes**

Tanto el portal cautivo por software y por hardware poseen:

- Interfaz gráfica [25].
- Estadísticas de uso por usuario [25].
- Configuración en base a parametrización [25].
- Autenticación centralizada [25].
- Al no existir latencia por cifrado es mucho más rápido [25].
- Soluciones comerciales y libres [25].
- Seguridad basada en autenticación [25].

## ☒ Diferencias

Las diferencias de ambos tipos de portales cautivos se muestran en la Tabla 6.

**Tabla 6.** Diferencia de portales cautivos [25].

Característica	Portal cautivo por software			Portal cautivo por hardware		
	Bajo	Alto	Muy alto	Bajo	Alto	Muy alto
Precio	x				x	
Dificultad de implementación		x			x	
Estabilidad		x				x
Hardware adicional			x	x		

**Fuente:** Elaboración propia

La diferencia más sobresaliente es el precio, los portales por hardware no requieren un equipo adicional, por tanto son flexibles, presentan buenas características sin embargo para una futura implementación se debería tomar en cuenta los precios porque para que ese sistema sea bastante robusto, el precio del equipo con *hotspot* interno excede los 1000 dólares, pudiendo hacer lo mismo con una pc que tenga dos tarjetas de red y un punto de acceso cuyo presupuesto va desde los 199 a 799 dólares, además se debería considerar que dichos equipos sirven para grandes empresas que presentan gran demanda [25].

Sin embargo, para una cantidad media de estudiantes, el basado en hardware, es idóneo, principalmente por su flexibilidad y porque en el mercado se han encontrado equipos MikroTik que satisfacen a 200 usuarios con precios inferiores a 150 dólares americanos, [25].

### 4.3.3 MikroTik

Dentro de las múltiples marcas que ofrecen incluido en sus equipos el software del portal cautivo, se ha escogido la marca MikroTik [29] .

La flexibilidad que ofrecen los dispositivos de la marca MikroTik es asombrosa, existe bastante información en la red, sus routers tienen un funcionamiento igual a un router propietario, pero con costos significativamente menores, sus productos son compatibles

con las otras tecnologías como cisco, Huawei, Dell, Hp, 3com, etc., además es sencillo de configurar y de dar mantenimiento [29].

Los equipos MikroTik representan una gran solución a la hora de elección de dispositivos, esta marca posee RouterOS que es el sistema operativo basado en Linux, permite tener un control total del sistema *hotspot* y además podemos acoplarlo con otras soluciones como son un web cache en sistema Linux, identificación y priorización de tráfico, balanceo de conexiones WAN, enlaces punto a punto, etc., [29].

También ofrece aplicaciones de última tecnología para impedir que alguien pueda entrar en tu red como es *Firewall Nat* [29].

Todos sus equipos de licencia 4, 5 y 6 soportan *hotspot*, la diferencia es que el nivel 4 solamente se acepta hasta 200 usuarios para el nivel 5 incrementa a 500 y así mismo el nivel 6 da soporte a una cantidad mucho mayor, dentro del *hotspot* permiten el soporte de un servidor Radius para la autenticación y contabilidad [29].

La licencia no es la limitante, a la hora de elección del producto que incluya el software *hotspot*, por tanto, hay que tener muy en cuenta el dimensionamiento del hardware [29].

Para escogerlo se tomó en cuenta la facilidad al usar, robustez, escalabilidad, autonomía y, seguridad [29].

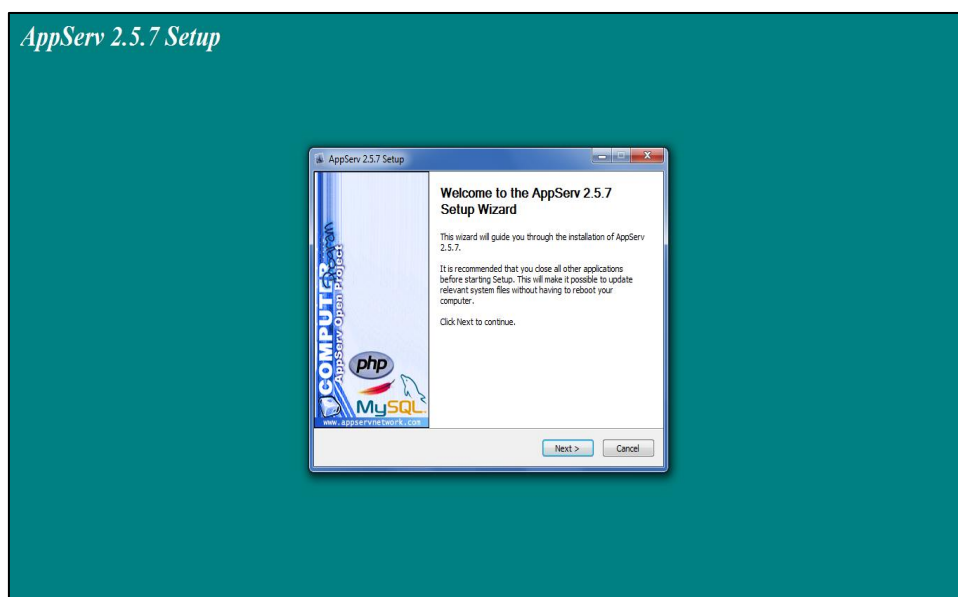
## 5. MATERIALES Y MÉTODOS

### 5.1 Descripción de los sistemas de autoidentificación.

Antes de la descripción de los sistemas de identificación se indicará los pasos para instalación y creación de la base de datos, utilizando AppServ:

#### 5.1.1 Instalación de AppServ

La instalación es sencilla, se descarga el paquete App Server de la página oficial y finalmente se ejecuta como administrador, obteniendo inicialmente la figura 7 y los pasos restantes son solo dar clic en el botón next manteniendo las configuraciones por defecto seleccionadas.



**Figura 11.** Instalación de App Server.

Se coloca el nombre del servidor, para este caso “localhost”, la contraseña, y puerto para el redireccionamiento de los datos (puerto 8000), por último, se inicializa Apache y MySQL.

Para lograr una comunicación con el lector RFID y la base de datos en MySQL se configura el paquete www que se encuentra en App server, localizado normalmente en el disco local (C:\AppSer), del ordenador que va a ser utilizado como servidor, dentro

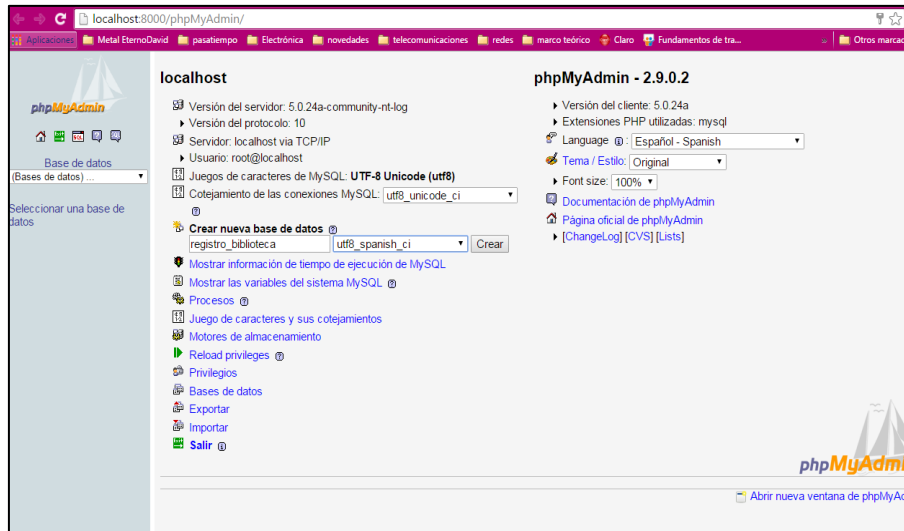
de esta carpeta se creará un archivo *PHP*, llamado reportes.php, todo depende donde se instaló.

Para crear dicho archivo hacemos uso de un editor de texto; con las características básicas de un entorno de desarrollo integrado que ocupa muy poco espacio en el disco del computador (servidor), sublime text.

Mediante el IDE (Integrated Development Environment, Entorno de Desarrollo Integrado) sublime se agrega al archivo reportes.php, el nombre de usuario y la contraseña esto ayuda a la conexión con la base de datos que se crea posteriormente en MySQL a la que solo se podrán ingresar con autenticación añadiendo seguridad, también se realiza la configuración para que sea posible la interacción entre las placas de Arduino y *PHP*, y puedan guardarse los datos en cuanto estos sean leídos. Luego se procede a abrir el navegador web donde se colocará la dirección de la máquina local, y el puerto (localhost: 8000), en la que se encuentra el paquete, la instalación de la base de datos se hará usando phpMyAdmin, que es una herramienta con interfaz gráfica de usuario que facilita en su totalidad el trabajo y es muy amigable con el usuario.

Por último, se crea una base de datos en el apartado “Crear nueva base de datos” colocando cualquier nombre, por ejemplo, registro.

Y en la parte de Cotejamiento que se refiere al tipo de codificación, se debe escoger una en la que se pueda diferenciar y utilizar letras que coincidan con el idioma (español) por ejemplo para poder reconocer y utilizar la letra “ñ” utilizamos el cotejamiento “utf8\_unicode\_ci” (véase figura 8).



**Figura 12.** Creando una base de datos.

Como MySQL es un sistema relacional basado en tablas, se realiza la creación de 4 tablas:

1. registro: se usará para la sala 1, los campos a usar son 8, siendo estos los que se indican en la figura 9.

	Campo
<input type="checkbox"/>	<u>id_registro</u>
<input type="checkbox"/>	id_persona
<input type="checkbox"/>	persona
<input type="checkbox"/>	carrera
<input type="checkbox"/>	modulo
<input type="checkbox"/>	maquina
<input type="checkbox"/>	codigocedula
<input type="checkbox"/>	FECHA

**Figura 13.** Campos requeridos para la tabla de registro de la Sala 1.

Por último se configura los campos, quedando como se indica la figura 10.

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción							
<input type="checkbox"/>	id_registro	int(10)			No		auto_increment								
<input type="checkbox"/>	id_persona	int(10)			No										
<input type="checkbox"/>	persona	varchar(100)	utf8_spanish_ci		No										
<input type="checkbox"/>	carrera	varchar(8)	utf8_spanish_ci		No										
<input type="checkbox"/>	modulo	varchar(4)	utf8_spanish_ci		No										
<input type="checkbox"/>	maquina	int(4)			No	0									
<input type="checkbox"/>	codigocedula	varchar(11)	ascii_general_ci		No										
<input type="checkbox"/>	FECHA	timestamp			No	CURRENT_TIMESTAMP									

**Figura 14.** Modificación de los campos de la tabla “registro”.

A continuación se explica cada uno de los campos:

- ✓ id\_registro: Corresponde al número de estudiantes que ingresan, siendo este un campo de tipo entero, que será autoincrementable.
- ✓ id\_persona: Es del tipo entero y corresponde a un número de identificación específico para el usuario, que se toma de otra tabla usuarios que se indicará posteriormente.
- ✓ persona: Es el nombre del usuario, tomado de la tabla “usuarios”, por tanto es de tipo varchar.
- ✓ carrera: Es de tipo varchar, tomada de la tabla “usuarios”, por ejemplo, se refiere si el usuario es administrativo, o se encuentra cursando la carrera de Ingeniería en Electrónica y Telecomunicaciones (CIEYT), o Carrera de Ingeniería en Electromecánica (CIEM), según corresponda.
- ✓ modulo: Se toma de la tabla usuarios, corresponde al número en romanos del módulo en el que se encuentra cursando el usuario, si se da el caso, o especificando si se encuentra egresado o es administrativo, por tanto, es del tipo varchar.
- ✓ maquina: Se toma del programa en Java, indica el número de máquina que se le ha asignado al usuario para su respectivo uso, es del tipo entero.
- ✓ codigocedula: Se toma dependiendo del caso, si el usuario hizo uso de la tag, dicho campo se llenará con los datos de la tabla tarjetas (de la que se hablara posteriormente) caso contrario si el usuario usa su número de cédula los datos se tomaran de la tabla usuarios, éste campo es del tipo varchar.

- ✓ FECHA: la fecha es de tipo timestamp, ya que este contiene informacion de fecha y hora: *Timestamp* (fecha; hora).

Cuando se realiza la lectura mediante el prototipo, la tabla queda de la siguiente forma (véase figura 11):

	id_registro	id_persona	persona	carrera	modulo	maquina	codigocedula	FECHA
<input type="checkbox"/>  	843	9	Hugo Adrian Idrovo Contento	CIEYT	VIII	1	D7812054	2016-09-21 17:07:16

**Figura 15.** Tabla que incluya la lectura de las tarjetas RFID.

Cuando se realiza el ingreso mediante la interfaz de Java usando el número de cédula, la tabla queda de la siguiente forma (véase figura 12):

	id_registro	id_persona	persona	carrera	modulo	maquina	codigocedula	FECHA
<input type="checkbox"/>  	844	1	Andreina Veronica Chunchu Roman	CIEYT	E	1	1105681710	2016-09-27 23:31:25





































**Figura 16.** Tabla que incluya la lectura de las tarjetas RFID de la Sala 1.

- registro\_wifi: se usará para la sala 2, los campos a usar son 6, siendo estos los que se indican en la figura 13.

Campo
<u>id_registro</u>
persona
carrera
modulo
fecha
id_persona

**Figura 17.** Campos requeridos para la tabla de registro de la Sala 2.

De igual forma se procede a la configuración de los campos, quedando como lo indica la figura 14.

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	<u>id_registro</u>	int(10)			No		auto_increment	      
<input type="checkbox"/>	persona	varchar(100)	utf8_spanish_ci		No			      
<input type="checkbox"/>	carrera	varchar(8)	utf8_spanish_ci		No			      
<input type="checkbox"/>	modulo	varchar(4)	utf8_spanish_ci		No			      
<input type="checkbox"/>	fecha	timestamp			No	CURRENT_TIMESTAMP		      
<input type="checkbox"/>	id_persona	int(10)			No			      

**Figura 18.** Modificación de los campos de la tabla “registro\_wifi”.



Ya que los campos de la tabla registro se tomaron de otras tablas (usuarios y tarjetas), de igual forma en esta tabla se toman los mismos datos, simplemente se realiza otro registro, dado que el usuario pedirá acceso a otra sala, estos registros no pueden estar unidos ambos en una misma tabla, dado que las bibliotecarias, contabilizan dos registros separados tanto para uso de máquinas de mesa como para el espacio para el internet inalámbrico, los sistemas por lo tanto se encargaran de hacer dos diferentes registros.

Quedando éste como se indica la figura 15.

id_registro	persona	carrera	modulo	fecha	id_persona
151	Guamo Morocho Andrea Katherine	CIEYT	VI	2016-04-24 13:41:40	6
152	Delgado Tapia Jeffry Vicente	CIEM	VI	2016-04-24 13:41:42	7

**Figura 19.** Tabla que incluya la lectura de las tarjetas RFID de la Sala 2.

3. tarjetas: Esta tabla sirve como referencia para la tabla “registro”, en la que se incluye el código de la tarjeta. En la figura 16 se puede observar los campos de esta tabla.

id_tarjeta	codigo
1	70BF6C44
2	0772F97C
3	A7092A54
4	17C81754
5	B7802254
6	B7B71854
7	976E2654
8	37981D54
9	D7812054
10	87FB1D54

**Figura 20.** Campos de la tabla “tarjetas”

La razón de tener 10 códigos, es por el número de tarjetas.

4. usuarios: Esta tabla sirve como referencia para la tabla “registro” y “registro\_wifi” (véase la figura 17).

id_persona	cedula	persona	estado	id_tarjeta	carrera	modulo
1	1105681710	Andreina Veronica Chuncho Roman	0	1	CIEYT	E
2	1105112476	Guillermo Xavier Palacios Arias	0	2	CIEM	IX
3	1104737273	Vicente Paul Quezada Patiño	1	3	Administ	AD
4	1104059215	Ernesto Vladimir Palacios Merino	1	4	Administ	AD
5	1104105026	Masaco Pinta Angel Ubaldo	0	5	CIEYT	IX
6	1104857444	Guamo Morocho Andrea Katherine	0	6	CIEYT	VI
7	1312554619	Delgado Tapia Jeffry Vicente	0	7	CIEM	VI
8	1105679151	Cedillo Benavides Lennin Stalin	0	8	CIEYT	IX
9	1900597830	Hugo Adrian Idrovo Contento	0	9	CIEYT	VIII
10	1900708858	Remache Montaña Jhoanna Elizabeth	0	10	CIEM	XI

**Figura 21.** Campos de la tabla “tarjetas”

De esta tabla se toma, el nombre de usuario, la carrera correspondiente si el usuario es administrativo o estudiante, el módulo cursado, el id\_tarjeta que se toma de la tabla “tarjetas”, estos parámetros son necesarios para los registros en ambas salas.

En cuanto al almacenamiento es importante mencionar el servidor de alojamiento web utilizado.

2FreeHosting, ofrece servicios de hosting gratis, para crear una cuenta se ingresa a la página oficial [www.2freehosting.com](http://www.2freehosting.com), véase la figura 18.



**Figura 22.** Página oficial del servidor web.

Los pasos para la creación de una base de datos son:

- Se crea la cuenta, llenando campos de información básica.
- Dentro del servidor, se crea una base de datos, para alojar la tabla de registro.

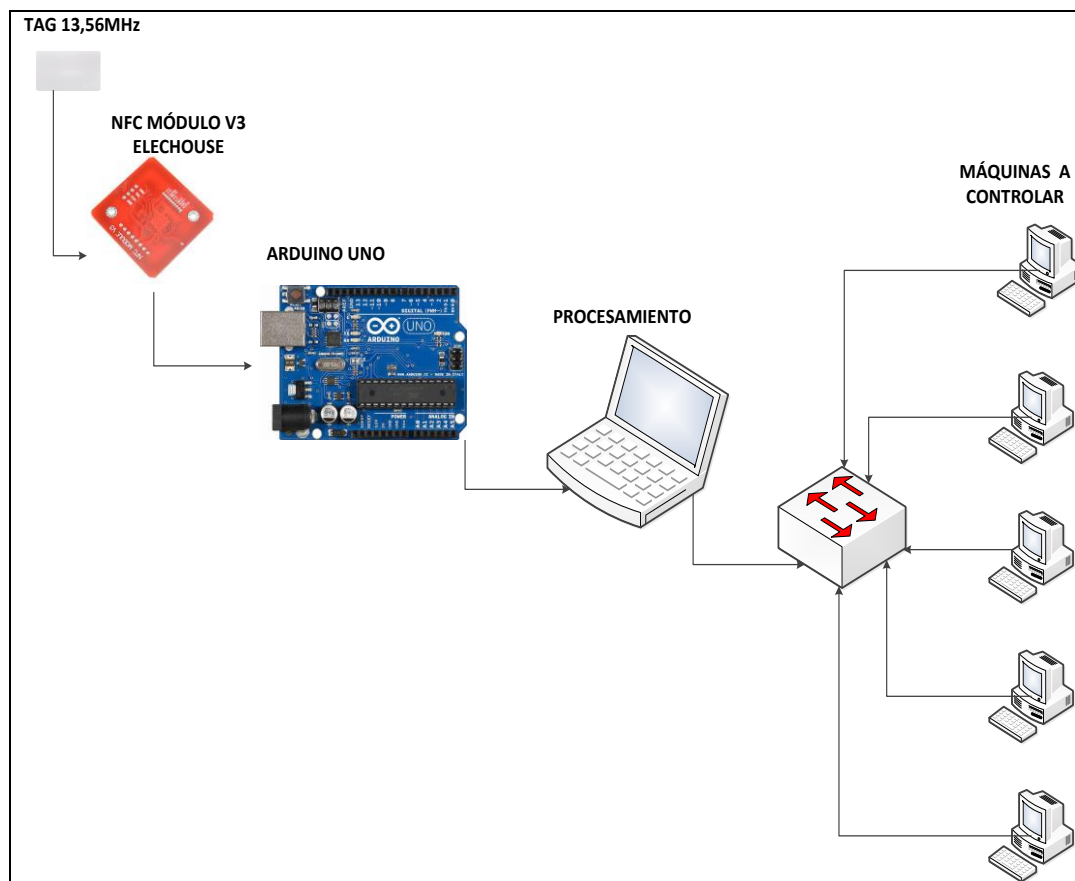
- Solamente se ingresa al cPanel, luego a Base de datos, se llena los requerimientos, y se pulsa en crear, véase la figura 19.

The screenshot shows a web form titled "Crear una nueva base de datos de MySQL y usuario nuevo" in red text. The form contains four input fields: "Nombre base de datos MySQL :", "Usuario MySQL :", "Contraseña:", and "Confirmar contraseña:". The first two fields have pre-filled values "u228809984\_" and "ADMIN" respectively, with a dropdown menu showing "REGIS" and "ADMIN". The password fields are masked with dots. To the right of the password fields is a "Generate password" button. At the bottom right of the form is a "Crear" button.

**Figura 23.** Creación de BD en el servidor web.

### **5.1.2 Sistema SALA 1: Activación y desactivación de las computadoras de mesa, mediante el registro automático.**

Para el presente sistema se toma en cuenta el siguiente diseño:



**Figura 24.** Sistema Sala 1.

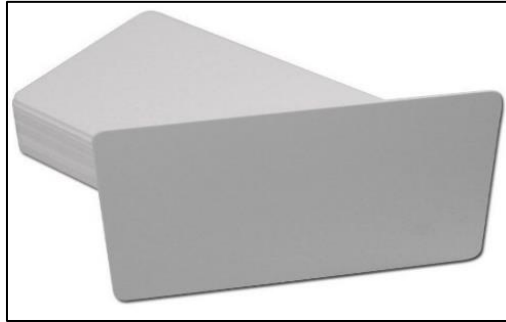
**Fuente:** Elaboración propia.

Los elementos a usar son: tarjetas de identificación RFID a 13,56 MHz (usadas tanto en sala 1 y sala 2), un módulo RFID de la marca Elechouse V3, una placa Arduino Uno y un computador donde se encontrará la base de datos, que debe estar en la misma red de los computadores de mesa de la sala 1, a continuación, se detalla cada elemento que se usaron:

- **Tarjeta de Acceso de Entrada RFID 13,56 MHz proximidad RFID**

La *Tags* o tarjeta están hechas de material PVC, resistente al agua, funcionan a una frecuencia de 13,56 MHz, según el estándar *ISO/IEC 14443A* y tecnología Mifare como se indica la figura 21 [30].

- ✓ Material: Plástico [30].
- ✓ Color: Blanco [30].
- ✓ Tamaño: 85.5x 54x 0.87mm / 3,37 "x 2,12" x 0,034 "(L \* W \* T) [30];
- ✓ Peso neto: 54g [30].
- ✓ Etiquetas pasivas [30].
- ✓ Distancia operativa: Hasta 100 mm, tarjeta inteligente sin contacto [30].
- ✓ Lectura/escritura 1 Kbyte [30].
- ✓ La transmisión de datos y suministro de energía se realiza por inducción electromagnética (No requiere batería) [30].
- ✓ Frecuencia operativa: 13.56 MHz [30].
- ✓ Velocidad de transferencia de datos: 106 kbit/s;
- ✓ Alta integridad de datos: 16 *bit* de Control de Redundancia Cíclica (CRC), paridad, *bit coding*, *bit counting* [30].
- ✓ 1KByte, organizado en 16 sectores con 4 bloques de 16 *bytes* cada uno [30].
- ✓ Retención de datos de 10 años [30].
- ✓ Resistencia de escritura de 100.000 ciclos;
- ✓ Autenticación mutua en tres pasos [30].
- ✓ Retención de datos del canal RF con protección de ataque repetitiva [30].
- ✓ Número de serie único [30].
- ✓ Tecnología Mifare [30].
- ✓ RF interface: ISO 14443A [30].
- ✓ Temperatura: -10°C a +50°C [30].
- ✓ *Middleware*: gestiona el intercambio de información [30].



**Figura 25.** Tag RFID 13,56 MHz [30].

- **Módulo RFID NFC V3**

Este pequeño lector con dimensiones de 42,7mm x 40.4 mm y 4mm de ancho, perteneciente a la marca Elechouse, soporta comunicación I2C (Inter- Integrated Circuit, Circuito Interintegrado), HSU (Hight Speed UART, UART de alta velocidad), SPI (Serial Peripheral Interface, Interfaz Serial Periférica), cubre distancias de 5-7 cm, es compatible con Arduino como lo indica la figura 22 [31].

Trabaja como lector y escritor, puede leer Tags con el protocolo *ISO/IEC* 1443-A, para su funcionamiento se requiere un voltaje de alimentación de 3.3 v – 5v, voltaje suficiente para poder activar los módulos de *I2C*, *HSU*, en este caso se usará SPI [31].

Para elegir entre esas opciones, el lector posee un switch, el cual puede estar en las siguientes posiciones para poder elegir la comunicación requerida (véase Tabla 7) [31].

**Tabla 7.** Opciones de lectura de la tag RFID NFC V3 [31].

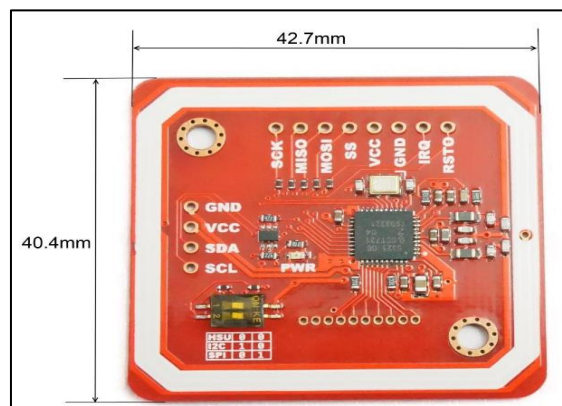
Interfaz de trabajo	Canal 1	Canal 2
HSU	0	0
I2C	1	0
SPI	0	1

**Fuente:** Elaboración propia

La antena se encuentra debajo de la pintura blanca que está bordeando la placa de lector [31].

## Características

1. Soporte I2C, SPI y HSU (UART de alta velocidad) [31].
2. RFID compatibilidad con el modo de lectura/escritura [31].
  - Mifare 1k, 4k, Ultralight, y tarjetas DesFire<sup>1</sup> [31].
  - Tarjetas *ISO/IEC* 14443-4 tales como CD97BX, luz de CD, DesFire, P5CN072 (SMX) [31].
  - Tarjetas Innovision Jewel tales como tarjetas de IRT5001 [31].
  - Tarjetas FeliCa tales como RCS\_860 y RCS\_854 [31].
3. El *plug and play*, compatible con Arduino [31].
4. Antena construida en el PCB, con distancia de comunicación de 5 cm ~ 7 cm [31].
5. Cambiador de nivel abordo, estándar *TTL*<sup>2</sup> 5V para *I2C* y *UART*, *TTL* 3.3V *SPI* [31].
6. Trabaja como lector/grabador RFID [31].
7. Trabajo como tarjeta 1443-A o una tarjeta virtual [31].
8. Soporte *NFC* con teléfono Android [31].
9. Tamaño pequeño: 43 mm \* 41 mm \* 4 mm [31].



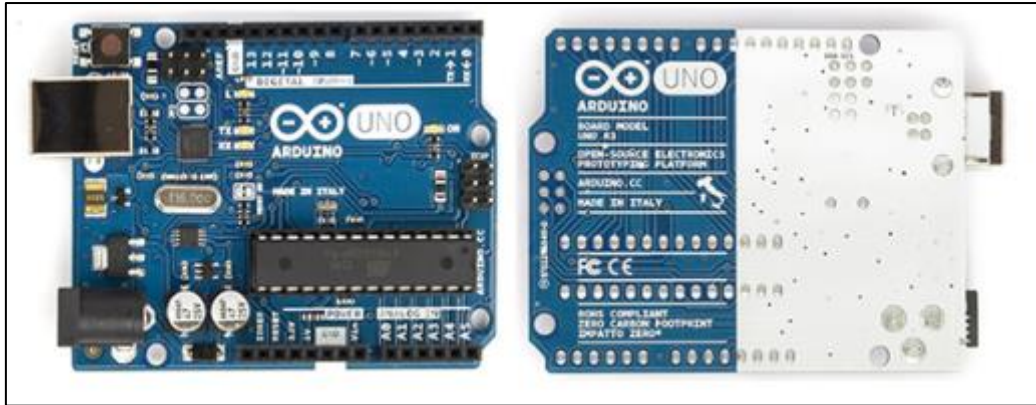
**Figura 26.** Placa RFID Elechouse V3 [31].

- **Placa Arduino Uno**

<sup>1</sup> Mifare, Ultralight y Desfire son tecnologías que soporta RFID, difieren del tipo de cifrado, y del cumplimiento de la partes de la norma ISO/IEC 1443A.

<sup>2</sup> TTL por sus siglas en inglés Transistor- Transistor Logic, es decir, lógica de transistor a transistor.

El Arduino Uno es una placa electrónica basada en el ATmega 328, Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 pueden utilizarse para salidas *PWM*), 6 entradas analógicas, un 16 MHz resonador cerámico, una conexión *USB*, un conector de alimentación y un botón de reinicio, como lo indica la figura 23 [32].



**Figura 27.** Placa Arduino Uno [32].

## Alimentación

El Arduino Uno puede ser alimentado a través de la conexión *USB* o con una fuente de alimentación externa [32].

La placa puede funcionar con un suministro externo de 6 a 20 voltios. Si se suministra con menos de 7V el pin de 5V puede suministrar menos de cinco voltios y el funcionamiento puede ser inestable. Si se utiliza más de 12 V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios [32].

Los pines de alimentación son como sigue:

- **VIN:** El voltaje de entrada a la placa Arduino cuando se trata de utilizar una fuente de alimentación externa (en oposición a 5 voltios de la conexión *USB* u otra fuente de alimentación regulada). Suministro de tensión a través de este pin, o, si el suministro de tensión a través de la toma de alimentación, acceso a él a través de este pin [32].
- **5V:** Pin de salida, suministra 5V. La placa puede ser alimentado ya sea desde la toma de alimentación de CC (7 - 12 V), el conector *USB* (5V), o por el pin *VIN*



(7-12V), y no por este pin ya que este pasa por el regulador, y puede causar daños en la placa [32].

- 3V3: Un suministro de 3,3 voltios generada por el regulador de a bordo. Sorteo de corriente máxima es de 50 mA [32].
- *GND* pines de tierra [32].
- Instrucción IOREF: Este pin de la placa Arduino proporciona la referencia de tensión con la que opera el microcontrolador, trabaja con 5V o 3.3V [32].

## Memoria

El ATmega 328 tiene 32 KB (con 0,5 KB utilizan para el gestor de arranque). También cuenta con 2 KB de *SRAM* y 1 KB de *EEPROM* (que puede ser leído y escrito con la librería *EEPROM*) [32].

## Entrada y Salida

Cada uno de los 14 pines digitales en la placa Arduino Uno se puede utilizar como una entrada o salida, utilizando `pinMode ()`, `digitalWrite ()`, y `digitalRead ()` [32].

Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de *pull-up* (desconectado por defecto) de 20 a 50 kohm. Además, algunos pines tienen funciones especializadas [32]:

- Serial: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir datos en serie (TX). Estos se encuentran conectadas a los pines correspondientes del ATmega 8U2 USB-to-TTL chip de serie [32].
- Interrupciones externas: 2 y 3 Estos pines pueden configurarse para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor [32].
- PWM: 3, 5, 6, 9, 10, 11, proporcionan una salida PWM de 8 *bits* con el `analogWrite ()` [32].
- *SPI*: 10 (*SS*), 11 (*MOSI*), 12 (*MISO*), 13 (*SCK*). Estos pines admiten la comunicación *SPI* utilizando la librería *SPI* [32].

- LED: Hay un LED incorporado conectado al pin digital 13. Cuando la placa es conectada el LED está encendido, y se apaga cuando la placa se desconecta.
- TWI (Two- Wire Interface, Interfaz de dos hilos): Pin A4 *SDA* y pin A5 o *SCL*. *TWI* soporte de comunicación utilizando la biblioteca de *Wire* [32].

## Resumen de características Técnicas

**Tabla 8.** Características Técnicas Arduino uno [32].

Microcontrolador	ATmega328
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7 – 12V
Voltaje de entrada (Límite)	6 – 20V
Pines para entrada- salida digital	14 (6 pueden usarse como salida de PWM)
Pines de entrada analógica.	6
Corriente continua por pin IO	40 mA
Corriente continua en el pin 3.3V	50 mA
Memoria Flash	32 KB (0,5 KB ocupados por el <i>bootloader</i> )
<i>SRAM</i>	2 KB
<i>EEPROM</i>	1 KB
Frecuencia de reloj	16 MHz
Longitud	68.6mm
Ancho	53.4mm
Peso	25g

Fuente: Elaboración propia

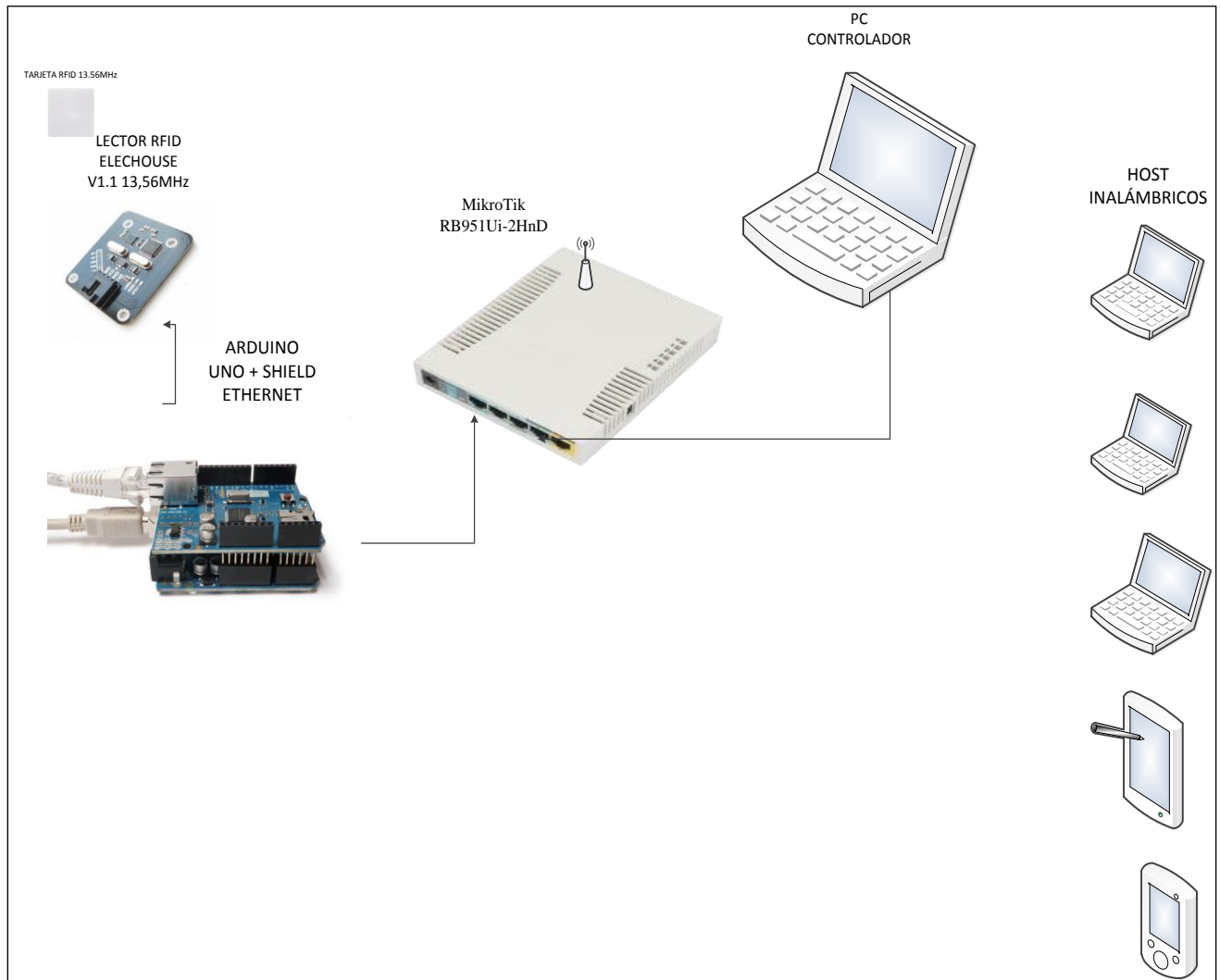
## PROCESAMIENTO

Dentro del procesamiento se usó, java, comunicación por sockets que es la comunicación tcp básica, lectura del puerto serial, en este caso todo lo que le llegaba al puerto serial del Arduino mediante un programa de java que captura la trama, es decir cada código de la tarjeta que en este caso corresponden a las lecturas serán tomados, cada tarjeta desbloquea a una máquina de forma aleatoria.

Lo que se ocupó también fue una base de datos, que se encontraba en AppServ.

### 5.1.3 Sistema SALA 2: Acceso a INTERNET mediante autenticación usando *Hotspot*, después del registro automático.

Para el presente sistema se tomó en cuenta el siguiente diseño:



**Figura 28.** Sistema Sala 2, internet inalámbrico.

**Fuente:** Elaboración propia.

Los elementos a usar son: tarjetas de identificación RFID a 13,56 MHz (usadas tanto en sala 1 y sala 2), un lector RFID de la marca Elechouse V1.1, una placa Arduino Uno, un *Shield Ethernet Arduino*, un computador donde se encontrará la base de datos y donde se realizará la configuración del equipo Mikrotik que permite crear el *hotspot*, y a esto se puede conectar la sala 2 que en este caso son portátiles *tablets* y *Smartphones*, a continuación, se detalla cada elemento que se usó:

## DESCRIPCIÓN DE MATERIALES

- **Tarjeta de Acceso de Entrada RFID 13,56 MHz de proximidad RFID**

Dicha tarjeta ya fue descrita anteriormente en el sistema de Sala 1, dado que se usará el mismo tipo de tarjetas para ambos sistemas.

- **Elechouse Módulo RFID V1.1**

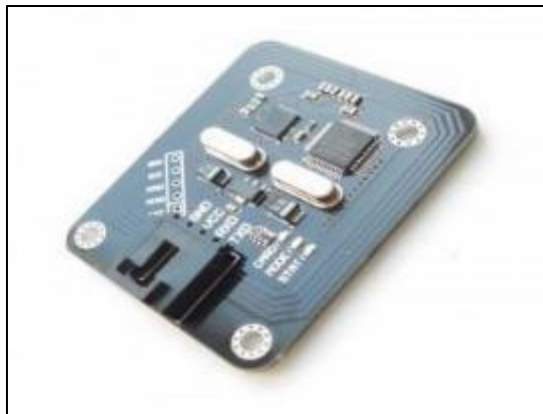
Este módulo RFID es basado en MFRC522, es un lector/escritor altamente integrado para la comunicación sin contacto a 13,56 MHz. Es compatible con el modo *ISO 14443A* / MIFARE y MIFARE Classic (por ejemplo, MIFARE estándar) los productos de comunicación sin contacto MIFARE usando transferencia más alta velocidad de hasta 848 kbit/s en ambas direcciones, como lo indica la figura 25 [33].

El funcionamiento es muy sencillo. La mayoría del trabajo es realizado por la Unidad de Control Multipunto (MCU, por sus siglas en inglés Multipoint Control Unit) Embebido. También hay una interfaz de SPI en el tablero [33].

Varias interfaces de host se implementan en este módulo:

1. Interfaz *SPI* [33].
2. *UART* serie [33].
3. Circuitería analógica para demodular y decodificar respuestas, altamente integrado [33].
4. Soporta *ISO / IEC14443A* Soporte / MIFARE
5. Soporta encriptación MIFARE Classic en modo lectura/escritura [33].
6. Soporta alta transferencia-velocidad de comunicación *ISO/IEC14443A* a 848 kbit/s [33].
7. El chip soporta algoritmo estándar de cifrado *ISO14443* y MIFARE [33].
8. Soporta comunicación *ISO/IEC14443A* con velocidad de transferencia más alta hasta 848kbit / s [33].
9. Control de interfaz *UART* serie: Velocidad de transmisión podría ser 2400bps ~ 115200 bps, con velocidad 9600bps defecto [33].

10. Los datos de configuración se conservan en la memoria *EEPROM*<sup>3</sup> contra la energía cortada [33].
11. Construido en 8K *EEPROM*, fácil de acceder a él mediante el envío de comandos [33].
12. Fácil de usar, mediante el envío de comandos simples usuarios pueden controlarlo para leer o escribir tarjetas [33].
13. Al mando compacto, comando es tan corto como un byte [33].
14. Tensión Libre: 4.5V ~ 5.5V, típicamente 5V [33].
15. Interfaz: *UART* (5V *TTL*) y *SPI* (*TTL* 3.3V) [33].
16. Dimensiones: 40 mm x 50 mm [33].
17. Velocidad de transmisión podría ser 2400bps ~ 115200bps [33].
18. *Bit* de paridad: Ninguno [33].
19. Iniciar *bits*: 1 [33].
20. *Bit* de datos: 8 [33].
21. *Bit* de parada: 1 [33].



**Figura 29.** Módulo RFID, Elechouse V1.1 [33].

- **Placa Arduino Uno**

Dicha placa ya fue descrita anteriormente en el sistema de Sala 1, dado que se usará el mismo tipo de placas para ambos sistemas.

---

<sup>3</sup> La tecnología de memoria es usualmente EEPROM (Electrically Erasable Programmable Read-Only Memory, Memoria de solo lectura programable y borrrable eléctricamente), esta permite 100.000 ciclos de escritura y la retención de los datos son de 10 años.

- ***Arduino Ethernet Shield***

La Arduino Ethernet Shield (véase la figura 26) permite a una placa Arduino para conectarse a internet. Se basa en el chip *ethernet* Wiznet W5100. El Wiznet W5100 ofrece un apilamiento de red (*IP*) capaz tanto *TCP* y *UDP*. Es compatible con hasta cuatro conexiones de *socket* simultáneas. El *Shield Ethernet* se conecta a una placa Arduino usando largos arrollamientos de hilos que se extienden a través del *Shield*. Esto mantiene la disposición de las clavijas intacta y permite que otros *Shields* puedan ser apilados en la parte superior [32].

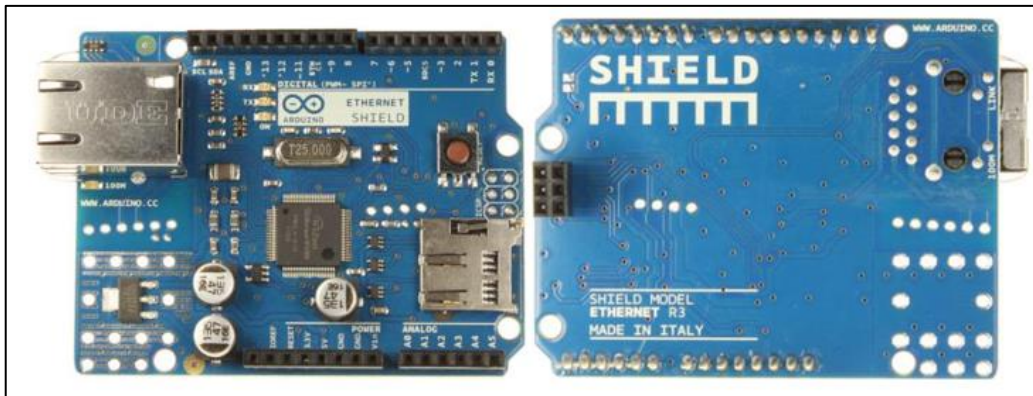
Arduino se comunica tanto con el W5100 y una tarjeta *SD* usando el bus *SPI* (a través de la cabecera *ICSP*) [32].

La *Shield de Ethernet* tiene un estándar de conexión RJ-45, con un transformador de línea integrada y alimentación a través de *Ethernet* activado [32]

Hay una ranura para tarjetas micro-*SD* a bordo, que se puede utilizar para almacenar archivos servidos sobre la red. Es compatible con todas las placas Arduino / Genuino [32].

La Shield tiene un módulo de alimentación a través de *Ethernet (PoE)*, diseñado para extraer energía de un cable Ethernet convencional de par trenzado Categoría 5 [32]:

- Compatible con el estándar IEEE802.3af [32].
- Baja producción rizados y ruido (100mVpp) [32].
- Entrada rango de voltaje de 36V a 57V [32].
- Protección de sobrecargas y cortocircuito [32].
- Salida 9V [32].
- Alta eficiencia del convertidor DC / DC: típico 75% @ 50% carga [32].
- Aislamiento 1500V (entrada a la salida) [32].



**Figura 30.** *Shield Ethernet* de Arduino [32].

- **MIKROTIK RB951Ui-2HnD**

Todo equipo mikrotik tiene como sistema operativo RouterOS basado en Linux, este nunca caduca, puede ser actualizado durante uno a 3 años, tiene un uso ilimitado de interfaces y cada licencia es solamente para una PC [34].

Los equipos de MikroTik ya vienen con la licencia incluida, la licencia que tiene el dispositivo que se escogió es de nivel 4, existiendo en el mercado hasta el nivel 6, esta licencia cuenta con las siguientes características [34].

Tiempo de Actualización.....	1 año [34]
<i>Wireless AP</i> .....	Sí
<i>Wireless Cliente y Bridge</i> .....	Sí
<i>RIP, OSPF, BGP</i> protocolos.....	Sí (2.10 = no)
Túneles <i>EoIP</i> .....	ilimitados
Túneles <i>PPPoE</i> .....	200
Túneles <i>PPTP</i> .....	200
Túneles <i>L2TP</i> .....	200
Interfaces <i>VLAN</i> .....	ilimitadas
Reglas <i>P2P firewall</i> .....	ilimitadas
Reglas <i>NAT</i> .....	ilimitadas
Usuarios activos <i>HotSpot</i> .....	200

RADIUS cliente.....	Sí
Colas.....	ilimitadas
Web proxy.....	Sí
Interfaces Síncronos.....	Sí
Sesiones activas de gestión.....	10

Por tanto, todo equipo que tenga licencia 4, cuenta con hotspot para 200 usuarios, y cumple con las necesidades que se requieren [34].

Entonces para la elección adecuada del dispositivo, tenemos que fijarnos que tenga licencia de nivel 4 y en algunas especificaciones técnicas como por ejemplo capacidad de procesamiento, memoria RAM, alimentación, etc.; el dispositivo que cumple con estas características corresponde a un dispositivo *SOHO* (small office / home office), es decir es apto para hogares y pequeñas oficinas [34].

El RB951Ui-2HnD (véase figura 27) *SOHO* es *AP* inalámbrico con antenas internas, con *CPU* Atheros de nueva generación, presenta una alta potencia de procesamiento. Tiene cinco puertos *Ethernet*, un puerto *USB* 2.0 y una potencia de salida en 2.4GHz de hasta 1000mW. Cumple el estándar 802.11b/g/n [34].



**Figura 31.** Mikrotik RB951Ui-2HnD [35].

Cuenta con una CPU de 600 MHz, 128 MB de RAM, la carga máxima en el puerto es de 500mA, en la Tabla 2 se describen otras características técnicas del dispositivo [34].



**Tabla 9.** Especificaciones técnicas del dispositivo RB951Ui-2HND [34].

Especificaciones RB951Ui-2HND	
CPU	600MHz
CPU núcleos	1
RAM	128
Puertos LAN	5
Gigabit	No
MiniPCI	No
Wireless integrado	Si
Estándares	802.11b/g/n
USB	1
Power Jack	Si
Soporta 802.3af	No
PoE	8-30V DC en Ether1
Voltage Monitor	No
PCB temperature monitor	No
CPU temperature monitor	No
Dimensiones	113x138x29mm
Sistema Operativo	RouterOS
Rango Temperatura	-20C +50C
RouterOS Licencia	Nivel 4
Ganancia de antena	2 antenas de 2.5dBi
Monitor corriente	No
Potencia Máx. TX	30dBm
Max Consumo potencia	7W
Puertos SFP	0

**Fuente:** Elaboración propia.

## 5.2 Diseño del software.

La primera parte de almacenar información que vienen de las etiquetas RFID en una base de datos, se hizo utilizando una herramienta de *Open Source* llamada AppServ 2.5.7, herramienta para el sistema operativo Windows, que tiene incluido Apache, MySQL, PHP, phpMyAdmin, dicho paquete debe ser instalado en el computador principal de la biblioteca.

Existen otras herramientas que igual incluyen MySQL, PHP, Apache, etc., que se las puede utilizar para el sistema operativo Windows, como lo son WAMP y Xampp, sin embargo, AppServ es muy sencillo en cuanto a la instalación y configuración.

### 5.2.1 Sala I, desbloqueo de máquinas de mesa después del registro.

Para lograr desbloquear las máquinas se requiere realizar el registro con el código que permite la lectura de las tarjetas en Arduino (véase ANEXO 8), este código simplemente representa una lectura simple de la trama en la tarjeta RFID.

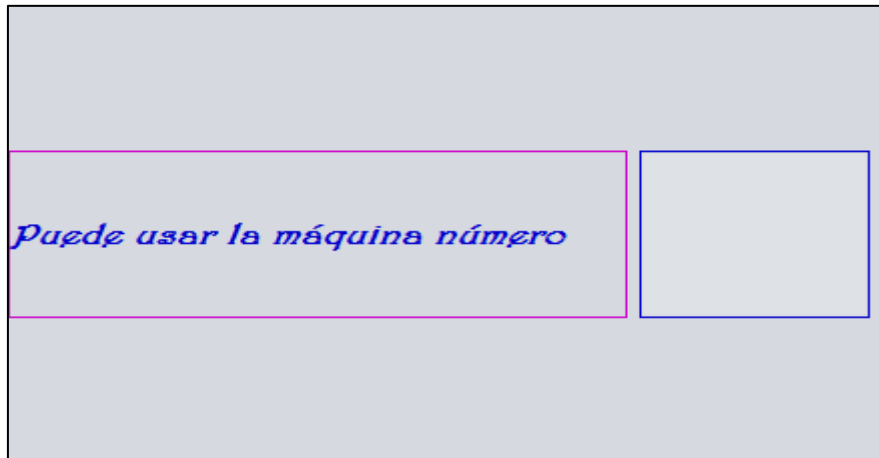
Luego se tiene el proyecto en java, que para su funcionamiento se necesita que las máquinas de mesa estén en una misma red con *ip* estática.

Los dos proyectos que se realizaron para hacer funcionar el sistema de bloqueo y desbloqueo de las máquinas de mesa fueron:

- Biblioteca.
- Hilos.

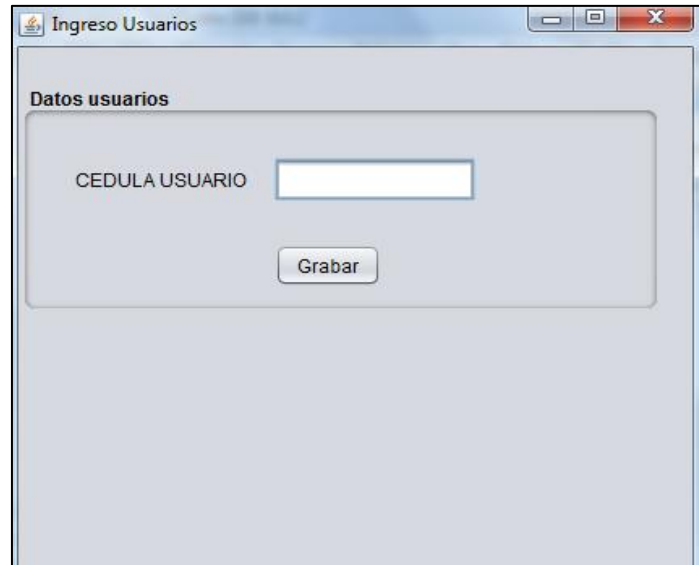
**BIBLIOTECA:** Dentro del proyecto biblioteca.java realizado en *Netbeans*, se encuentran 5 clases, el paquete principal corresponde a la clase *ingreseusuarios1.java*

- ✓ **JFrameGUI.java:** Es una estructura gráfica, que indica el número de máquina que ha sido desbloqueada para su respectivo uso (véase la figura 28).



**Figura 32.** Interfaz gráfica indicativa JFrameGui.

- ✓ **Serial.java:** Toma los datos que están llegando del puerto USB, es una forma de interacción de java con el puerto serial, luego se envía esa trama de la tarjeta a la base de datos, para ser guardados (véase ANEXO 1).
- ✓ **conector.java:** Es usador por el serial.java, para realizar la conexión a la base de datos, aquí se ingresa la ubicación del servidor donde se encuentra la base de datos, el usuario y contraseña (véase ANEXO 2).
- ✓ **IpConexion.java:** Verifica las *IPs* que están en la clase principal ingreseusuarios1.java, toma las ips de esa clase, las devuelve a la clase principal y comprueba su estado, es decir si están en uso o no.
- ✓ **ingresousuarios1.java:** Este es el programa principal (véase ANEXO 3), donde se encuentra el diseño de la interfaz del usuario (véase la figura 29), también se coloca la lista de las *IPs*, que tendrán las computadoras de mesa asignadas, esta lista será ocupada por la clase IpConexion.java; así como el número de puerto serial del Arduino, que le sirve a la clase serial.java para saber por qué puerto van a ser tomados los datos.



**Figura 33.** Interfaz gráfica para el registro, usando el número de cédula.

Como el diseño del paquete `ingreseusuarios1.java`, tiene opción a colocar el número de cédula, utiliza ese número y lo compara con el que está en la tabla de la base de datos, luego si existiese ese número, realiza nuevamente la conexión a la base de datos del servidor para guardar el registro, en caso de que el usuario haya olvidado su identificador RFID y no se ocupe la clase `Serial.java`.

La clase envía a la interfaz gráfica `JFrameGUI.java` el número de máquina desbloqueada, para informar a los usuarios, la computadora que se le ha asignado.

Se utiliza la conexión mediante sockets para el desbloqueo, cuando se tenga el registro, se envía una trama “desbloquear” que al ser recibida por el programa que bloquea la máquina `hilos.java` (del que se describirá posteriormente), desbloquea la máquina.

Las librerías principales que el proyecto uso fueron:

- `java 1.8.0.92 32 bits`: la librería principal de java.
- `Mysql-connector-java-5.1.23-bin.jar`: Creado y usada para la conexión con la base de datos.
- `javaSerial -comm.jar`: Librería que fue creada, la misma ocupa tres archivos:
  - `comm.jar`: Colocada en las siguientes direcciones:

`%JAVA_HOME%/lib`  
`%JAVA_HOME%/jre/lib/ext`

- win32comm.dll: Colocada en las siguientes direcciones:  
     %JAVA\_HOME%/bin  
     %JAVA\_HOME%/jre/bin  
     C:/windows/system32/
- javax.comm.properties: Colocada en las siguientes direcciones:  
     %JAVA\_HOME%/lib  
     %JAVA\_HOME%/jre/lib

Donde %JAVA\_HOME, es el directorio de instalación del jdk de java.

## HILOS

Al proyecto se convierte en extensión .jar, para que, al prenderse las máquinas de préstamo de la biblioteca, se ejecute.

El proyecto ocupa las siguientes clases:

- ✓ **JFrameBlocked.java:** Corresponde al diseño de la interfaz de bloqueo (véase la figura 30)



**Figura 34.** Interfaz gráfica de la pantalla de bloqueo.

- ✓ **JFrameGUI.java:** Es el diseño de la interfaz de desbloqueo, aparece en la máquina del cliente (véase la figura 31)



**Figura 35.** Interfaz gráfica al desbloquear la pantalla.

- ✓ **Modelo.java:** En esta clase se coloca el tiempo de desbloqueo de la máquina, al cumplirse dicho periodo, llama al JFrame de bloqueo, y esta vuelve al estado de espera hasta que se desbloquee nuevamente llamando al JFrame de desbloqueo al realizar el registro (véase ANEXO 4).
- ✓ **Servidor.java:** Es el programa que, mediante sockets, espera la trama “desbloquear”, para proceder a habilitar la máquina (véase ANEXO 5).
- ✓ **jBlocked.java:** Mantiene al JFrameBlocked (Frame de bloqueo), siempre al frente, sobreponiéndolo en la pantalla cada medio segundo y repitiéndolo N número de veces (véase ANEXO 6).
- ✓ **Test.java:** Es la clase principal donde se ejecutan las clases Servidor.java y Modelo.java (véase ANEXO 7).

La librería principal que el proyecto uso fue:

- JDK 1.8: librería principal de java que viene por defecto.

Para realizar la acción de desbloquear una máquina de escritorio se utiliza la estructura cliente-servidor y programación en java, trabajando con sockets en java versión 1.8.0.40 con el IDE *Netbeans* 8.0.2, donde se realizará la programación para lograr obtener el script del cliente que tiene un socket que responde a un puerto específico, mientras que el servidor va a estar constantemente escuchando y esperando a través del socket a que el cliente realice una petición. Por tanto, el servidor siempre debe estar activado.

En cuanto a la creación de la script del cliente se debe conocer el número del puerto en el que el servidor se encuentra ejecutándose y el nombre del host en el que dicho servidor está instalado.

### **5.2.2 Sala II, préstamo de espacio para utilizar wifi en computadoras personales o PDAs.**

La segunda parte del presente proyecto consiste en activar un usuario del *hotspot* al pasar la tarjeta RFID por el lector, logrando que únicamente la persona que se registró al ingresar a una página web pueda autenticarse, con usuario y contraseña correspondientes al código de la tarjeta, parámetros que pueden ser modificables, así mismo como se puede cambiar el tiempo de conexión a internet, en el proyecto se implementa un programa que corre al momento que se enciende la máquina principal (servidor) o donde se realiza el procesamiento, este programa ingresa a la configuración de mikrotik y desactiva todos los usuarios, para que cada día nuevamente se vuelvan a registrar, si el cliente no se registra el usuario en el equipo mikrotik estará en estado desactivado, por tanto es necesario el registro.

Se usó el software Arduino, Winbox, *Sublime Text*, para lograr la comunicación entre el equipo mikrotik y la placa Arduino con su *Shield Ethernet* fue el uso de *php* y la *API* (Interfaz de programación de aplicaciones) de mikrotik.

El API de Mikrotik no es una terminal SSH (Interprete de ordenes seguro, por sus siglas en español) solo es un Server Socket esperando conexiones por cierto puerto (8798 por defecto) y en el que se reciben las instrucciones, para usar la api se la debe activar entrando al router mikrotik.

Los programas que se usaron en Sala 2, fueron:

- ✓ **RFID sala 2**, en el software Arduino (Véase ANEXO 9).

El programa en Arduino pasa los datos seriales de la lectura, al software reportes2.php, donde se procede a almacenar en la base de datos.

- ✓ **Reportes2.php**, en el software Sublime Text (Véase ANEXO 10).

El archivo reportes2.php es el lazo que une tanto a la base de datos como al equipo mikrotik, mediante la *API* de mikrotik.

La API o Interfaz de programación de aplicaciones sirve para controlar funciones del Mikrotik desde programas externos, como en este caso se activarán usuarios, sin la necesidad de abrir el winbox y modificar manualmente, esto permite que todo se realice de forma automática.

Este archivo principal va desbloqueando cada usuario registrado, se programó en *php*, pudiendo haberlo realizado en java, en c, c++, perl, python, etc., el resultado es indiferente al software de programación que se utilice.

Para la comunicación entre este software y Mikrotik se requiere una librería, que está al principio del código (véase anexo 11); lo primero que se realiza es la toma de datos como usuario, contraseña, nombre de la base de datos para su posterior conexión, de igual manera la información que permite el ingreso al Winbox de Mikrotik, e ir activando cada usuario de la tarjeta que pase por el lector RFID, en este programa también se relacionan los campos de la tabla de registro correspondiente a la base de datos.

- ✓ **Activar2.php**, en el software Sublime Text (Véase ANEXO 11).

Su función es desactivar todos los usuarios anteriormente activados, para comenzar el día de registro con una tabla de usuarios desactivados.

Este archivo.php de igual forma usa la API de mikrotik, ingresando al winbox del mikrotik y haciendo correr un script denominado creando (Véase ANEXO 12), el que desactiva todos los usuarios que se encontraban en estado activo.

Para lograr que ese software se ejecute de forma automática se creó un programa que llama al archivo activar2.php y este a su vez a un script q se encuentra dentro de mikrotik, se utilizó la siguiente línea, en un archivo ejecutable.bat, que se colocó en la carpeta inicio para que cada vez que se prenda la máquina principal los usuarios empiecen todos como desactivados:

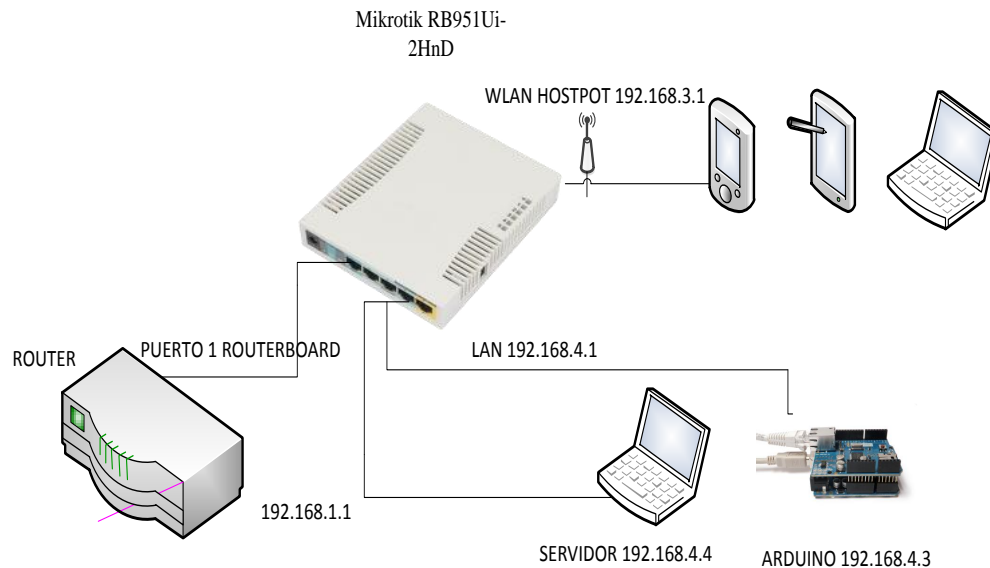
```
C:\AppServ\php5\php.exe -f C:\AppServ\www\test4\activar2.php
```



### 5.2.2.1 Configuración de *hotspot* en Mikrotik.

Para la configuración del *hotspot*, se realizaron los siguientes pasos:

1. Reseteamos el routerboard.
2. Configuramos la entrada de internet (WAN) y LAN (véase la figura 32).



**Figura 36.** Conexiones WAN, LAN Y WLAN al routerboard.

**Fuente:** Elaboración propia.

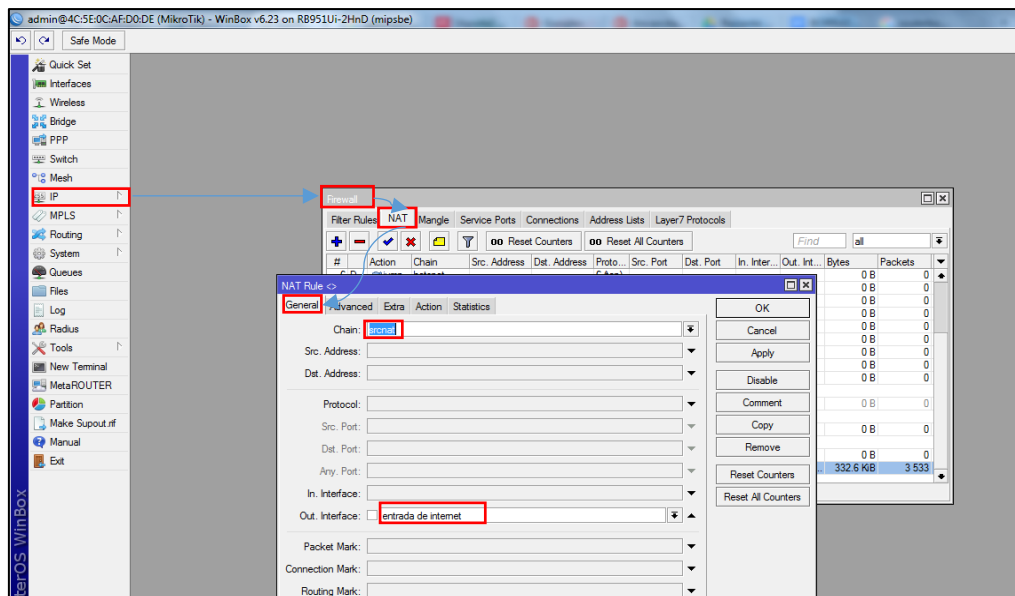
Para que sea posible el acceso a internet, se deberá conectar el puerto uno del routerboard Mikrotik a un puerto del router de la biblioteca, por tanto, se configuran esas interfaces, así como la red LAN que sirve para conectar la máquina principal y la Shield Ethernet Arduino.

Para agregar las direcciones, se accede a *IP -> ADDRESSES*, quedando como se muestra en la figura 33.

	Address	Network	Interface
D	192.168.1.7/26	192.168.1.0	entrada de inter...
	192.168.3.1/24	192.168.3.0	wlan1
	192.168.4.1/24	192.168.4.0	LAN

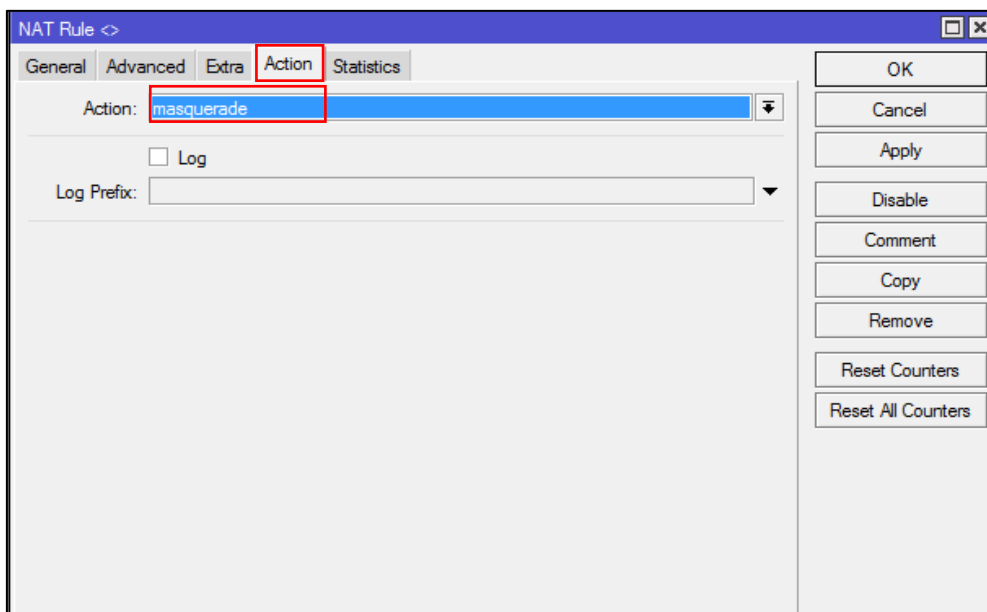
**Figura 37.** Lista de direcciones IP, para las interfaces de red.

Luego se procede a enmascarar, *IP* -> Firewall -> Pestaña *Nat* -> General, En chain se selecciona *srcnet*, y en *out* interface la *wan* que en este caso vendría a ser la entrada de internet (véase la figura 34), luego se aplican los cambios.



**Figura 38.** Enmascaramiento de las IP's parte 1.

En la misma ventana se pulsa en *Action* y se selecciona *masquerade*, como se indica la figura 35.



**Figura 39.** Enmascaramiento de las IP's parte 2.

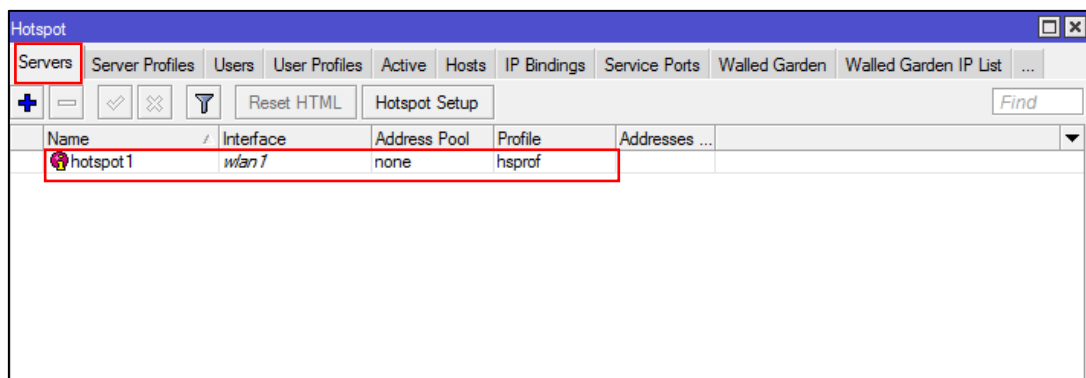
La acción enmascarada, aísla la red *LAN*, al enmascarar las conexiones a través de la entrada de internet.

Después se debe hacer un ruteo al Gateway disponible, se dirige a *IP -> Routes*.

	Dist. Address	Gateway	Distance	Routing Mark	Pref. Source
DAS	0.0.0.0/0	192.168.1.1 reachable entrada de internet	0		
DAC	192.168.1.0/26	entrada de internet reachable	0		192.168.1.7
DC	192.168.3.0/24	wlan1 unreachable	255		192.168.3.1
DAC	192.168.4.0/24	LAN reachable	0		192.168.4.1

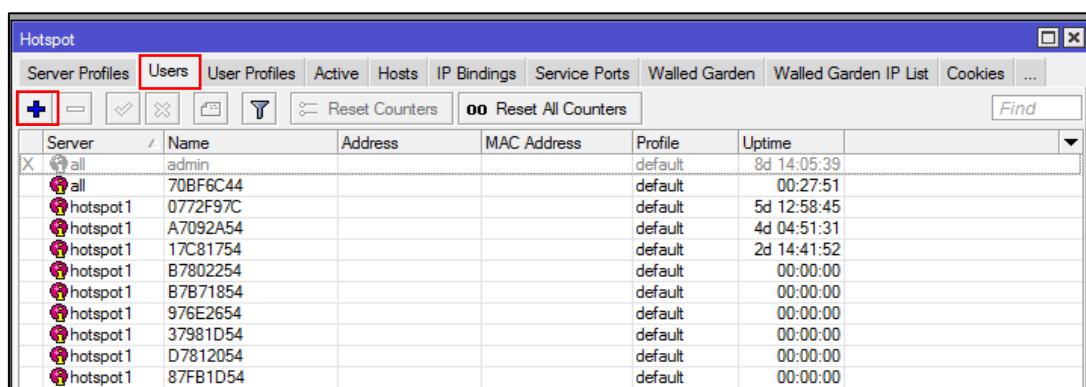
**Figura 40.** Lista de rutas y su puerta de enlace.

Para realizar la configuración final, correspondiente al *hotspot*, en *IP -> hotspot ->* en la pestaña *servers ->* se presiona *hotspot Setup* y con ayuda del asistente de configuración automático se finaliza la creación del *hotspot*, quedando como se indica en la figura 37.



**Figura 41.** Creación del *hotspot* en Mikrotik.

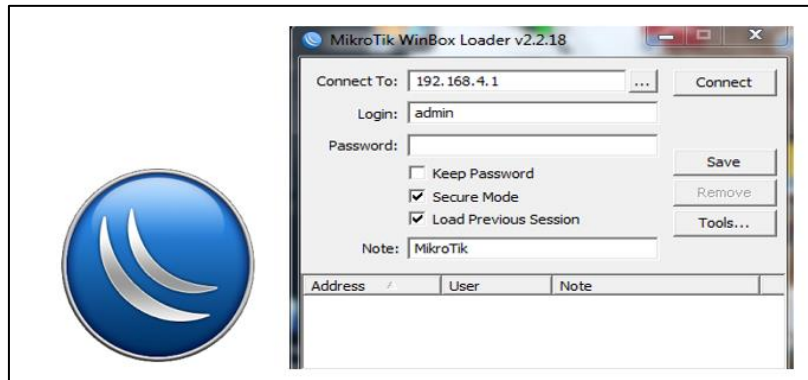
Luego de realizar el *hotspot* se procede a crear los usuarios (véase la figura 38)



**Figura 42.** Usuarios creados para *Hotspot*.

Por último, se modificó la página de autenticación del hotspot, se realizó un nuevo diseño, en el que se ocupó imágenes *GIF*, e indicaciones acerca del registro (véase la figura 39).





**Figura 44.** Winbox

### 5.2.3.2 Arduino

Software que permite la escritura de los programas para su respectiva verificación (véase la figura 41), compilación y quemado para las diferentes placas de Arduino; contiene a su vez librerías para programación de las placas Arduino, y de otras placas adicionales. Contiene también proyectos y ejemplos que sirven de guía.



**Figura 45.** Arduino

### 5.2.3.3 Netbeans

Es un entorno de desarrollo para el lenguaje de programación Java, de código abierto, que permite la creación, modificación, compilación, ejecución entre otras cosas, de los distintos programas que puedan ser desarrollados en este lenguaje. Cuenta a su vez con distintas librerías para este fin, además la creación de interfaces gráficas programables (véase la figura 42).



**Figura 46.** Netbeans

#### 5.2.3.4 Sublime Text

Software de código libre, es un editor de texto y editor de código fuente que permite la creación y/o modificación programas multiplataforma dado que soporta un gran número de lenguajes de programación (véase la figura 43).



**Figura 47.** Sublime text.

## 5.3 Diseño del hardware

### 5.3.1 Diseño del lector RFID para Sala 1

La conexión del Arduino Uno al módulo RFID se muestra en la Tabla 10 cuyo diseño de la placa adaptativa se realizó en Proteus (Ares), se puede observar en el Anexo 13(b).

**Tabla 10.** Conexión del módulo RFID V3 con la placa Arduino Uno, sala 1

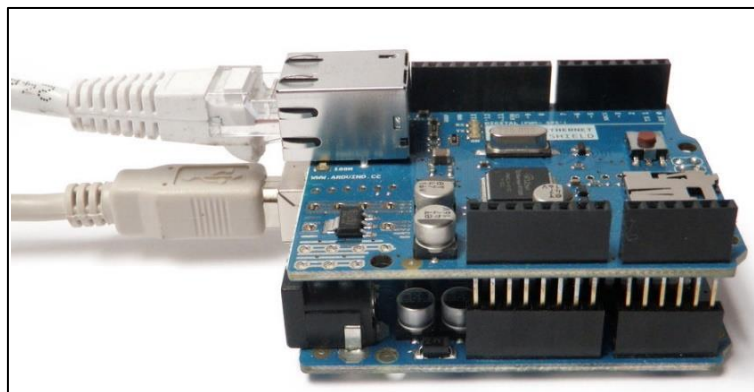
ARDUINO UNO	MÓDULO RFID V3
GND	GND
5V	VCC
SCK (Serial clock)	PIN 13
MISO (Master In Slave Out)	PIN 12
MOSI (Master Out Slave In)	PIN 11
SS (Slave Select)	PIN 10

Fuente: Elaboración propia

Para el lector de la sala 2 se ocupó la comunicación ISP, por eso es la razón de conexión de sus puertos, el módulo RFID v3, necesita dos puertos seriales para hacerlo por comunicación serial, si se ocupara la misma versión de lector de la Sala 2, sería posible realizarlo con la misma comunicación.

### 5.3.2 Diseño del lector RFID para Sala 2

Primero se monta el Shield Ethernet en la placa Arduino Uno (véase figura 44).



**Figura 48.** Arduino Uno más Shield Ethernet [32].



Luego se debe sobreponer la placa realizada en el simulador Proteus (Ares), que se muestra en el ANEXO 14 (b), que es una placa adaptativa para el módulo lector RFID V1.1, en la Tabla 12, se indica la conexión.

**Tabla 11.** Conexión del módulo RFID V1.1 con la placa Arduino uno, sala 2

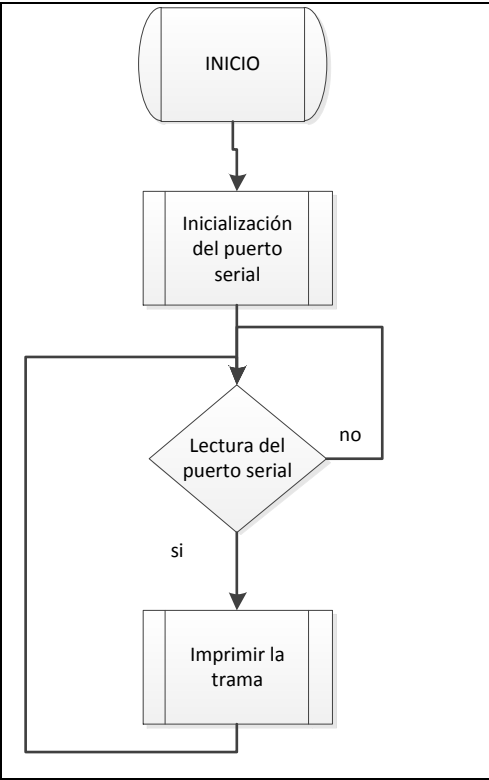
SHIELD ETHERNET +ARDUINO UNO	MÓDULO RFID
GND	GND
5V	VCC
Pin 6	TXD
Pin 7	RXD

Fuente: Elaboración propia.

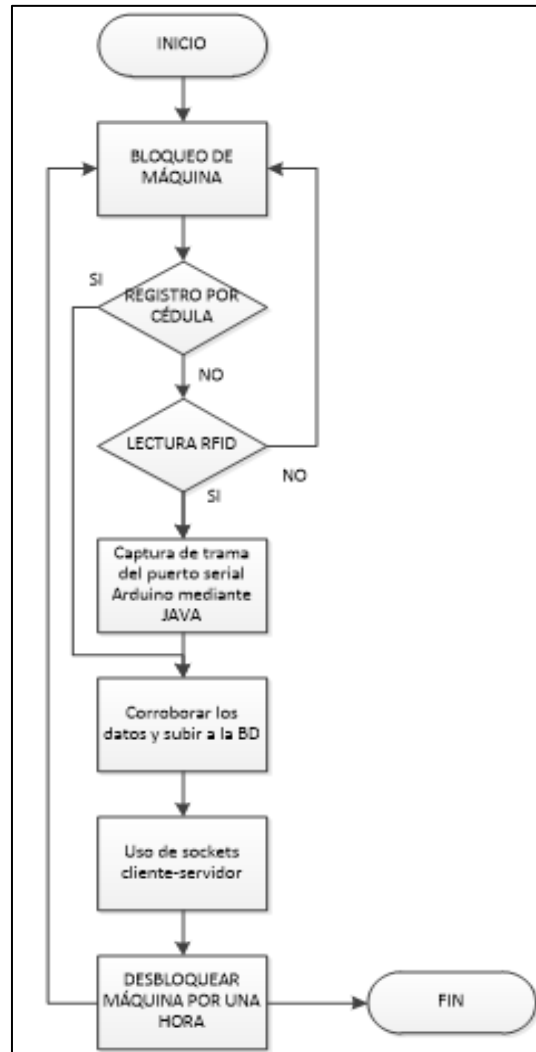
El diseño de la caja del prototipo de la Sala 1 y Sala 2, se hizo en sketchup, herramienta para el diseño gráfico en 3D.

### 5.4 Diagramas de flujo

#### 5.4.1 Diagrama Sala 1, software Arduino.

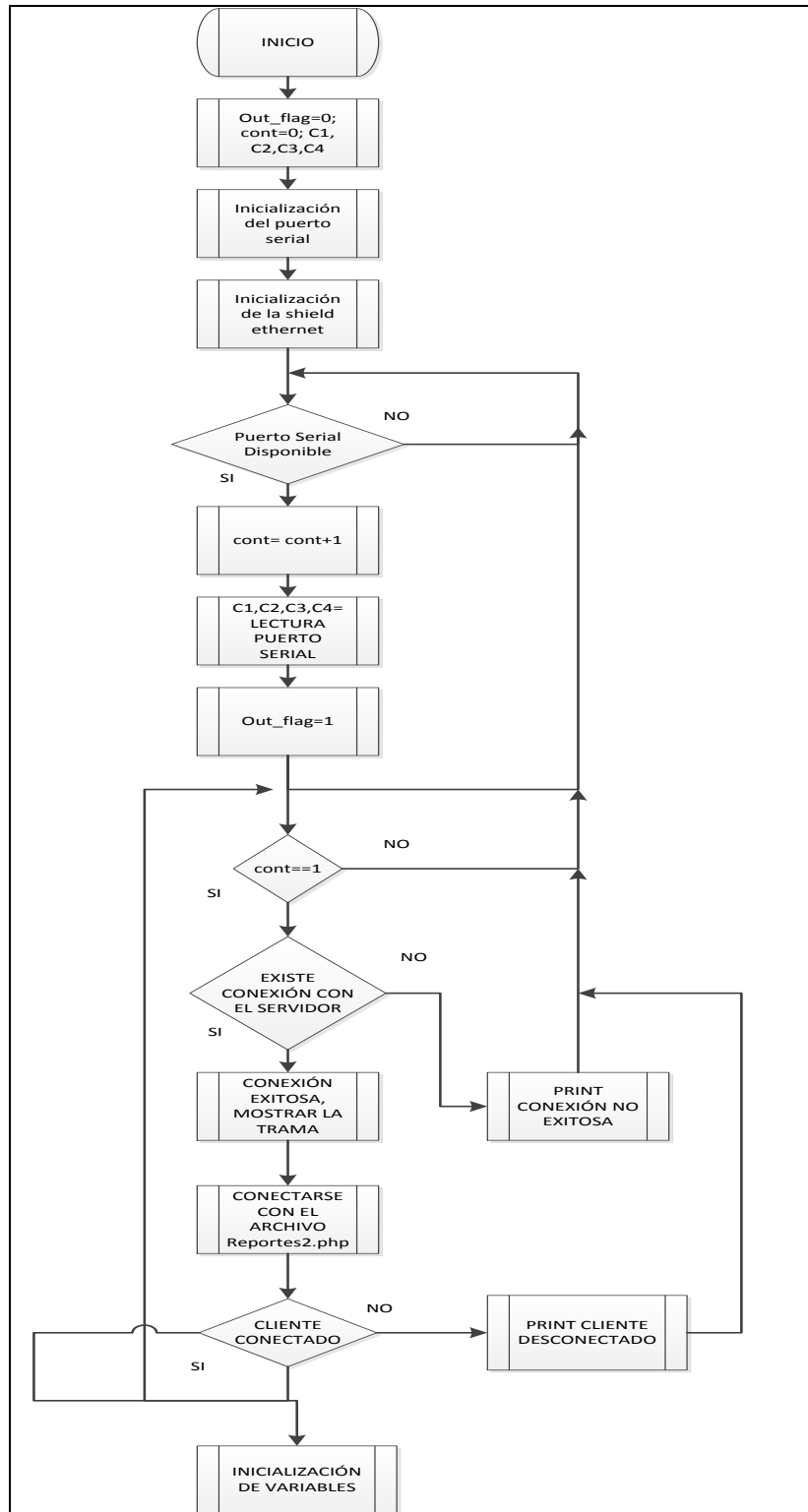


**Figura 49.** Diagrama de flujo del programa en Arduino, para la lectura serial sala 1.

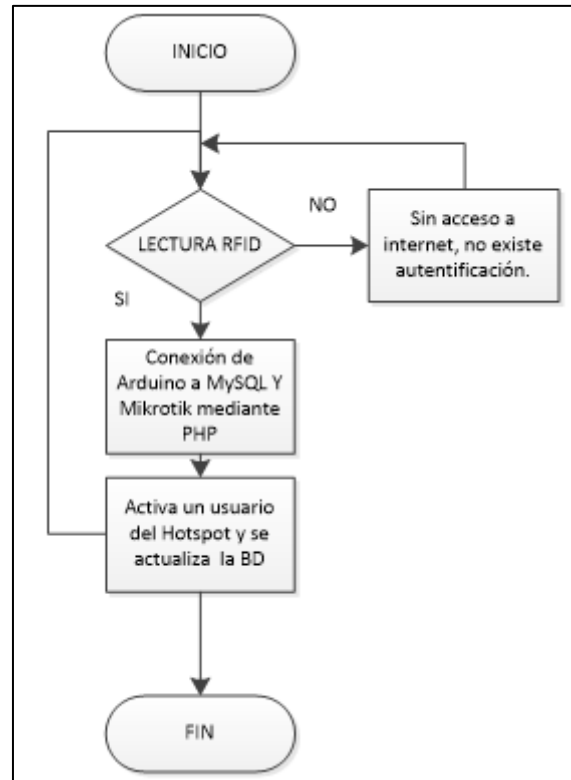


**Figura 50.** Diagrama general de flujo de la sala 1.

### 5.4.2 Diagrama Sala 2, software Arduino



**Figura 51.** Diagrama de flujo del programa en Arduino, para la lectura serial sala 2



**Figura 52.** Diagrama general de flujo de la sala 2

## 6. RESULTADOS

A continuación, se muestra los resultados que se obtuvieron, para una pequeña muestra de usuarios, se indicará la tabla de registro para las dos salas y sus interfaces antes y después del registro.

### 6.1 Prototipo Sala 1

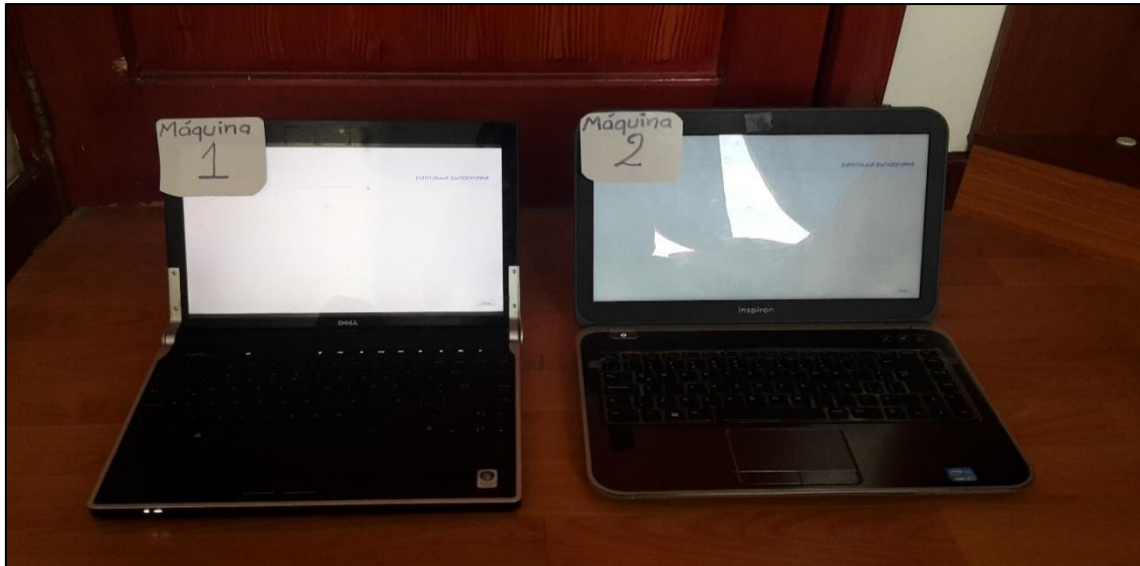
#### 6.1.1 Tabla de registro “Biblioteca”

id_registro	id_persona	persona	carrera	módulo	máquina	código_cédula	FECHA
908	8	Cedillo Benavides Lennin Stalin	CIEYT	IX	1	37981D54	2016-11-09 00:52:28
909	1	Andreina Veronica Chuncho Roman	CIEYT	E	1	1105681710	2016-11-09 00:53:03
910	8	Cedillo Benavides Lennin Stalin	CIEYT	IX	1	37981D54	2016-11-09 00:53:26
911	8	Cedillo Benavides Lennin Stalin	CIEYT	IX	1	37981D54	2016-11-09 00:53:47
912	1	Andreina Veronica Chuncho Roman	CIEYT	E	1	1105681710	2016-11-09 00:54:35
913	1	Andreina Veronica Chuncho Roman	CIEYT	E	1	1105681710	2016-11-09 00:56:38
914	8	Cedillo Benavides Lennin Stalin	CIEYT	IX	1	37981D54	2016-11-09 00:57:56
915	3	Vicente Paul Quezada Patiño	Administ	AD	1	A7092A54	2016-11-09 13:41:31
916	1	Andreina Veronica Chuncho Roman	CIEYT	E	1	1105681710	2016-11-09 13:53:50
917	7	Delgado Tapia Jeffry Vicente	CIEM	VI	1	976E2654	2016-11-09 14:43:26

**Figura 53.** Tabla de registro de los usuarios para el uso de las computadoras de escritorio.

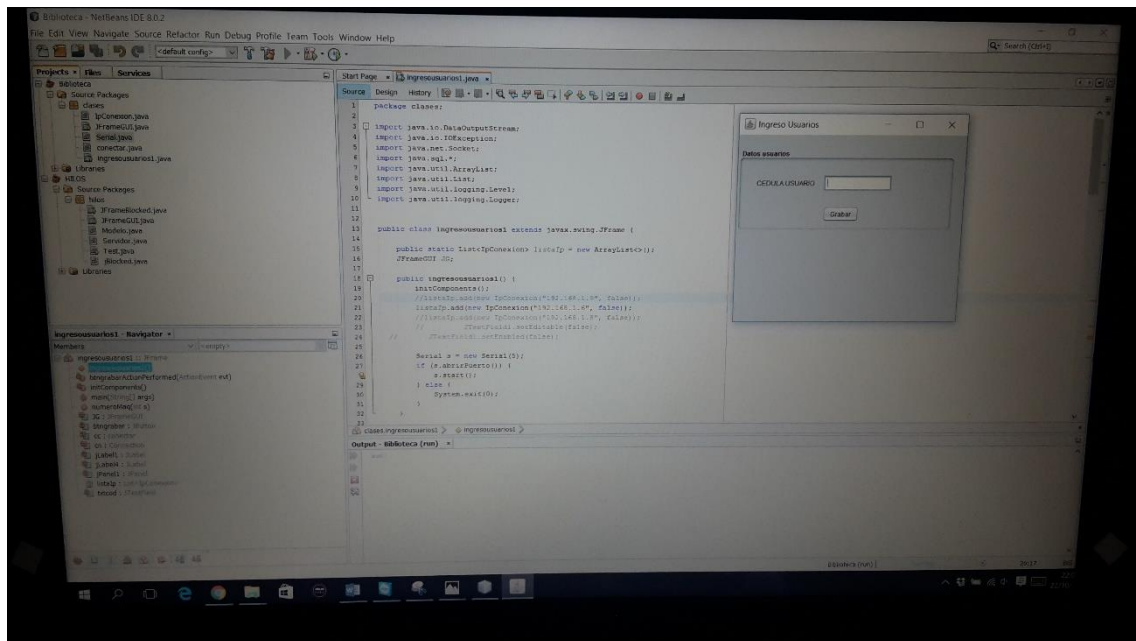
#### 6.1.2 Interfaces de bloqueo y desbloqueo.

En la figura 50 podemos observar que dos máquinas muestran una interfaz de bloqueo.



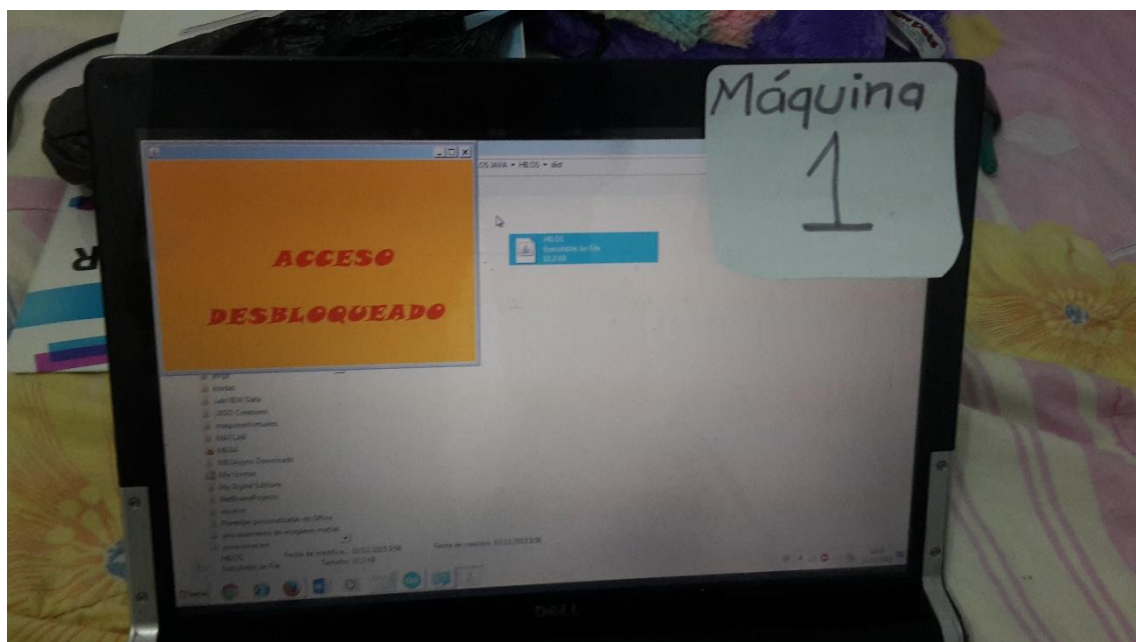
**Figura 54.** Computadoras bloqueadas, sala 1.

En la figura 51 se tiene la máquina principal, que da la opción de ingresar el número de cédula, o pasar el identificador RFID.



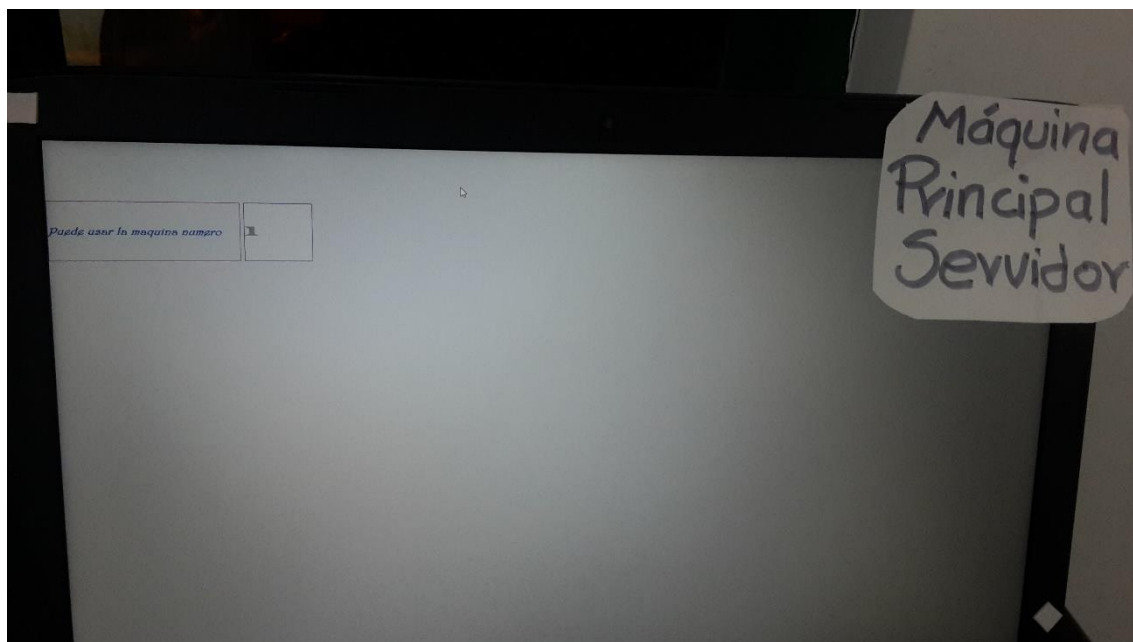
**Figura 55.** Máquina servidor principal.

Al momento de ingresar el número de cédula o pasar el identificador, se tiene como resultado la figura 52.



**Figura 56.** Máquina desbloqueada, después del registro.

Y en la máquina servidor se presenta la siguiente visualización mostrada en la figura 53.



**Figura 57.** Máquina principal al pasar el identificador RFID por el lector.

## 6.2 Prototipo Sala 2

### 6.2.1 Tabla de registro “registro\_wifi”

Al pasar la tarjeta por el lector RFID, la información se almacena en la tabla de la base de datos (véase la figura 54), de igual manera activa los usuarios *hotspot* como se indica la figura 55.

id_registro	persona	carrera	modulo	fecha	id_persona
181	Cedillo Benavides Lennin Stalin	CIEYT	IX	2016-10-05 02:34:16	8
182	Andreina Veronica Chuncho Roman	CIEYT	E	2016-10-05 02:34:21	1
183	Guillermo Xavier Palacios Arias	CIEM	IX	2016-10-05 02:34:25	2
184	Ernesto Vladimir Palacios Merino	Administ	AD	2016-10-05 02:34:29	4
185	Masaco Pinta Angel Ubaldo	CIEYT	IX	2016-10-05 02:34:31	5

**Figura 58.** Registro de los usuarios para el uso de la Sala 2.

The screenshot shows a web-based interface for managing a hotspot. The title bar is 'Hotspot'. There are several tabs: 'Server Profiles', 'Users', 'User Profiles', 'Active', 'Hosts', 'IP Bindings', 'Service Ports', and 'Walled Garden'. The 'Users' tab is selected. Below the tabs are several icons and buttons: a plus sign, a minus sign, a checkmark, an 'X', a folder icon, a funnel icon, '00 Reset Counters', '00 Reset All Counters', and a 'Find' search box. The main area is a table with the following columns: 'Server', 'Name', 'Address', 'MAC Address', 'Profile', and 'Uptime'. The table contains 11 rows of data. The first row is 'all' with name 'admin' and uptime '8d 14:05:39'. The next two rows are 'all' with MAC addresses '70BF6C44' and '0772F97C' and uptime '00:27:51' and '5d 12:58:45' respectively. The following seven rows are 'hotspot1' with various MAC addresses and uptime values, including '00:00:00'. The last row is 'hotspot1' with name '1' and uptime '00:02:24'.

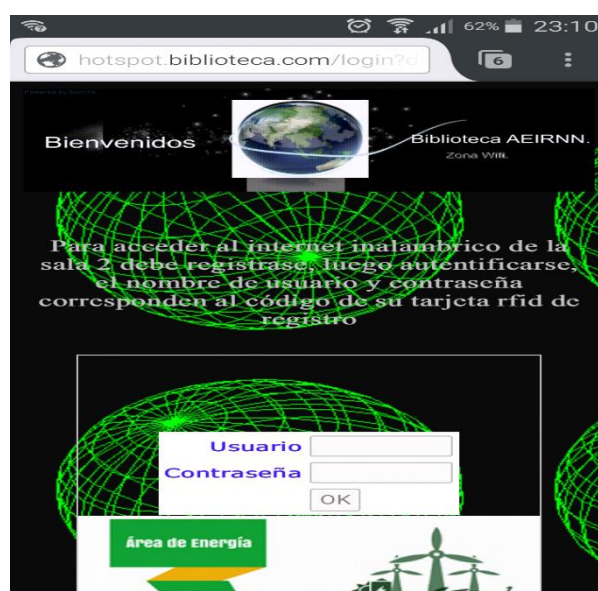
Server	Name	Address	MAC Address	Profile	Uptime
X all	admin			default	8d 14:05:39
X all	70BF6C44			default	00:27:51
X hotspot1	0772F97C			default	5d 12:58:45
X hotspot1	A7092A54			default	4d 04:51:31
X hotspot1	17C81754			default	2d 14:41:52
X hotspot1	B7802254			default	00:00:00
X hotspot1	B7B71854			default	00:00:00
X hotspot1	976E2654			default	00:00:00
X hotspot1	37981D54			default	00:00:00
X hotspot1	D7812054			default	00:00:00
X hotspot1	87FB1D54			default	00:00:00
X hotspot1	1			default	00:02:24

**Figura 59.** Usuarios activos *Hotspot*.

Se puede observar que los usuarios activos, corresponden a los ingresados en la tabla de registro, mostrada en la figura 54.

## 6.2.2 Pantalla de autenticación

Al realizarse o no el registro el usuario tendrá en la pantalla de navegación la interfaz mostrada en la figura 56, lo único diferente es que se le activará su nombre de usuario y contraseña para que pueda navegar.



**Figura 60.** Página de autenticación *Hotspot*, sala 2.



### 6.3 Costos

El costo de los materiales usados, en los prototipos de sala 1 y sala 2, se lo indica en la siguiente tabla:

**Tabla 12** Precio de los materiales usados.

<b>CANTIDAD</b>	<b>Material</b>	<b>Costo</b>
1	Equipo MikroTik RB951UI-2HND	\$131.99
2	Placa Arduino uno	\$76.00
1	Arduino Ethernet Shield	\$15.00
1	Módulo RFID elechouse V1.1	\$21.90
1	Módulo RFID elechouse V3	\$21.90
10	Tarjetas RFID a 13,56 MHz	\$20.00
<b>TOTAL</b>		<b>\$264.89</b>

Un sistema RFID ofrece autonomía al usuario, como podemos observar la tabla 12 nos muestra la inversión para realizar los lectores, el costo total es bajo, totalmente accesible, esta inversión se incrementará en la implementación del prototipo al requerirse una mayor cantidad de tarjetas RFID, las mismas tienen una vida prácticamente ilimitada, por tanto, el usuario adquirirá otra solamente en caso de pérdida o por algún daño físico que afecte a la antena o chip.

## 7. DISCUSIÓN

El presente trabajo tiene como propósito específico presentar una alternativa al registro manual que actualmente existe en la biblioteca del AEIRNNR, mediante prototipos de registro automático utilizando la tecnología RFID, que serán probados para una muestra de 10 usuarios, además de almacenar los registros de la sala 1 y sala 2 en una base de datos, el prototipo desbloquea máquinas de mesa (sala 1), y permite la navegación por autenticación que otorga *Hotspot* (Sala 2), después de pasar un identificador por el lector RFID.

- **DESARROLLO DEL PROTOTIPO**

**Objetivo 1: Diseñar un sistema RFID utilizando Arduino UNO, módulo RFID.**

Para el cumplimiento de este objetivo, se toma en cuenta los fundamentos teóricos que indican que un sistema RFID, está compuesto por tres partes. La primera es el lector, la segunda la tarjeta y la tercera el sistema de procesamiento.

En el diseño del lector RFID se utilizó hardware libre como es la placa de desarrollo Arduino Uno, el diseño del lector se puede realizar con todos los tipos de placa Arduino (Arduino Uno, Leonardo, Mega, etc.), en si depende del módulo lector RFID, algunos lectores necesitan dos puertos seriales, otros solamente uno. Las placas Arduino son muy fáciles de usar, de bajo costo y muy versátiles, para el funcionamiento del prototipo solo es necesario conectarlo a un puerto USB del computador.

El tag o tarjeta identificativa debe trabajar a la misma frecuencia que el lector RFID, en este caso a 13,56 MHz, frecuencia de gran uso en sistemas de registro, por usar tarjetas pasivas, es decir no necesitan batería adicional, simplemente el lector induce un voltaje a la tarjeta que permitirá el paso de la trama al encontrarse dentro de un rango de lectura, presenta mayor durabilidad al no depender del desgaste de las baterías.

En cuanto al sistema de procesamiento, solo se requiere de una computadora, en la que se encuentre la base de datos; el programa realizado en php, para la conexión con la base de datos (BD) y el router MikroTik para la sala 2, o los proyectos en java que permiten la toma de datos del puerto serial Arduino y los envía a la BD y mediante comunicación TCP básica, desbloquean una máquina de mesa, este proyecto se podrá hacer ejecutable de tal manera que el bibliotecario no necesite un IDE para poder abrirlo.

El presente objetivo se cumplió al crearse un sistema con una frecuencia que opera a distancias cortas, ayudando a que el ruido ni los sistemas adyacentes influyan con el funcionamiento correcto del prototipo, por tanto, es un prototipo pequeño que no ocupa mucho espacio y no requiere gran cantidad de energía para su funcionamiento, alimentándose con voltaje continuo.

#### **Objetivo 2: Integrar el sistema a una base de datos libre *MySQL* y *PHP*.**

MySQL es una base de datos muy usado por su carácter de software libre de distribución gratuita, y al unirse con PHP, se obtiene PhpMyAdmin, que es una interfaz gráfica de gestión de base de datos, para cumplir este objetivo, simplemente se ocupó un paquete para la instalación de MySQL, Apache, PHP y la interfaz gráfica para administrar *MySQL*, llamado AppServ todo esto en Windows.

Al tener un alojamiento local, el siguiente paso es lograr que los datos del lector se suban automáticamente a la tabla de la base de datos relacional que es MySQL, en este caso se ocupa *PHP*, el programa en Arduino envía la trama al programa en PHP y este a su vez se conecta con la Base de datos local (Sala 2), para ello se requiere conocer el nombre de la base, la contraseña, el nombre de la tabla y a que campo va dirigida.

En cuanto al préstamo de máquinas en la sala 1, se usa un programa en java que se conecta con la base de datos y actualiza la información.

El objetivo dos se cumplió al lograr conectarse con la base de datos, dentro de la misma se crearon otras tablas en las que estaban los códigos de las tarjetas asociados con datos de los usuarios, como su nombre, cédula, carrera, módulo, etc.

**Objetivo 3: Desarrollar un portal cautivo con *Hotspot* para limitar la señal de internet inalámbrico solamente para las personas que llevan a cabo el registro.**

Para el préstamo de computadoras de mesa, no se utilizó ningún equipo adicional, ya que simplemente se realiza un programa en java, que hace que una interfaz o pantalla siempre se mantenga sobrepuesta, simulando un bloqueo, esto permite que el usuario no pueda usar la máquina, ya que la pantalla sobrepuesta limita su entrada a cualquier programa o a la navegación, esto es cuanto al cliente; para la máquina principal se realizó otro programa igualmente en java que abre un puerto virtual y envía una orden de desbloqueo al realizar el registro, así mismo se necesita que cada máquina de mesa tenga una *IP* fija, de tal manera que el programa desbloqueará una máquina, basándose en una *IP* aleatoria, que se encuentra dentro de una lista.

Sin embargo, en el control del wifi surge el inconveniente que las portátiles no siempre tendrán *IP* fija, es decir el usuario no lleva siempre la misma computadora, hasta incluso a veces utilizan celulares, Tablets, PDAs, etc., prevaleciendo el uso de dhcp, es decir no contamos con esa lista de *IP*s fijas del software de la sala 1, entonces surge el portal cautivo con *hotspot*, que, gracias a la autenticación, podemos integrarlo al registro automático.

Para el cumplimiento de este objetivo se usó un router mikrotik que es una solución flexible y fácil de implementar, con una red dinámica, controlada y segura, podemos crear múltiples usuarios *hotspot*, el nombre de los usuarios y contraseña corresponderá al código de la tarjeta RFID.

A todos los que intenten navegar, conectándose a la red inalámbrica de la biblioteca se les abrirá una página de autenticación muy común en los portales cautivos, en la que se pide un usuario y contraseña; sin embargo, solo la personas que se registren activarán ese usuario y la autenticación se validará, esto permite que solo los usuarios que llevan el registro podrán navegar en internet. Para lograr esto, se creó un programa en *PHP* que se conecta a la interfaz de MikroTik y desactiva todos los usuarios cada vez que se prenda la máquina donde se realiza todo el procesamiento, es así que, al iniciar el día, todos los usuarios están desactivados y se van habilitando en cuanto se pase una tarjeta RFID por el lector.

## 8. CONCLUSIONES

- Se automatizó el ingreso de 10 usuarios y el acceso al internet cableado e inalámbrico, mediante la lectura de las etiquetas identificativas RFID; se realizó en una pequeña muestra, pero este prototipo puede funcionar para 200 usuarios activos en *hotspot*.
- Los lectores RFID, no necesitan línea de vista, ni contacto físico con la etiqueta, por la frecuencia a la que trabajan; las etiquetas tienen mayor durabilidad, y su costo es medio, por tanto, son una buena alternativa como carnet identificativo, que le servirá al usuario para ingresar a las áreas dedicadas a la investigación como son las bibliotecas y tener acceso a los servicios de préstamo de máquinas e internet.
- El diseño de las tablas en la base de datos ayuda a que no cualquier tarjeta RFID desbloquee una máquina o permita el uso del Wifi, solamente lo harán los códigos de las tarjetas que se encuentran en las tablas, de igual forma permite exportar la información de los registros en diferentes formatos entre ellos .xls, .text, esta información es de gran utilidad para los encargados de la biblioteca para su posterior tabulación y creación de informes.
- La creación de un portal cautivo por Hardware, con la tecnología MikroTik, nos permite ingresar a la API (Interfaz de programación de aplicaciones) mediante un programa en php, este es un intermedio, entre Arduino y el router; la trama es tomada, luego pasa a PHP y este a su vez realiza la comunicación con mikrotik y la base de datos, finalmente se activan los usuarios *hotspot* permitiendo la navegación a internet.
- El IDE *Netbeans* para el lenguaje de programación java, es muy fácil de usar y permitió el diseño de la interfaz gráfica que interactúa con el usuario al indicarle la máquina de mesa asignada y la pantalla de bloqueo y desbloqueo.

## 9. RECOMENDACIONES

- Para evitar interferencias con campos electromagnéticos, ya que los sistemas RFID HF presentan sensibilidad, se recomienda tener al sistema lejos de motores eléctricos, los cuales generan este tipo de interferencias.
- Se recomienda no cambiar los campos de las tablas, estos fueron creados según los requerimientos de información de la biblioteca, la configuración del tamaño del campo se la hizo utilizando la cantidad necesaria de recursos, sin desperdiciar espacio de memoria.
- Se recomienda apagar la máquina donde se realiza el procesamiento y que esté dentro de la misma red, porque al prenderla se desactivarán todos los usuarios *hotspot* y se podrá iniciar con el proceso de activación individual al realizar el registro.
- Se recomienda que en la sala 1, las computadoras de mesa tengan direccionamiento *IP* estático, para asegurar la estabilidad del sistema, luego estas direcciones *IPs* serán colocadas en una lista del proyecto en java.
- Para una futura implementación se recomienda que para la sala 2 se implemente un servidor Radius externo, que no tiene ningún costo adicional y ofrece administración centralizada en toda la red, en el prototipo los usuarios fueron creados dentro del equipo porque se trabajó con una pequeña muestra de 10 personas, al crear más usuarios se consumen recursos del equipo MikroTik, por tanto, se debe tener un servidor Radius que haga ese trabajo, que usa como hardware mínimo 32 MB de RAM y 2 MB de disco duro.
- Se recomienda la creación de VLANs para aislar el tráfico, entre cliente y servidor y así brindar mayor seguridad estableciendo una comunicación directa.
- Si en caso de superarse la giga en tráfico se puede escalar a equipos Mikrotik giga, por ejemplo, el rb95g, que tienen mayor procesador y CPU.
- Para implementar el sistema de registro automático, se recomienda instalar en el servidor el software *Netbeans*, y en el cliente el paquete *jdk* versión 8 o superior, los dos prototipos deben estar conectados al *server* mediante cable USB, en las tablas de la base de datos se ingresan los datos del usuario y trama de la tarjeta RFID, y en el servidor Radius se crean nuevos usuarios para *hotspot* cuyo usuario y contraseña corresponden a la trama de la tarjeta.

- Si se requiere la administración remota de la base de datos (BD) se recomienda validar la IP remota, colocando la *IP* del PC que se quiera conectar al BD, otra alternativa es realizar esta conexión mediante una VPN<sup>4</sup>, usando el servicio SSH<sup>5</sup> o por escritorio remoto, en caso del equipo MikroTik el hotspot puede ser configurado para conectarse a un servidor *Radius* remoto y poder autenticar a los usuarios sin ningún problema.

---

<sup>4</sup> Una VPN es una red privada virtual, por sus siglas en inglés Virtual Private Network.

<sup>5</sup> SSH por sus siglas en inglés Secure Shell, en español intérprete de órdenes seguro para acceso remoto.

## 10. BIBLIOGRAFÍA

- [1] C. M. Salgado, «uam.mx,» 13 julio 2007. [En línea]. Available: <http://www.azc.uam.mx/publicaciones/enlinea2/num1/1-1.htm>. [Último acceso: 8 Noviembre 2015].
- [2] G. Izard, «Iesrivera,» Noviembre 2005. [En línea]. Available: <http://www.iesribera.es/documentos/Electronica/Codigo%20Barras%20EAN.pdf>.
- [3] J. A. A. Sánchez, «cinvestav,» Enero 2008. [En línea]. Available: <http://www.cs.cinvestav.mx/TesisGraduados/2008/tesisJorgeAlvarado.pdf>. [Último acceso: 26 noviembre 2015].
- [4] C. L. Pérez, «Controlarán asistencia de los estudiantes con un sistema biométrico,» 2016.
- [5] Spetel, «Spetel,» 2015. [En línea]. Available: <http://www.spetel.com/portfolio/sistema-de-banda-magnetica/>. [Último acceso: 28 noviembre 2016].
- [6] M. E. Casero, «Biblioteca Unirioja,» 17 septiembre 2013. [En línea]. Available: [http://biblioteca.unirioja.es/tfe\\_e/TFE000285.pdf](http://biblioteca.unirioja.es/tfe_e/TFE000285.pdf). [Último acceso: 27 diciembre 2015].
- [7] R. H. Atilano, «Saber.cic,» Mayo 2007. [En línea]. Available: <http://www.saber.cic.ipn.mx/cake/SABERsvn/trunk/Repositorios/webVerArchivo/319/1>. [Último acceso: 23 Diciembre 2015].
- [8] A. R. Hernández, 25 Noviembre 2009. [En línea]. Available: <http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/5441/1/C2.302.pdf>. [Último acceso: 28 diciembre 2015].
- [9] U. Ruiz, «UDLAP Bibliotecas,» 2011. [En línea]. Available: [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lep/urbina\\_r\\_rd/indice.html](http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/urbina_r_rd/indice.html). [Último acceso: 27 Abril 2016].
- [10] E. S. C. José María Ciudad Herrera, «upcommons,» 2005. [En línea]. Available: <https://www.google.com.ec/#q=ESTUDIO%2C+DISE%2C%91O+Y+SIMULACI%2C%93N+DE+UN+SISTEMA+DE+RFID+BASADO+EN+EPC>. [Último acceso: 18 Abril 2016].
- [11] D. C. M. D. A. R. S. L. G. Aguilar, «Repositorio Digital ESPE,» 2011. [En línea]. Available: <http://repositorio.espe.edu.ec/handle/21000/4697>. [Último acceso: 16



Enero 2016].

- [12] J. d. C. y. León, «Tecnología de identificación por Radiofrecuencia y sus principales aplicaciones,» Consejería de Fomento., España, 2007.
- [13] IberWave, «IberWave,» 3 agosto 2015. [En línea]. Available: <http://www.iberwave.com/index.html>. [Último acceso: 17 enero 2016].
- [14] A. B. A. B. Javier Portillo, «vt,» 2008. [En línea]. Available: [http://www.madrimasd.org/informacionidi/biblioteca/Publicacion/Vigilancia-tecnologica/descargar\\_documentos/fichero.asp?id=VT13\\_RFID.pdf](http://www.madrimasd.org/informacionidi/biblioteca/Publicacion/Vigilancia-tecnologica/descargar_documentos/fichero.asp?id=VT13_RFID.pdf). [Último acceso: 24 Diciembre 2015].
- [15] L. A. A. Saez, «Repositorio UACH,» 2007. [En línea]. Available: <http://cybertesis.uach.cl/tesis/uach/2007/bmfcia452e/doc/bmfcia452e.pdf>. [Último acceso: 15 Diciembre 2015].
- [16] akrocard, «akrocard,» 2016. [En línea]. Available: <http://akrocard.com/tarjetas-rfid-mifare-desfire/>. [Último acceso: 25 noviembre 2016].
- [17] Datasheet, «Elechouse Dataheets,» Mayo 2001. [En línea]. Available: [http://www.elechouse.com/elechouse/images/product/13.56MHZ\\_RFID\\_Module/mifare\\_S50.pdf](http://www.elechouse.com/elechouse/images/product/13.56MHZ_RFID_Module/mifare_S50.pdf).
- [18] J. S. E. G. Miguel Huguet, Administración de Sistemas Operativos de Red, Barcelona: UOC, 2008.
- [19] H. K. S. S. Avil Silberschatz, Fundamentos de Base de datos, España: McGrawHill, 2006.
- [20] M. Sierra, «apr.com,» 2006. [En línea]. Available: [http://aprenderaprogramar.com/index.php?option=com\\_attachments&task=download&id=500](http://aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=500). [Último acceso: 8 julio 2015].
- [21] L. H. Ibañez, Administración de Sistemas Gestores de Bases de datos, España: Ra-Ma, 2014.
- [22] MySQL, «MySQL.com,» 2015. [En línea]. Available: <http://www.mysql.com/documentation/index.html>. [Último acceso: 17 julio 2016].
- [23] J. G. Raghu Ramakrishnan, Sistema de Gestion de Bases de datos, España: McGrawHill, 2007.

- [24] P. G. D. P. R. R. Ángel Cobo, PHP Y MySQL tecnología para el desarrollo de aplicaciones web, España: Diaz de Santos, 2005.
- [25] A. M. Tapia, «Repositorio UPS,» Septiembre 2012. [En línea]. Available: <http://dspace.ups.edu.ec/handle/123456789/4167>. [Último acceso: 28 junio 2015].
- [26] D. C. Herrera, «Repositorio EPN,» 2011. [En línea]. Available: <http://bibdigital.epn.edu.ec/handle/15000/3953>. [Último acceso: 27 junio 2015].
- [27] Antamedia, «Antamedia,» [En línea]. Available: <http://www.antamedia.com/hotspot/>. [Último acceso: 28 junio 2015].
- [28] Endian, «Endian,» [En línea]. Available: <http://www.endian.com/>. [Último acceso: 28 junio 2015].
- [29] MikroTik, «MikroTik,» 2016. [En línea]. Available: <http://www.mikrotik.com/>. [Último acceso: 12 Diciembre 2015].
- [30] z.-. software, «Soluciones Biométricas avanzadas,» 26 Septiembre 2016. [En línea]. Available: [http://www.zk-software.com/tarjetas\\_mifare.html](http://www.zk-software.com/tarjetas_mifare.html). [Último acceso: 28 octubre 2016].
- [31] Elechouse, «Elechouse,» [En línea]. Available: [http://www.elechouse.com/elechouse/images/product/PN532\\_module\\_V3/PN532\\_%20Manual\\_V3.pdf](http://www.elechouse.com/elechouse/images/product/PN532_module_V3/PN532_%20Manual_V3.pdf). [Último acceso: 12 Diciembre 2015].
- [32] «Arduino Genuino,» 2016. [En línea]. Available: <https://www.arduino.cc/>. [Último acceso: 14 Diciembre 2015].
- [33] Elechouse, «Elechouse,» 2016. [En línea]. Available: [http://www.elechouse.com/elechouse/index.php?main\\_page=product\\_info&products\\_id=2156](http://www.elechouse.com/elechouse/index.php?main_page=product_info&products_id=2156). [Último acceso: 15 Diciembre 2015].
- [34] «Routerboard MikroTik,» 2016. [En línea]. Available: <https://routerboard.com/RB951Ui-2HnD>. [Último acceso: 16 diciembre 2015].
- [35] Routerboard, «Mikrotik,» [En línea]. Available: <http://routerboard.com/RB951Ui-2HnD>. [Último acceso: 30 junio 2015].

## 11. ANEXOS

### ANEXO 1

#### CÓDIGO LECTURA SERIAL ARDUINO UNO, SALA 1

```
#include <SPI.h>
#include <PN532_SPI.h>
#include "PN532.h"

PN532_SPI pn532spi(SPI, 10);
PN532 nfc(pn532spi);

void setup(void) {
  Serial.begin(115200);
  nfc.begin();
  uint32_t versiondata = nfc.getFirmwareVersion();
  nfc.setPassiveActivationRetries(0xFF);
  nfc.SAMConfig();
}
void loop(void) {

  boolean success;
  uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0, 0 };
  uint8_t uidLength;

  success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, &uid[0], &uidLength);

  if (success) {
    for (uint8_t i=0; i < uidLength; i++)
    {
      if (uid[i]<10){
        Serial.print(0);
        Serial.print(uid[i]);
      }
      else{
        Serial.print(uid[i], HEX);
      }
    }
    delay(1000);
  }
  else
  {
  }
}
```

## ANEXO 2

### CLASE: Serial.java

```
package clases;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.Socket;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.comm.CommPortIdentifier;
import javax.comm.NoSuchPortException;
import javax.comm.PortInUseException;
import javax.comm.SerialPort;
import javax.comm.UnsupportedCommOperationException;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Serial extends Thread {

    private final int puerto;
    private SerialPort sPuerto;

    public Serial(int puerto) {
        this.puerto = puerto;
    }

    @Override
    public void run() {
        //86 3F F5 16 tiene 8 caracteres sin contar espacios
        //private jTextField tf = new jTextField(10);
        int caracAss;
        String trama = "";
        conectar cc = new conectar();
        Connection cn = cc.conexion();

        while (true) {
            caracAss = leerDatosCode();
            if (caracAss < 10) {
```

```

        trama += "";
    } else if (caracAss == 10 || caracAss == 13) {
        trama += "@";
    } else {
        trama += (char) caracAss;
    }
}

trama = trama.replace(" ", "");
if (trama.length() == 8) {
    System.out.println("Trama: " + trama);

    try {
        Statement stmt = cn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM tarjetas t, usuarios u "
            + "WHERE t.id_tarjeta = u.id_tarjeta "
            + "AND t.codigo = '" + trama + "'");
        rs.next();
        PreparedStatement pst = cn.prepareStatement("INSERT INTO registro (id_persona, persona,
            carrera, módulo, máquina, código_cédula) VALUES (?, ?, ?, ?, ?, ?)");
        pst.setInt(1, rs.getInt("id_persona"));
        pst.setString(2, rs.getString("persona"));
        pst.setString(3, rs.getString("carrera"));
        pst.setString(4, rs.getString("modulo"));
        pst.setString(6, rs.getString("codigo"));
        int n = -1;

        for (int i = 0; i < ingresousuarios1.listaIp.size(); i++) {
            if (!ingresousuarios1.listaIp.get(i).isEstado()) {
                n = i;
                System.out.println(i+1);
                // JFrameGUI f = new JFrameGUI();
                // f.setVisible (true);
                // JFrameGUI.jTextField1.setText(String.valueOf(i+1));
                pst.setInt(5, i+1); // el número de maquina a la base de datos
                pst.executeUpdate();
                break;
            }
        }

        if (n != -1) {
            IpConexion aux = ingresousuarios1.listaIp.get(n);
            Socket socket = new Socket(ingresousuarios1.listaIp.get(n).getIp(), 7777);
            DataOutputStream out = new DataOutputStream(socket.getOutputStream());
            out.writeUTF("desbloquear");
            aux.setEstado(true);
            System.out.println("bien");
            numeroMq(n+1);
        }

        } catch (SQLException ex) {
            Logger.getLogger(Serial.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(Serial.class.getName()).log(Level.SEVERE, null, ex);
        }

        trama = ""; //vaciar la trama
    }

}

}

public void numeroMq(int s) {
    JFrameGUI JG = new JFrameGUI();
    JG.setVisible(true);
    JG.jTextField1.setText(String.valueOf(s)); // aqui se lo visualiza en el gui
}

public boolean abrirPuerto() {
    try {
        CommPortIdentifier id_Puerto = CommPortIdentifier.getPortIdentifier("COM" + puerto);
        sPuerto = (SerialPort) id_Puerto.open("SBC", 5000); //tiempo de bloqueo
        setParametros(115200, SerialPort.DATABITS_8, SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
        return true;
    } catch (PortInUseException ex) {
        System.err.println("Puerto en uso por otra aplicación.");
    } catch (NoSuchPortException ex) {
        System.err.println("No se ha encontrado el puerto serial o no estan disponibles los drivers.");
    }

    return false;
}

```

```

private boolean setParametros(int baudRate, int dataBits, int stopBits, int paridad) {
    try {
        sPuerto.setSerialPortParams(baudRate, dataBits, stopBits, paridad);
        return true;
    } catch (UnsupportedCommOperationException ex) {
        System.out.println("El Puerto no soporta la operación");
    }
    return false;
}

private int leerDatosCode() {
    InputStream is;
    try {
        is = sPuerto.getInputStream();
        return is.read();
    } catch (IOException | NullPointerException ex) {

    }

    return 0;
}
}

```

## ANEXO 3

### CLASE: conectar.java

```

package clases;

import java.sql.Connection;
import java.sql.DriverManager;

public class conectar {
    Connection conectar=null;
    public Connection conexion ()
    {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conectar=DriverManager.getConnection("jdbc:mysql://localhost/registro","root","telecomunicaciones");
        } catch (Exception e) {
            System.out.print(e.getMessage()); //Para mostrar error
        }
        return conectar;
    }
}

```

## ANEXO 4

### CLASE: ingreseusuarios1.java

```
package clases;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ingreseusuarios1 extends javax.swing.JFrame {
    public static List<IpConexion> listaIp = new ArrayList<>();
    JFrameGUI JG;

    public ingreseusuarios1() {
        initComponents();
        listaIp.add(new IpConexion("192.168.4.1", false));
        //listaIp.add(new IpConexion("192.168.4.2", false));
        //listaIp.add(new IpConexion("192.168.4.8", false));

        Serial s = new Serial(5);
        if (s.abrirPuerto()) {
            s.start();
        } else {
            System.exit(0);
        }
    }

    private void btngrabarActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            Statement stmt = cn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM usuarios WHERE cedula = '"+txtcod.getText()+"'");
            rs.next();

            PreparedStatement pst = cn.prepareStatement("INSERT INTO registro (id_persona, persona, carrera,
            módulo, máquina, código_cédula) VALUES (?, ?, ?, ?, ?, ?)");
            pst.setInt(1, rs.getInt("id_persona"));
            pst.setString(2, rs.getString("persona"));
        }
    }
}
```

```

pst.setString(3, rs.getString("carrera"));
pst.setString(4, rs.getString("modulo"));
pst.setString(6, rs.getString("cedula"));

int n = -1;
for (int i = 0; i < ingresousuarios1.listaIp.size(); i++) {
    if (!ingresousuarios1.listaIp.get(i).isEstado()) {
        n = i;
        System.out.println(i+1);// aqui esta el número de maquina
        pst.setInt(5, i+1);// el número de maquina a la base de datos
        pst.executeUpdate();
        break;
    }
}
if (n != -1) {
    IpConexion aux = ingresousuarios1.listaIp.get(n);
    Socket socket = new Socket(ingresousuarios1.listaIp.get(n).getIp(), 7777);
    DataOutputStream out = new DataOutputStream(socket.getOutputStream());
    out.writeUTF("desbloquear");
    aux.setEstado(true);
    System.out.println("bien");
    numeroMq(n+1);
}

} catch (SQLException ex) {
    Logger.getLogger(ingresousuarios1.class.getName()).log(Level.SEVERE, null, ex);
} catch (IOException ex) {
    Logger.getLogger(ingresousuarios1.class.getName()).log(Level.SEVERE, null, ex);
}
}

public static void main(String args[]) {
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(ingresousuarios1.class.getName()).
            log(java.util.logging.Level.SEVERE, null, ex);} catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(ingresousuarios1.class.getName()).
            log(java.util.logging.Level.SEVERE, null, ex);} catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(ingresousuarios1.class.getName()).
            log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(ingresousuarios1.class.getName()).
            log(java.util.logging.Level.SEVERE, null, ex);
    }

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ingresousuarios1().setVisible(true);
        }
    });
}

public void numeroMq(int s) {
    JG = new JFrameGUI();
    JG.setVisible(true);
    JG.jTextField1.setText(String.valueOf(s));// aqui se lo visualiza en el gui
}

private javax.swing.JButton btngrabar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField txtcod;
conectar cc= new conectar();
Connection cn=cc.conexion();
}

```



## ANEXO 5

### CLASE: Modelo.java

```
package hilos;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Modelo extends Thread {

    int c = 3600;
    JFrameBlocked jf;
    JFrameGUI jc;
    @Override
    public void run() {
        while (true) {
            try {
                if (c == 3600) {
                    jf = new JFrameBlocked();
                    jf.setVisible(true);
                }
                c++;
                sleep(1000);
            } catch (InterruptedException ex) {
                Logger.getLogger(Modelo.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }

    public void desbloquear() {
        c = 0;
        jf.dispose();
        jc = new JFrameGUI();
        jc.setVisible(true);
    }
}
```

## ANEXO 6

### CLASE: Servidor.java

```
package hilos;
import java.io.DataInputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Servidor extends Thread {

    Modelo m;
    ServerSocket serverSocket;

    public Servidor(Modelo m) {
        this.m = m;
    }

    @Override
    public void run() {
        try {
            serverSocket = new ServerSocket(7777);
            while (true) {
                Socket socket = serverSocket.accept();
                DataInputStream dis = new DataInputStream(socket.getInputStream());
                String data = dis.readUTF();
                if (data.equals("desbloquear")) {
                    m.desbloquear();
                }
            }
        } catch (IOException ex) {
            Logger.getLogger(Servidor.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

## ANEXO 7

### CLASE: jBlocked.java

```
package hilos;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;
import javax.swing.JFrame;

public class jBlocked {

    private JFrame jframe=null;
    public jBlocked( JFrame f )
    {
        this.jframe = f;
    }
    public void block()
    {
        ScheduledExecutorService scheduler = Executors.newSingleThreadScheduledExecutor();
        scheduler.scheduleAtFixedRate(
            new Runnable()
            {
                @Override
                public void run() {
                    front();
                }
            }, 500, 50 , TimeUnit.MILLISECONDS ); //comienza dentro de 1/2 segundo y luego se repite cada N segundos
    }

    public void front()
    {
        jframe.setExtendedState( JFrame.MAXIMIZED_BOTH );//maximizado
        jframe.toFront();
    }
}
```

## ANEXO 8

### CLASE: Test.java

```
package hilos;

public class Test {

    public static void main(String[] args) {
        Modelo t = new Modelo();
        t.start();

        Servidor c = new Servidor(t);
        c.start();
    }
}
```

## ANEXO 9

### PROGRAMA ARDUINO PARA MÁQUINAS INALÁMBRICAS

```
#include <SoftwareSerial.h>
#include <SPI.h>
#include <Ethernet.h>
SoftwareSerial mySerial(6, 7); // Pin7 Rx, Tx pin6
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0x96, 0xB4 };
byte ip[] = { 192,168,4,3}; //direccion del puerto del router
byte server[] = { 192,168,4,4}; // direccion del server maquina principal
EthernetClient client;
byte posicion=0;
int comlen =0;
int out_flag =0;
int C1=0;
int C2=0;
int C3=0;
int C4=0;
int a=0;
int cont=0;
void setup()
{
    Serial.begin (9600);
    delay(1000);
    Ethernet.begin (mac,ip);
    mySerial.listen();
    mySerial.begin(9600);
    delay(10);
    mySerial.write (0x02);
}
void loop ()
{
    while (mySerial.available ()) //Mientras este activo el puerto serial
    {
        cont=cont+1;
        byte C1= mySerial.read();
        delay(5);
        byte C2= mySerial.read();
        delay(5);
        byte C3= mySerial.read();
```

```

        byte C4= mySerial.read();

        out_flag =1;
        if (cont==1)
        {
if (client.connect(server, 8000))
{
    if (C1<10){Serial.print(0);Serial.print(C1,HEX);}
    else {Serial.print(C1,HEX);}
    if (C2<10){Serial.print(0);Serial.print(C2,HEX);}
    else {Serial.print(C2,HEX);}
    if (C3<10){Serial.print(0);Serial.print(C3,HEX);}
    else {Serial.print(C3,HEX);}
    if (C4<10){Serial.print(0);Serial.print(C4,HEX);}
    else {Serial.print(C4,HEX);}
    client.print("GET http://192.168.4.4/test4/reportes2.php?tt=");
    if (C1<10){client.print(0);client.print(C1,HEX);}else{client.print(C1,HEX);}
    if (C2<10){client.print(0);client.print(C2,HEX);}else{client.print(C2,HEX);}
    if (C3<10){client.print(0);client.print(C3,HEX);}else{client.print(C3,HEX);}
    if (C4<10){client.print(0);client.print(C4,HEX);}else{client.print(C4,HEX);}
client.println(" HTTP/1.0");
    client.println(" User-Agent: Arduino 1.0");
    client.println();
    a=0;
    cont=0;
}
else
{
    Serial.print("Conexion no existosa");
}
    if (client.connected()) {
}
else {
    Serial.println("Desconectado");
}
    client.stop();
    client.flush();

    }

}
    if(out_flag > 0)
    {
        out_flag =0;
    }
}

```

## ANEXO 10

### Activar2.php

```
<?php require_once('api_mt_include2.php'); ?>

<?php
$myusername = "anver";>
$mypassword = "12345";
$ipofsvr="192.168.3.1";
$api_puerto=8728;

$API = new routeros_api();
$API->debug = true;

if ($API->connect($ipofsvr, $myusername, $mypassword, $api_puerto))
{
    $API->write('/system/script/run',false);
    $API->write(".id=creando");
    $API->read(false);

    $API->disconnect();
}
```

## ANEXO 11

### Reportes2.php

```
<?php require_once('api_mt_include2.php'); ?>
<?php
//conexión
$host = "localhost";
$user = "root";
$pw = "thebest";
$db = "registro";

$username = "anver"; //
$password = "12345";
$ipofsvr="192.168.3.1"; // direccion del hotspot
$api_puerto=8728;
$API = new routeros_api();
$API->debug = true;

$dato = htmlspecialchars($_GET["tt"], ENT_QUOTES);
echo $dato;
//conexión con la base de datos mysql

$con = mysql_connect ($host, $user, $pw) or die ("Problemas al conectarse con el servidor");
echo "Conexion exitosa";
mysql_select_db ($db,$con) or die ("Problemas al seleccionar a Base de Datos");
echo "Conexion exitosa de la DB";

$consulta = "SELECT * FROM usuarios u, tarjetas t WHERE (u.id_tarjeta = t.id_tarjeta) AND t.codigo = '$dato'";
$resultado = mysql_query($consulta,$con);
echo $resultado;

while ($fila = mysql_fetch_assoc($resultado)) {
    mysql_query("INSERT INTO registro_wifi (persona, carrera, modulo, id_persona) VALUES ('".$fila['persona']."',
    '".$fila['carrera']."', '".$fila['modulo']."', '".$fila['id_persona']."'")", $con);
    echo "Registro Correcto";
    echo $fila;

    if ($API->connect($ipofsvr, $username, $password, $api_puerto))
    {
        echo "Registro Correcto";
        $API->write('/ip/hotspot/user/enable', false);
        $API->write("=.id=".$dato, true);
        $API->read(false);
        $API->disconnect();
    }
}
?>
```

## ANEXO 12

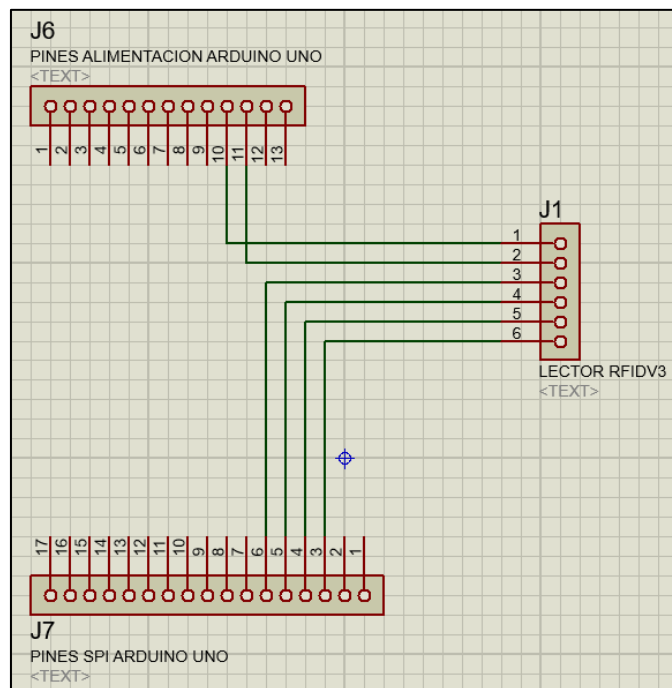
Script en Winbox.

```
/ip firewall address-list remove [find comment="Desactivar"];
delay (1);
local number [/ip hotspot user print count-only];
:for e from 0 to ($number-1) do={
local user [/ip hotspot user get number=$e name];
/ip firewall address-list add list=$user address=0.0.0.0 comment=Desactivar
disabled=yes
}
:foreach a in[/ip firewall address-list find disabled dynamic=no] do={
local coment [/ip firewall address-list get $a comment];
local list [/ip firewall address-list get $a list];
if ($coment="Desactivar") do={
put $list;
/ip hotspot user disable [find name="$list"];
/ip hotspot cookie remove [find user="$list"];
/ip hotspot active remove [find user="$list"];
}}
```

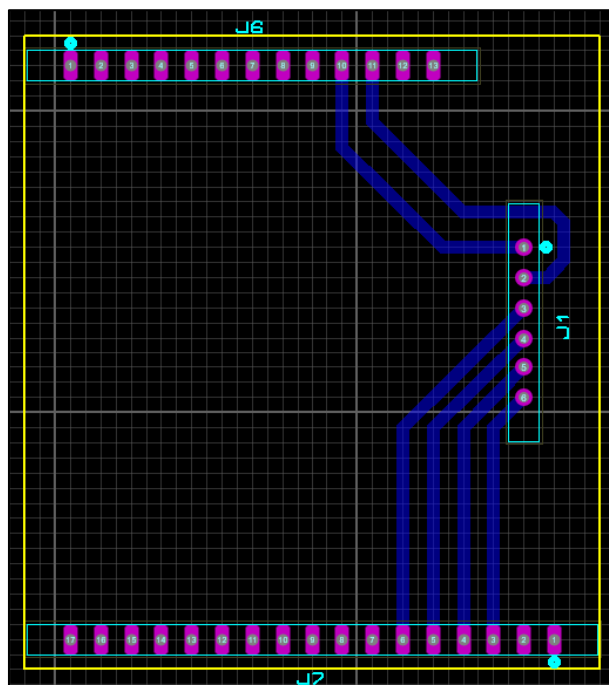


## ANEXO 13

(a) Placa diseñada en Proteus (ISIS), (b) Proteus (ARES) para la sala 1



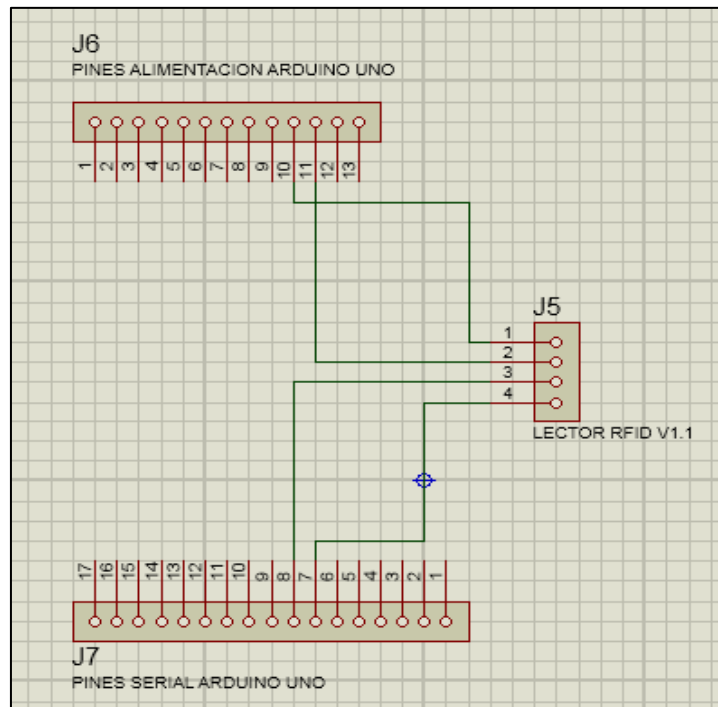
(a)



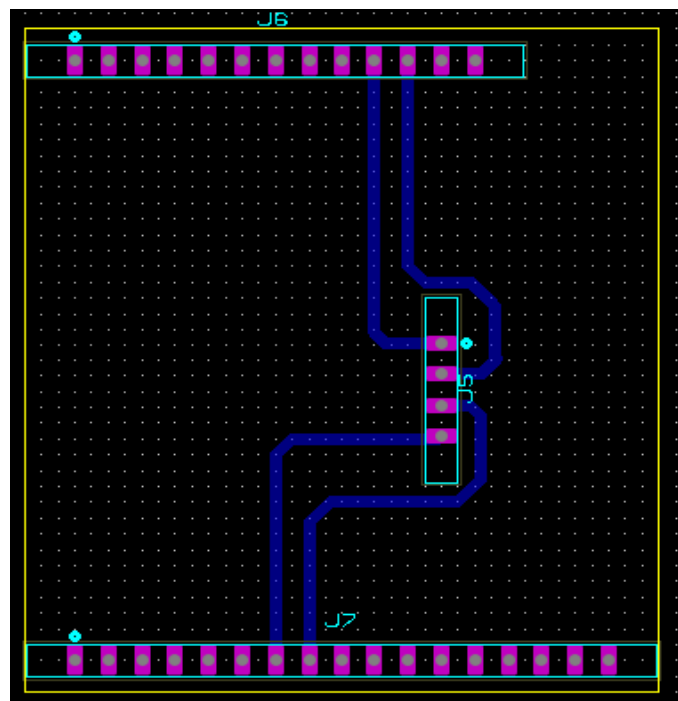
(b)

## ANEXO 14

(a) Placa diseñada en Proteus (ISIS), (b) Proteus (ARES) para la sala 2



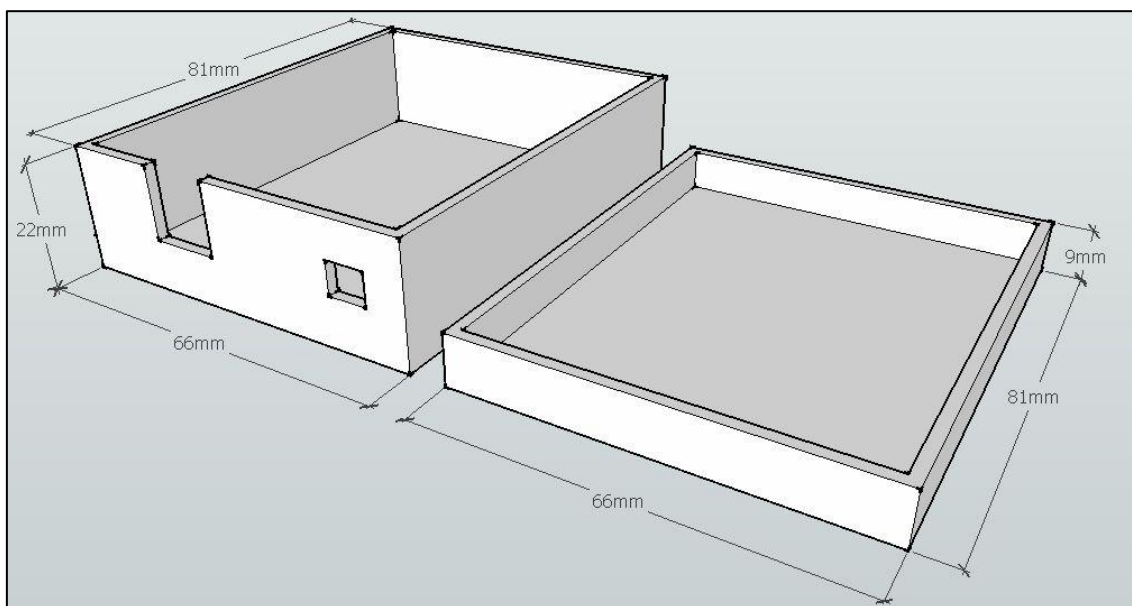
(a)



(b)

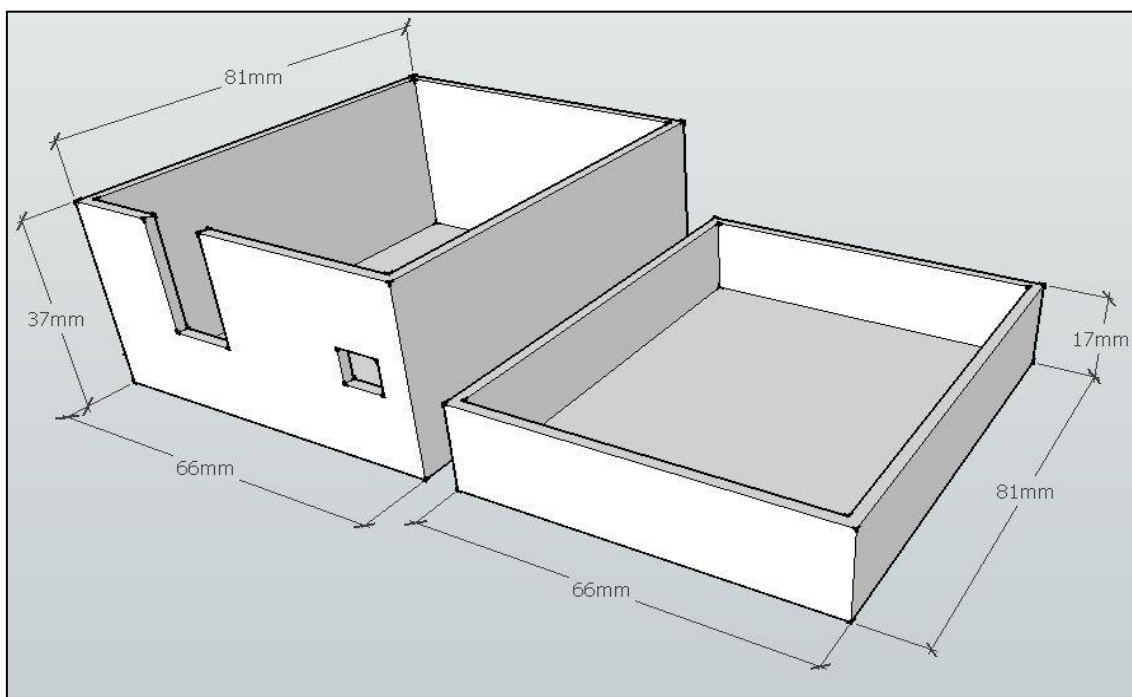
## ANEXO 15

Caja diseñada en sketchup e impresa en 3D, para la sala 1



## ANEXO 16

Caja diseñada en sketchup e impresa en 3D, para la sala 2



## ANEXO 17

Prototipo Sala 1, por fuera (a), por dentro (b).



(a)



(b)

## ANEXO 18

Prototipo Sala 2, por fuera (a), por dentro (b).



(a)



(b)