



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

SISTEMA ONLINE DE INVENTARIO Y CONTROL AUTOMATIZADO DE
ENTRADA Y SALIDA EN BODEGA DE BICICLETAS BICIQ A TRAVÉS DE
TECNOLOGÍA POR IDENTIFICACIÓN DE RADIO FRECUENCIA RFID

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniera en Electrónica y Redes de
Información

Profesor Guía

Ing. Jean Michel Clairand Gómez

Autora

Diana Ximena Cueva Cañarte

Año

2016

DECLARACIÓN PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el (los) estudiante(s), orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

Jean Michel Clairand Gómez
Ingeniero Electrónico
C.C.:1714736681

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro (amos) que este trabajo es original, de mi (nuestra) autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Diana Ximena Cueva Cañarte
C.C.: 1715049167

AGRADECIMIENTOS

A mi padre y hermanas que han sido siempre el motor para seguir adelante y culminar mis metas enseñándome que con esfuerzo, dedicación y constancia nada es imposible.

A mi tutor el Ing. Jean Michel Clairand quien me brindó su orientación y guía en el presente trabajo de titulación.

A la universidad de las Américas por acogerme en ella y a los docentes que con sus conocimientos fueron partícipes en mi formación académica.

DEDICATORIA

A la inmortal memoria de mi madre, a mi padre y hermanas que fueron el estímulo para mi constante superación y que nuestra identidad se proyecte al infinito.

RESUMEN

La ciudad de Quito cuenta con un sistema de transporte público denominado BiciQ, el cual consiste en ofrecer a la capital un servicio complementario de fácil y rápida transportación a través de bicicletas públicas para trayectos cortos dentro del hipercentro de la ciudad prometiendo a sus usuarios una alternativa gratuita, ecológica y saludable.

El presente proyecto tiene como objetivo desarrollar un sistema online de inventario y control de entrada y salida de las bicicletas públicas BiciQ en bodega basado en la Tecnología de Identificación por Radiofrecuencia, este sistema muestra el estado en el que se encuentran las bicicletas, registra el ingreso y salida de las mismas y permite un control de stock de repuestos con el fin de automatizar y mejorar la administración de la logística de BiciQ.

Para el desarrollo de este proyecto fue fundamental una amplia recolección de información basada en la búsqueda de interrogantes que permitan aclarar el funcionamiento de la tecnología RFID. El prototipo comprende de dos partes principales: el hardware RFID donde se detallan los elementos empleados (etiquetas, lector, arduino uno, etc.), y el software RFID donde se desarrolla una página web para el procesamiento y gestión de la información enviada por el hardware. Primero se identifican las bicicletas con las etiquetas para su control, el lector mediante comunicación por radiofrecuencia lee el código de cada etiqueta obteniendo su información, finalmente, los datos extraídos por el lector se envían al software quien los procesa y almacena y los trasmite al host final. Mediante pruebas del prototipo tanto técnicas como de software se verifica que el sistema desarrollado satisfice y cumpla con los objetivos y requisitos funcionales.

Con este proyecto se pretende mejorar el actual sistema de inventario y control de entrada y salida en bodega de las bicicletas de BiciQ mediante su automatización con el fin de mejorar la administración de la logística brindar un mejor servicio a sus usuarios.

ABSTRACT

The city of Quito has a public transportation system called BiciQ which provides easy and fast transportation through the use of public bicycles to the capital as a complementary service for short journeys in the city which promises users a free, ecological and healthy alternative.

This project aims to develop an online inventory system and control of entry and exit of public BiciQ bicycles in warehouse based on radiofrequency identification technology, this system shows the state in which bicycles are, registers their income and output and allows control of stock of spare parts in order to automate and improve the administration of BiciQ logistics.

A wide collection of information was based around the search for questions that clarifies the functionality of RFID technology and was fundamental for the development of this project. The prototype consists of two main parts: the RFID hardware detailing the elements used (tags, RFID reader, arduino Uno, etc.), and the RFID software where develops a webpage for the processing and management of the information sent by the hardware. First, bicycles are identified with tags for control and monitoring through radiofrequency communication. Next, the data is extracted by the reader and is sent to software, which processes, stores and transmits them to the final host. The technical and software testing of the prototype verified that the developed system met and complied with the objective and functional requirements.

This project seeks to improve the current system of inventory and control input and output of BiciQ bicycles in warehouse through its automation in order to improve the management of logistics to provide better service to its users.

ÍNDICE

INTRODUCCIÓN	1
1. MARCO TEÓRICO	4
1.1 Definiciones Fundamentales.....	4
1.2 Tecnología RFID.....	5
1.3 Arquitectura de un sistema RFID	10
1.3.1 Etiqueta RFID	11
1.3.2 Antena	18
1.3.3 Lectores.....	21
1.3.4 Software RFID	23
1.4 Clasificación RFID.....	23
1.5 Regulación y Estandarización	25
1.5.1 ISO	25
1.5.2 EPC GLOBAL.....	27
2. CONSTRUCCIÓN DEL PROTOTIPO	29
2.1 Funcionamiento actual de la bodega de BiciQ	29
2.2 Tecnología a utilizar.....	30
2.2.1 Ventajas de la tecnología RFID frente al Código de Barras	30
2.3 Esquema del prototipo	31
2.4 Componentes del Hardware	33
2.4.1 Módulo lector ID-12LA	33
2.4.2 Arduino UNO	36
2.4.3 Arduino Ethernet Shield.....	38
2.5 Diseño del Lector RFID.....	40
3. DESARROLLO DEL SOFTWARE	47
3.1 Software del Arduino	47
3.1.1 Lectura de Etiquetas.....	47
3.1.2 Conexión del Ethernet Shield	51
3.1.3 Programa Arduino.....	54
3.2 Entorno del Software del Sistema	57
3.2.1 Base de Datos	57

3.2.2	Lenguaje de Programación.....	57
3.2.3	Servidor	58
3.3	Análisis de Requerimientos.....	58
3.3.1	Determinación de los requerimientos de información	59
3.3.2	Diagrama de casos de uso	60
3.3.2.	Diagrama de casos de uso	60
3.3.3	Casos de uso.....	61
3.4	Descripción de la Base de Datos.....	67
3.4.1	Diagrama general de la base de datos	70
3.5	Desarrollo de interfaz de usuario.....	70
3.5.1	Interfaz de Administrador.....	71
3.5.2	Interfaz de Mecánico	77
3.5.3	Interfaz de Usuario	80
4.	PRUEBAS Y COSTO DEL PROTOTIPO	83
4.1	Pruebas de Hardware.....	83
4.2	Pruebas de Software	84
4.3	Resultados índice de Satisfacción.....	87
4.4	Análisis de Costos.....	92
4.4.1	Cuadro comparativo de costos	95
5.	CONCLUSIONES Y RECOMENDACIONES.....	97
5.1	Conclusiones	97
5.2	Recomendaciones.....	98
	REFERENCIAS.....	100
	ANEXOS	104

INTRODUCCIÓN

La ciudad de Quito cuenta con una alternativa gratuita, ecológica y saludable de transportarte denominada BiciQ, la cual es un servicio que ofrece bicicletas públicas a los quiteños para facilitar su movilidad dentro del hipercentro de la ciudad. (BiciQ, s.f)

Una de las tantas obligaciones del personal de BiciQ es controlar el estado en el que se encuentran las bicicletas al ser devueltas por los usuarios. Si las bicicletas tienen algún daño ya sea con los frenos, llantas, asientos, etc., son enviadas a la bodega para que los mecánicos realicen el mantenimiento adecuado con el cambio de piezas en caso de ser necesario. Al terminar de realizar cualquier mantenimiento de las bicicletas, se registra manualmente todas las reparaciones realizadas en cada una de ellas con los repuestos utilizados, posteriormente esa información es devuelta al administrador quien se encarga de realizar reportes y el control del stock de los repuestos para su análisis e inspección. Debido a que este proceso es manual lastimosamente se reduce el tiempo en la mano de obra de los operadores de BiciQ lo cual podrá generar deficiencia en el servicio que se ofrece a la ciudadanía.

Por ello, surge la intención de crear un sistema de control online de la información que se maneja en la logística a través de la tecnología por identificación de radiofrecuencia (RFID) con el fin de optimizar tiempos y recursos en BiciQ.

La tecnología por identificación de radiofrecuencia se encarga de identificar automáticamente a personas, animales u objetos mediante ondas de radio, esta identificación es realizada a través de etiquetas (*tags*) de RFID que poseen un código único que identifica a cada objeto (en este caso las bicicletas públicas) y que al pasar por el lector RFID le transmite esa información, el lector transforma las ondas de radio enviadas por el *tag* RFID en datos digitales que se los puede emitir a computadores para su uso.

Los sistemas con RFID han tenido mayor auge en los sectores industriales, médicos y de consumo especialmente en la cadena de suministros para el manejo de inventarios y logística debido a su fácil implementación y costo por lo que se convierte en una tecnología muy asequible. Actualmente, esta tecnología se ha prestado para la realización de estudios que permiten solventar y satisfacer necesidades y mejoras en los recursos empresariales. En el 2013, se publicó un artículo basado en la creación de una Infraestructura inalámbrica para la gestión de inventarios de petróleo y gas, utilizando tecnología RFID con el fin de lograr un sistema de administración de inventario online que permita monitorear los eventos esenciales de petróleo y gas en altamar utilizando una aplicación web. (Vellingiri, Ray y Kande, 2013).

En el 2012, se realizó un estudio que propone la implementación de un marco de gestión de desastres mediante un sistema online de planificación y control de inventario de emergencia humanitaria durante las operaciones de socorro basado en el seguimiento a tiempo real en la transportación de los suministros de emergencia durante el periodo de asistencia que permita el uso y distribución eficiente de los mismos mediante la integración de la tecnología RFID. (Ozguven y Ozbay, 2012). Por otro lado, con el fin de resolver algunos problemas que sufren los sistemas RFID como: escalabilidad de datos, ataques inalámbricos (*Man in the middle*), capacidades de las etiquetas RFID, etc., un grupo de investigadores realizó un estudio de un protocolo de autenticación basado en la nube para sistemas en la cadena de suministros RFID, mediante una novedosa arquitectura *cloud computing* con esquemas de autenticación, transferencia de propiedad e intercambio de datos para brindar mayor seguridad, un óptimo entorno y mejorar la eficiencia en la cadena de suministros. (Lin, Hsu y Cheng, 2014)

Considerando el éxito y aceptación que ha tenido esta tecnología en distintas aplicaciones de la industria, mediante este proyecto se pretende sistematizar y mejorar la logística de las bicicletas públicas BiciQ al entrar y salir de la bodega a fin de controlar su información por medio de un sistema online que optimice

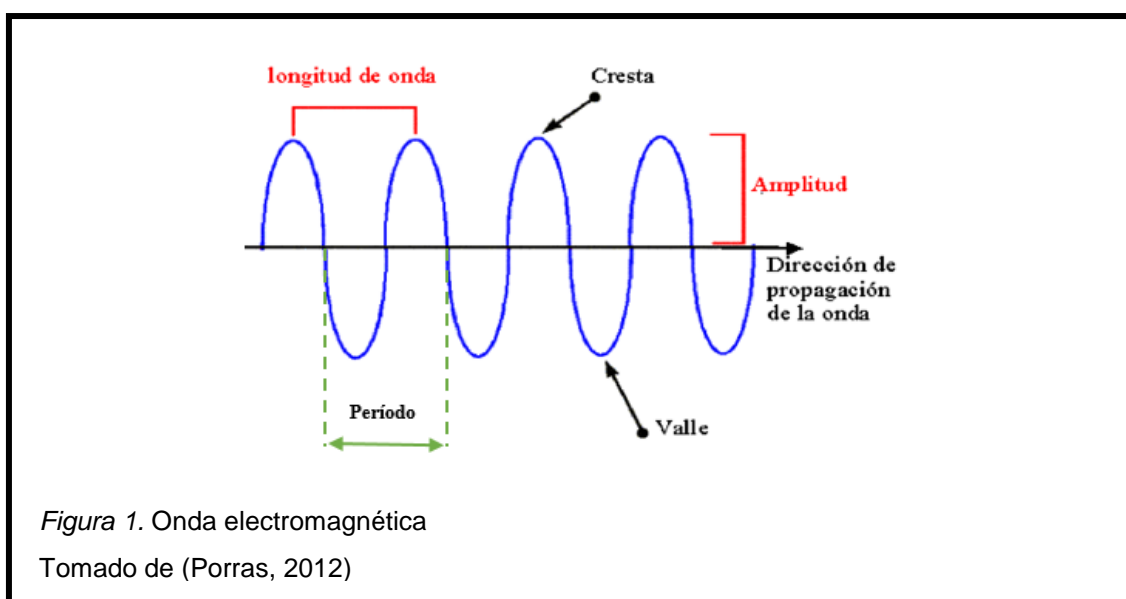
los procesos que se registran manualmente por el personal logrando incrementar la eficiencia, reducir errores en el ingreso de información y liberar a los operadores para que puedan realizar sus tareas eficientemente a través de tecnología por RFID. (PC-Doctor, s.f)

1. MARCO TEÓRICO

1.1 Definiciones Fundamentales

Las ondas electromagnéticas son perturbaciones que no necesitan un medio material para propagarse debido a variaciones en los campos magnéticos y eléctricos, y transportan equilibradamente energía de ambas clases a la velocidad de la luz, en función de su frecuencia, las ondas electromagnéticas tienen diferentes características y producen fenómenos diversos, su clasificación según este criterio se denomina espectro radioeléctrico.

Los espectros se pueden observar por medio de espectroscopios los cuales permiten realizar medidas sobre el espectro tales como la longitud de onda, la frecuencia y la intensidad de la radiación. (Porras, 2012)



A la longitud de onda se la entiende como la distancia que existe entre dos crestas continuas, el punto más alto de una onda se denomina cresta y el más bajo se denomina valle. Un ciclo es una onda completa y el período es la distancia en que se completa un ciclo de la onda electromagnética. La amplitud de una onda es la distancia desde el punto más alto de la onda (cresta) hasta la base de la onda (el eje horizontal de equilibrio) o desde el punto más bajo de la onda (valle) hasta la base de la onda. (Porras, 2012)

La radiofrecuencia es el conjunto de frecuencias situadas dentro de la gama de entre los 3Hz y 300GHz, correspondiente a la parte menos energética del espectro electromagnético, las frecuencias comprendidas en la tecnología de identificación por radiofrecuencia van entre los 30KHz a los 5,8Ghz.

1.2 Tecnología RFID

La tecnología de identificación por radiofrecuencia mejor conocida como RFID (*Radio Frequency Identification*) surgió en la segunda Guerra Mundial, sin embargo, fue hasta la década de los 80's que se la consideró como una pieza fundamental para aplicaciones comerciales en el negocio de la gestión y el seguimiento de las cadenas de suministro, especialmente en el de materiales, fabricación y la venta minorista (Dataflows, s.f.)

Tabla 1. Cronología de la tecnología RFID

DÉCADA	EVENTO
1940-1950	La tecnología del radar fue por primera vez usada para identificar aeronaves enemigas. Harry Stockman inventa la tecnología RFID en 1948. Exploración de técnicas RFID
1960 - 1970	Desarrollo de la teoría de RFID. Tecnología enfocada para establecimientos comerciales para contrarrestar el robo de mercadería.
1970 - 1980	Explosión de desarrollo RFID. Realización de pruebas y producción de dispositivos de RFID.
1980 - 1990	La década de 1980 se convirtió en la década de plena convergencia de la tecnología RFID. Una clave para la rápida expansión de las aplicaciones RFID fue el desarrollo de la computadora personal (PC) que permitió la recopilación y gestión de datos de los sistemas RFID conveniente y económica.
1990 - 2000	Aparición de normas y estándares. RFID se convierte en una parte de la vida cotidiana de todos.

Adaptado de (Transcore, 2001, p.7.)

Esta tecnología permite la identificación de objetos, animales o personas a cierta distancia sin necesidad de contacto, ni siquiera visual, que interroga una

característica única del objeto mediante ondas de radio. Su aplicabilidad está basada en escenarios enfocados en la seguridad y control de acceso, administración en cadenas de suministro, logística, habilitación de entrada y salida de vehículos, identificación/control de mascotas mediante un chip que es colocado en la piel del animal, optimización de stocks, control del peaje, etc.

La identificación por radiofrecuencia es un sistema en el que los datos son almacenados de manera inalámbrica mediante elementos añadidos a los objetos para facilitar su identificación, estos elementos son conocidos como etiquetas, transpondedores o *tags* RFID (compuestos por un chip de silicio y unido a una antena de radiofrecuencia) los cuales se encargan de intercambiar la información, mediante ondas de radio, con un dispositivo transmisor/receptor conocido como lector de RFID.

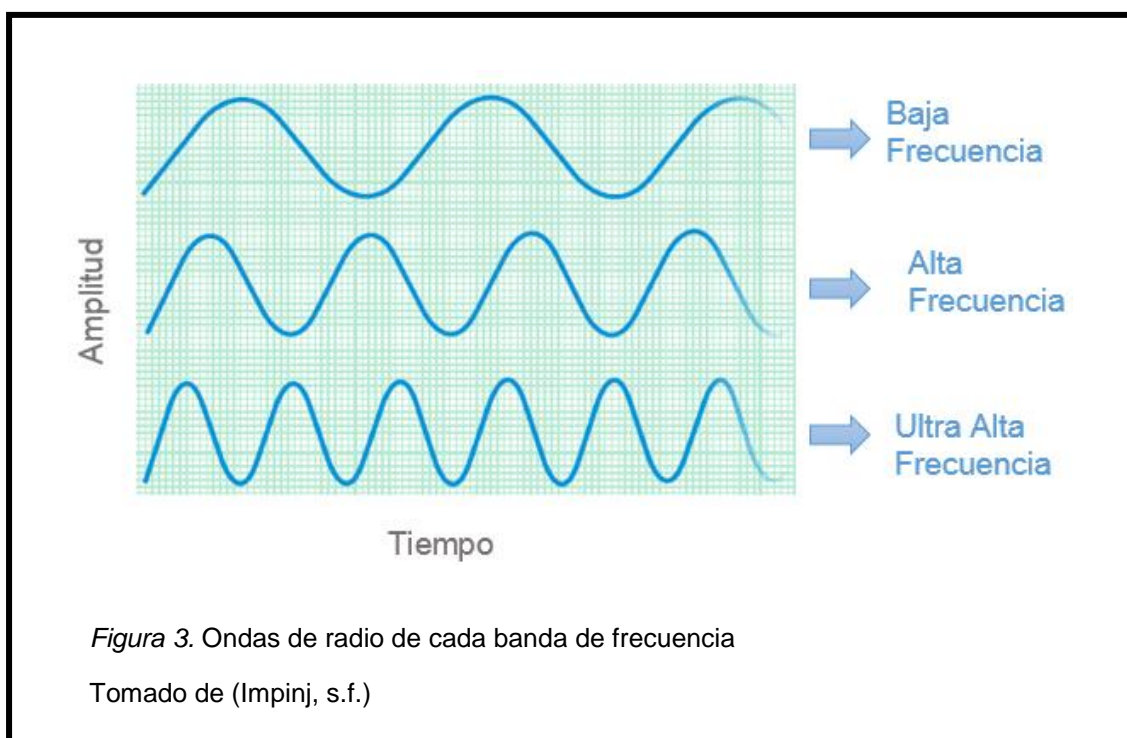
A la tecnología RFID se le ha asignado varias bandas de frecuencia, las cuales varían en todo el mundo, sus distintas frecuencias de operatividad dependen de la aplicación:

Banda de Frecuencias	Descripción	Rango
125KHz – 134 KHz	LF (Baja Frecuencia)	Hasta 45 cm.
13,553MHz- 13,567 MHz	HF (Alta Frecuencia)	De 1 a 3 m.
400 MHz – 1000MHz	UHF (Ultra Alta Frecuencia)	De 3 a 10 m.
2,45 GHz – 5,8 GHz	Microondas	Más de 10m.

Figura 2. Rango de frecuencias utilizadas en RFID

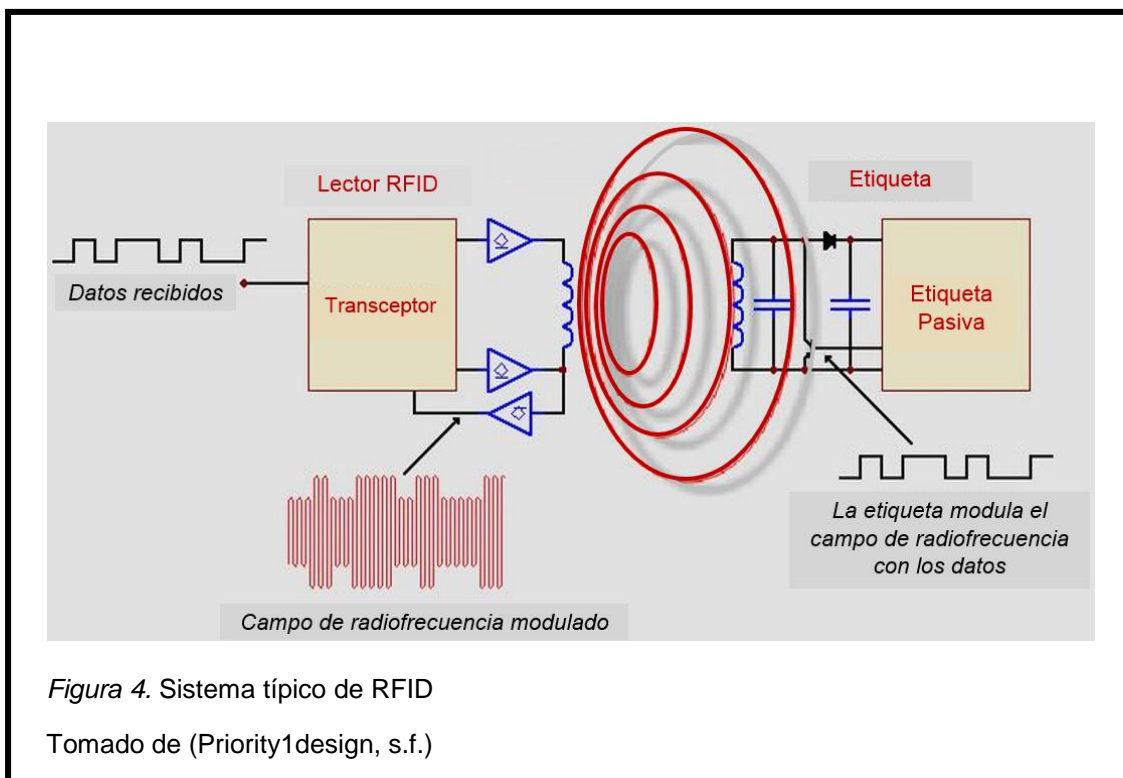
Tomado de (Ruiz, 2013)

La frecuencia se refiere al tamaño de las ondas de radio usadas para la comunicación entre los componentes del sistema RFID. Las ondas de radio se comportan de diferente manera en cada una de estas frecuencias (Figura 3.) con ventajas y desventajas asociadas con el uso de cada banda de frecuencias.



Si un sistema de RFID opera a una frecuencia baja, el rango de lectura será más corto y la velocidad de lectura de datos será más lenta, pero tendrá mayores capacidades para la lectura de cerca o sobre superficies de metal o líquidas. Por otro lado, si opera a una frecuencia alta, la velocidad de transferencia de datos será rápida con rangos de lectura más largos que en los sistemas de baja frecuencia, pero tendrán más sensibilidad a la interferencia de ondas de radio causada por líquidos y metales en el ambiente. (Impinj, s.f.)

A continuación se muestra la imagen de un sistema típico de identificación por radiofrecuencia en la que el lector se comunica con la etiqueta pasiva. En cada sistema RFID las etiquetas contienen información en bits que puede ser almacenada en formato digital binario.

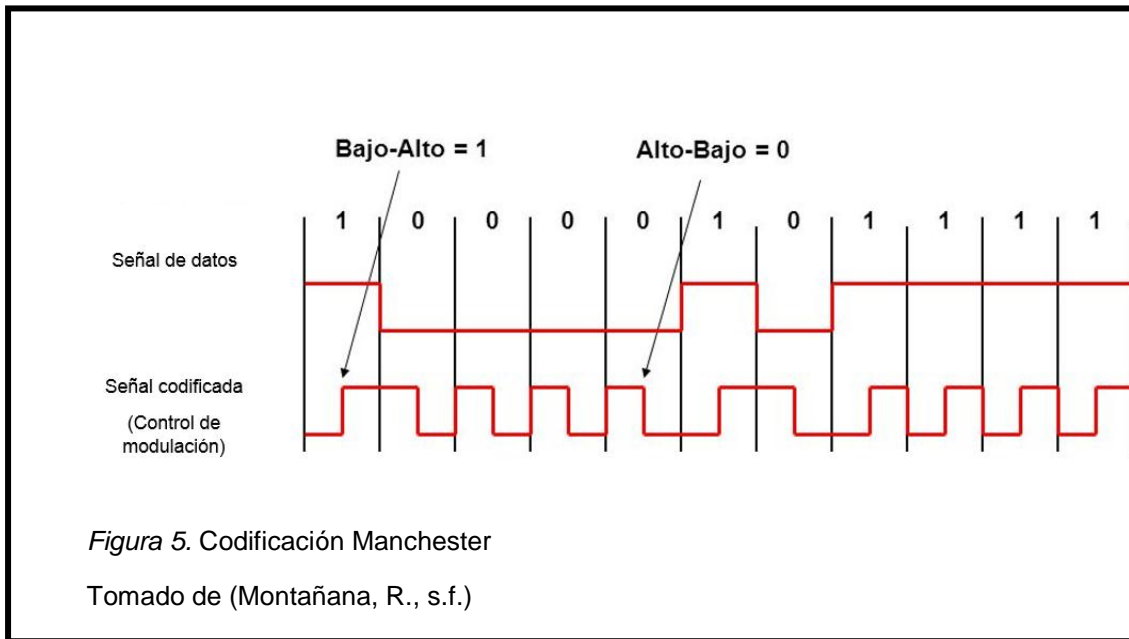


Cuando la etiqueta entra al campo de radiofrecuencia generado es capaz de extraer suficiente energía desde el campo para acceder a su memoria interna y transmitir su información almacenada. Al controlar la cantidad de energía del campo de RF la etiqueta puede modular el voltaje detectado en el lector (transceptor) según el patrón de bits que desea transmitir.

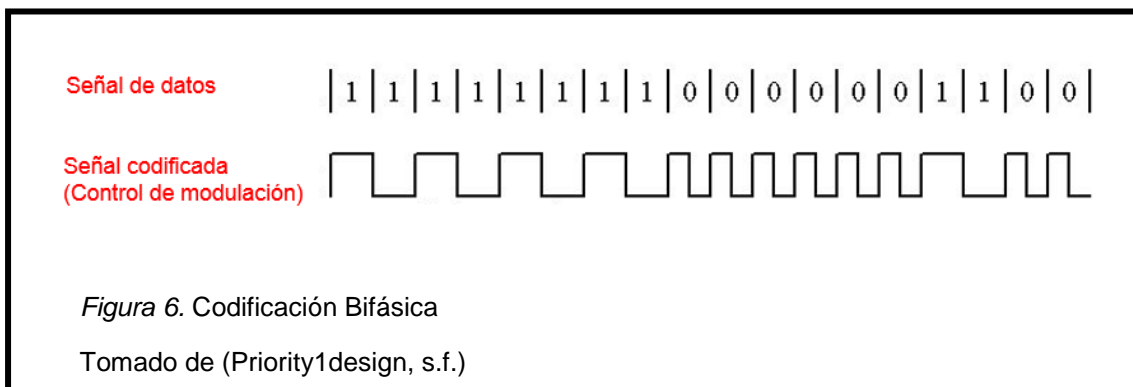
La etiqueta y el lector emplean los ciclos individuales del campo de radiofrecuencia (RF) para sincronizar la transmisión de datos entre los dos. La frecuencia del reloj de sincronización se convierte simplemente en la frecuencia del campo de RF utilizado. La duración de cada bit se especifica en términos de ciclo de reloj. (Priority1design, s.f.)

Existen 3 esquemas populares de codificación para la modulación:

- Esquema de codificación Manchester: es un método de codificación eléctrica binaria, en esta codificación la etiqueta produce un nivel de transición a la mitad del periodo de bits. Si la transición es de bajo a alto el estado lógico es 1 y si la transición es de alto a bajo el estado lógico es 0.

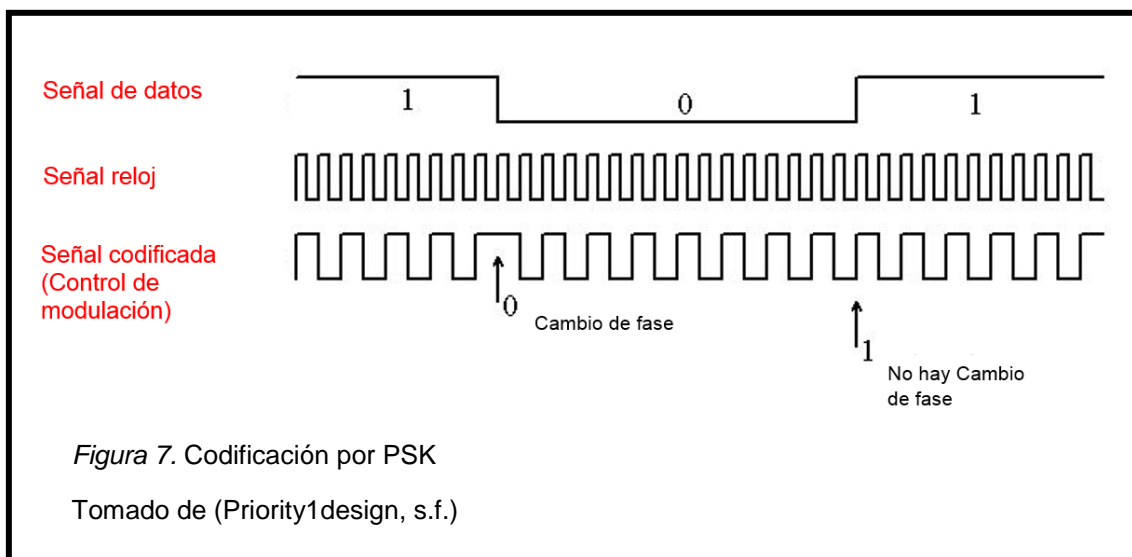


- Esquema de codificación bifásica: Modula el campo de radiofrecuencia para que exista una transición al inicio de cada límite de bit. Un estado lógico 0 tiene una transición en el medio del período del bit mientras que un estado lógico 1 no tiene transición durante todo el período del bit.



- Esquema de codificación por PSK (modulación por desplazamiento de fase): con esta codificación el campo de radiofrecuencia es modulado por lo que hay una transición con cada período de reloj, esto significa que puede haber hasta 64, 32 o 16 transiciones por bit dependiendo de la longitud de bit que esté usando la etiqueta.

Cuando se produce un cambio de fase se representa un estado lógico 0, mientras que un estado lógico 1 es interpretado cuando no hay cambio de fase en el límite del bit.



1.3 Arquitectura de un sistema RFID

Los sistemas RFID son producidos por muchos fabricantes y existen en innumerables variantes.

Su arquitectura consta principalmente de los siguientes componentes:

- Etiqueta
- Antena
- Lector
- Middleware/Software RFID

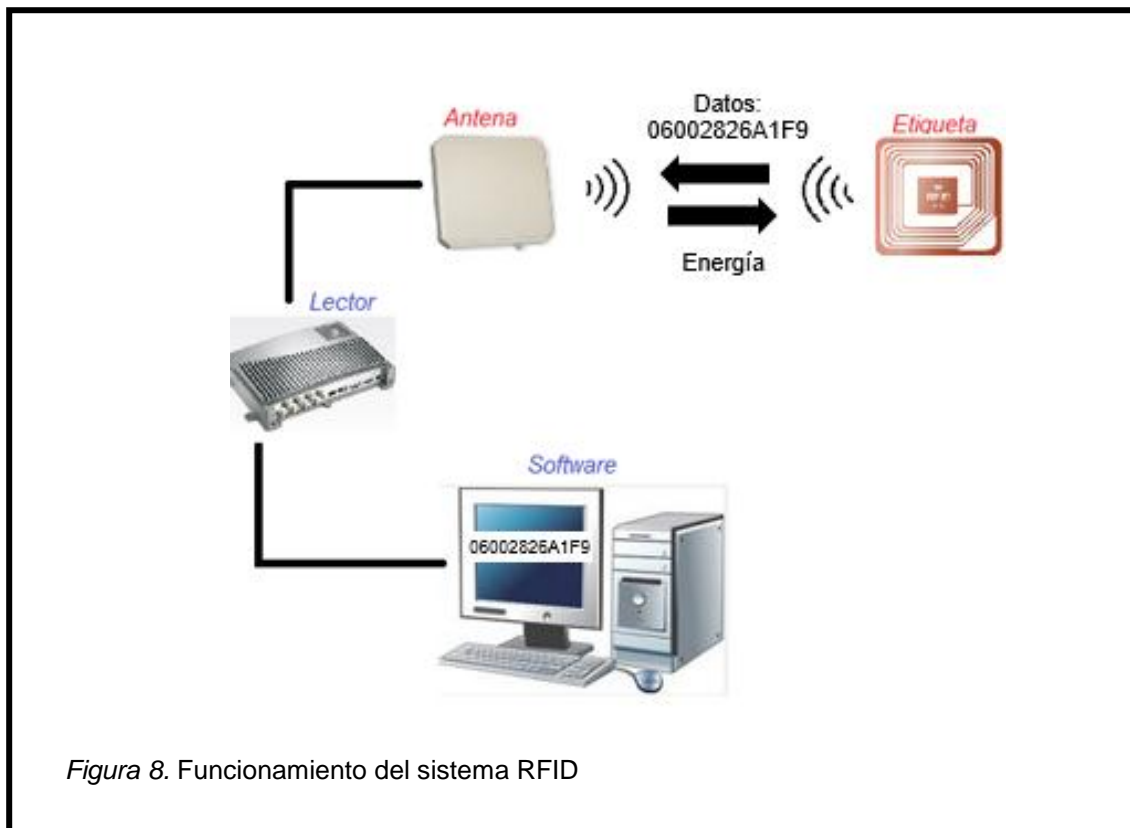
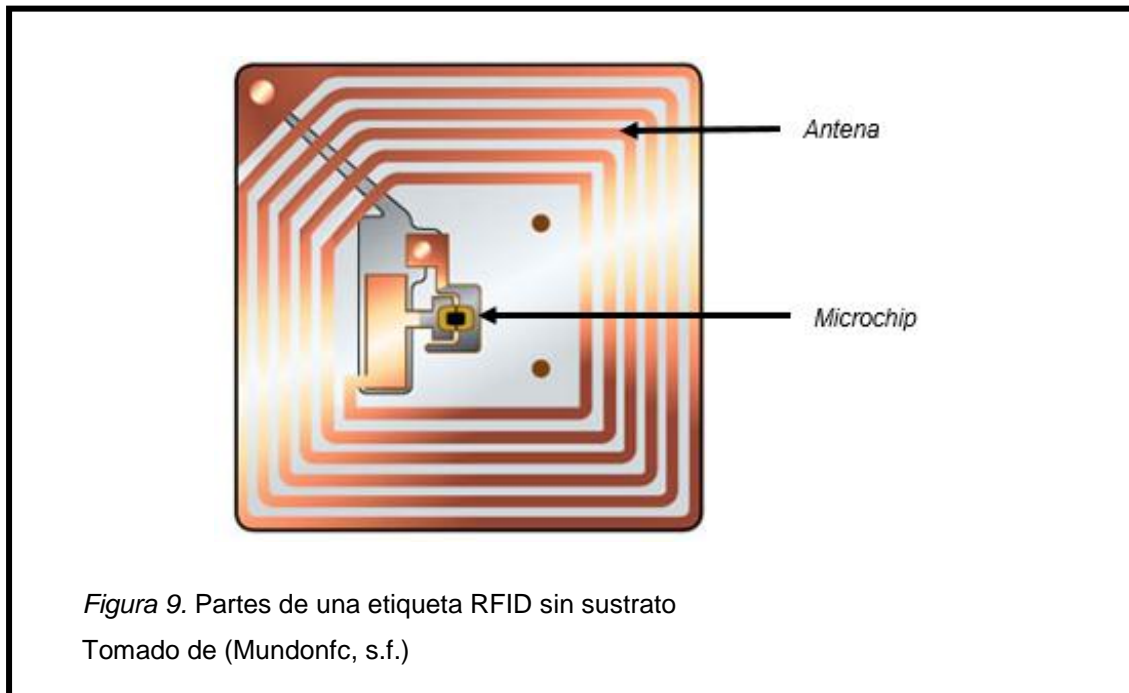


Figura 8. Funcionamiento del sistema RFID

En la Figura 8 la etiqueta permite que el objeto pueda identificarse enviando una señal para que ésta sea recibida por el lector. La antena emite un campo de radiofrecuencia, cuando entra en contacto con la etiqueta utiliza dicho medio para la transmisión de datos, el lector recibe la información enviada por la etiqueta y finalmente el software RFID almacena y procesa la información emitida por el lector.

1.3.1 Etiqueta RFID

Es el componente estrella del sistema RFID, también es conocido como transpondedor, *tag* o etiqueta RFID, es un pequeño artefacto que se le puede integrar a un objeto, animal o persona, está compuesto por un microchip que va adjunto a una antena de radio y sirve para identificar unívocamente al objeto portador de la etiqueta.



El microchip se encarga de almacenar un único número serial u otra información, esta información almacenada se define por sus características de lectura-escritura:

- Sólo lectura (*Read-only*): Solo son etiquetas de lectura, que vienen con un número identificación de fábrica y no es modificable o borrable.
- Solo una escritura (*Write-once*): El número de identificación de la etiqueta es de fábrica, al poseer la característica de una solo escritura, permite al usuario modificar dicho número una sola vez, una vez modificado es imposible cambiarlo.
- Escritura y Lectura (*Read-write*): Son etiquetas de lectura y escritura, se puede escribir y borrar varias veces el número de identificación.
- Anticolisión: Permite que la etiqueta sepa cuándo transmitir su información al lector para no molestar otras lecturas ya que cuando muchas etiquetas se encuentran próximas a un lector, éste puede tener problemas a la hora de comunicarse con ellas al mismo tiempo. Esta característica se realiza mediante protocolos basados en algoritmos bastante complejos que permiten controlar la comunicación entre la etiqueta y el lector. (Ruiz, 2013).

En general, los protocolos de las etiquetas anticolisión son algoritmos que se agrupan en dos categorías: protocolos ALOHA y protocolos de árbol binario.

Protocolos ALOHA: también se los conoce como probabilísticos o de acceso aleatorio, permiten la reducción en la probabilidad de colisiones desde que las etiquetas transmiten su información en distintos tiempos, se los utiliza en escenarios donde el lector no sabe exactamente cuando un *Tag* pasará por el área de interrogación. Las versiones de los protocolos ALOHA se dividen en 4 grandes grupos:

- ALOHA puro
- ALOHA ranurado (*slotted ALOHA*)
- ALOHA enmarcado ranurado (*frame slotted ALOHA - FSA*)
- ALOHA dinámico enmarcado ranurado (*dynamic frame slotted - DFSA*)

Protocolos basados en árbol binario: se basan en los algoritmos de árbol binario, el lector intenta reconocer un conjunto de etiquetas en varios ciclos de interrogación, en cada uno de estos ciclos, el lector envía una consulta conocida como "*Query*", si hay más de una etiqueta, el lector divide el conjunto en subconjuntos hasta encontrar una sola etiqueta por conjunto. (García, Escamilla, Álvarez y Mayoral, 2013, p. 5)

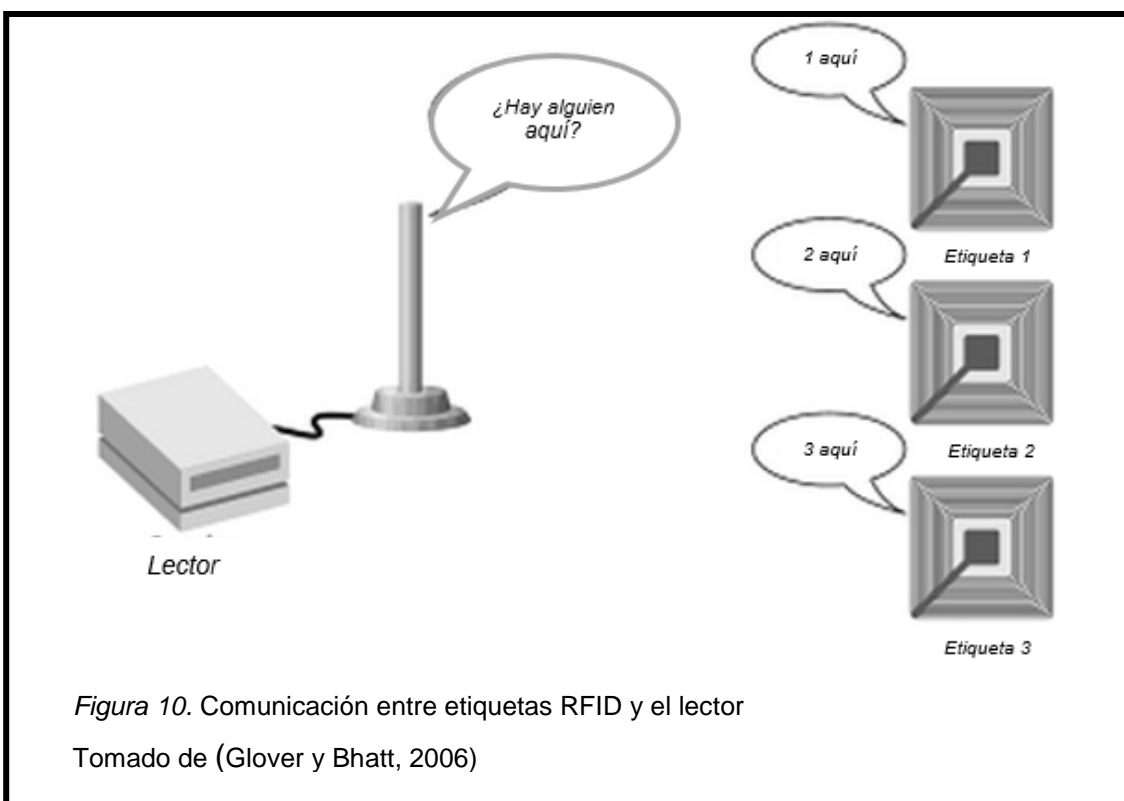
Este mecanismo no es eficiente cuando el número de etiquetas a reconocer es grande. Se dividen en dos grupos:

- Árbol de consulta (*Query Tree protocol*)
- Protocolos de árbol binario (*Binary Tree protocols*)

La antena de la etiqueta se encarga de transmitir señales de radio desde el microchip hacia el lector. Las antenas para este tipo de etiquetas deben ser diseñadas tanto para absorber la energía, como para transmitir la señal de respuesta.

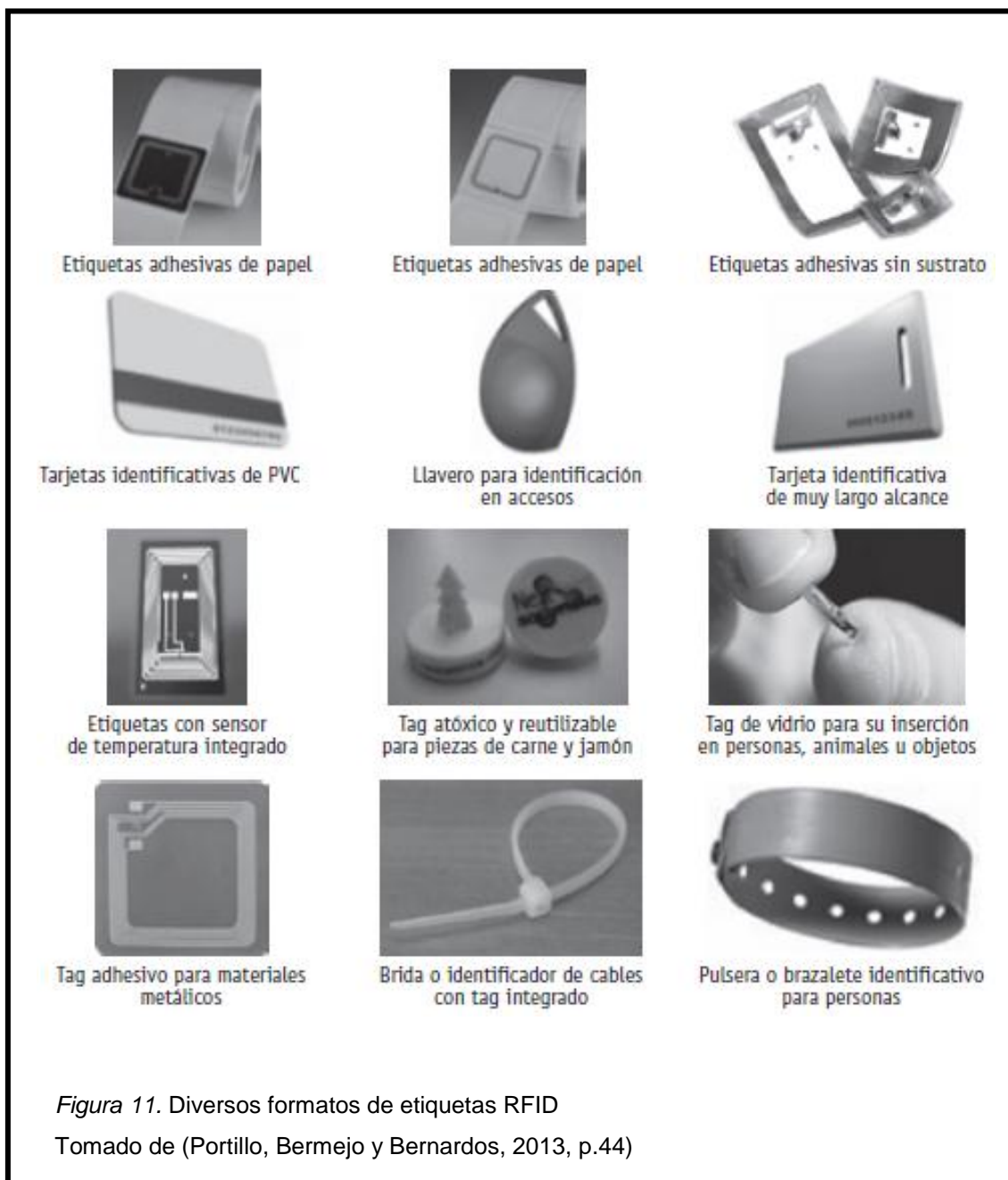
Finalmente el sustrato es el elemento encargado de que el microchip y la antena puedan estar en un solo dispositivo, permitiendo que se forme la etiqueta física.

La figura 9 explica el proceso de cómo una etiqueta notifica al lector acerca de su presencia e identidad. En esta figura, el lector transmite señales de radio a una frecuencia e intervalo preestablecido (generalmente ciento de veces cada segundo). Cualquiera de las etiquetas de radiofrecuencia que se encuentren en el rango de éste lector (Figura 2.) recogerán su transmisión debido a que cada una tiene una antena incorporada que es capaz de escuchar a las señales de radio a una frecuencia preestablecida. Las etiquetas utilizan la energía de la señal del lector para reflejar ésta señal de vuelta; pueden modular la señal para enviar información, como un número identificador, de regreso al lector. (Glover y Bhatt, 2006, p.34)



Las características de rendimiento de una etiqueta RFID depende de factores como: el tipo de microchip utilizado, la capacidad de lectura/escritura, la frecuencia de radio, la configuración de energía, medio ambiente, etc.

En la siguiente figura se puede observar una amplia variedad de formas existentes para las etiquetas RFID.



Existen 3 diferentes versiones de etiquetas: pasivas, activas y semi-pasivas.

Tabla 2. Comparación de los diferentes tipos de etiquetas.

Etiquetas pasivas	Etiquetas activas	Etiquetas semi-pasivas
<p>Tiempo de vida casi ilimitado (no necesitan una batería). Su precio es económico. Corto alcance de lectura (usualmente menos de 1 m.). Requiere de mayor potencia del lector.</p>	<p>Tiempo de vida limitado. Más costosas que las pasivas y semi-pasivas. Mayor rango de lectura que las pasivas y semi-pasivas (usualmente más de 10m.). Requiere de menor potencia del lector que las pasivas y semi-pasivas.</p>	<p>Mayor tiempo de vida que las etiquetas activas. Más costosas que las pasivas. Menor rango de lectura que las activas (menor a 10m.) Requiere mayor potencia del lector que las activas</p>

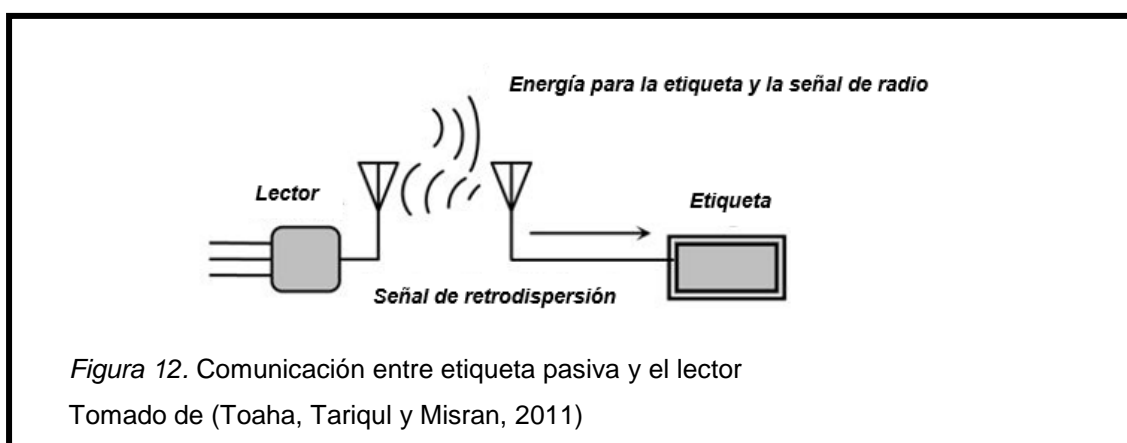
Tomado de (Prototypexpress, s.f.)

1.3.1.1 Etiquetas Pasivas

La etiqueta pasiva se caracteriza por carecer de una fuente de alimentación integrada, ésta se activa y envía su información cuando un lector RFID se encuentra cerca, ya que el lector le provee de energía suficiente para que se alimente.

La señal que recibe la etiqueta por parte de los lectores induce una pequeña y suficiente corriente eléctrica con la que trabaja el microchip de la etiqueta para generar y transmitir una respuesta.

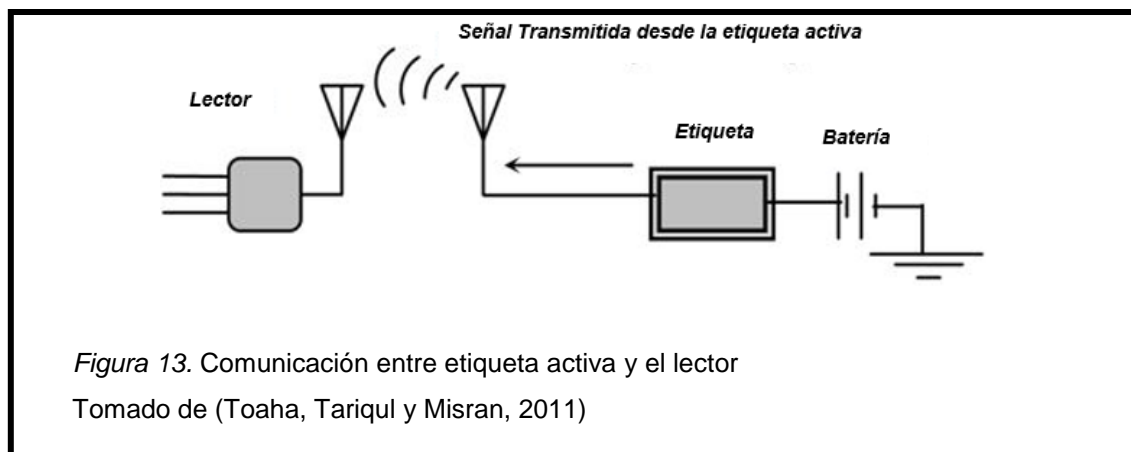
La etiqueta pasiva (Figura 12) utiliza la señal que recibe del lector para alimentar al microchip y para variar la reflexión de esta señal (retrodispersión) para poder transmitir la información de regreso al lector.



Estas etiquetas son las más comunes en aplicaciones sensibles al bajo coste ya que al no tener batería y ningún transmisor son muy económicas.

1.3.1.2 Etiquetas Activas

La etiqueta activa al contrario de la pasiva tiene su propia fuente de energía que utiliza para dar corriente a su microchip y transmitir su señal al lector. (Datamark, s.f.)



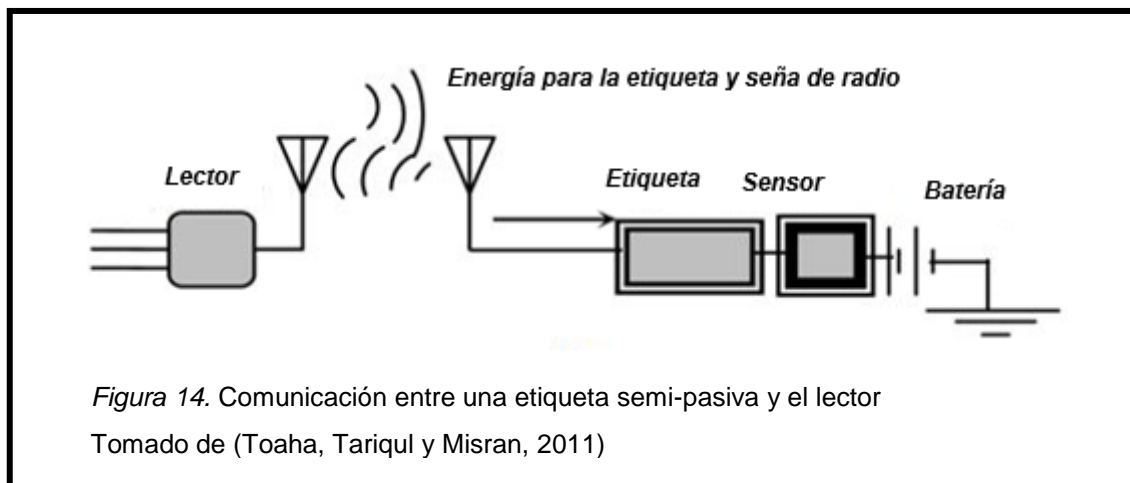
Las principales ventajas de este tipo de etiquetas son su rango de lectura y su fiabilidad, pues no necesitan una señal de radio continua para alimentar su microchip. En la figura 13 se observa la comunicación entre el lector y una etiqueta activa.

La distancia de estas etiquetas es mucho mayor a la de las pasivas, pueden transmitir señales de alto nivel y más potentes obteniendo respuestas claras a partir de recepciones débiles. Su precio es elevado y la vida útil es mucho más corta que la de las pasivas a causa del desgaste de la batería. (Datamark, s.f.)

1.3.1.3 Etiquetas Semi-Pasivas

Este tipo de etiquetas son conocidas como etiquetas pasivas asistidas por batería. Tienen una batería, así como las etiquetas activas, pero aún utilizan la energía del lector para transmitir de regreso la información al lector RFID usando una técnica conocida como retrodispersión. Por los tanto, las etiquetas

semi-pasivas tienen la fiabilidad de lectura de una etiqueta activa pero el rango de lectura de una etiqueta pasiva. También tienen una vida útil más larga que una etiqueta completamente activa.



Las etiquetas semi-pasivas tienen mayor alcance que las etiquetas pasivas y pueden ser usadas para monitorear/controlar las entradas de los sensores, incluso cuando estas no están en presencia de un campo de radiofrecuencia.

1.3.2 Antena

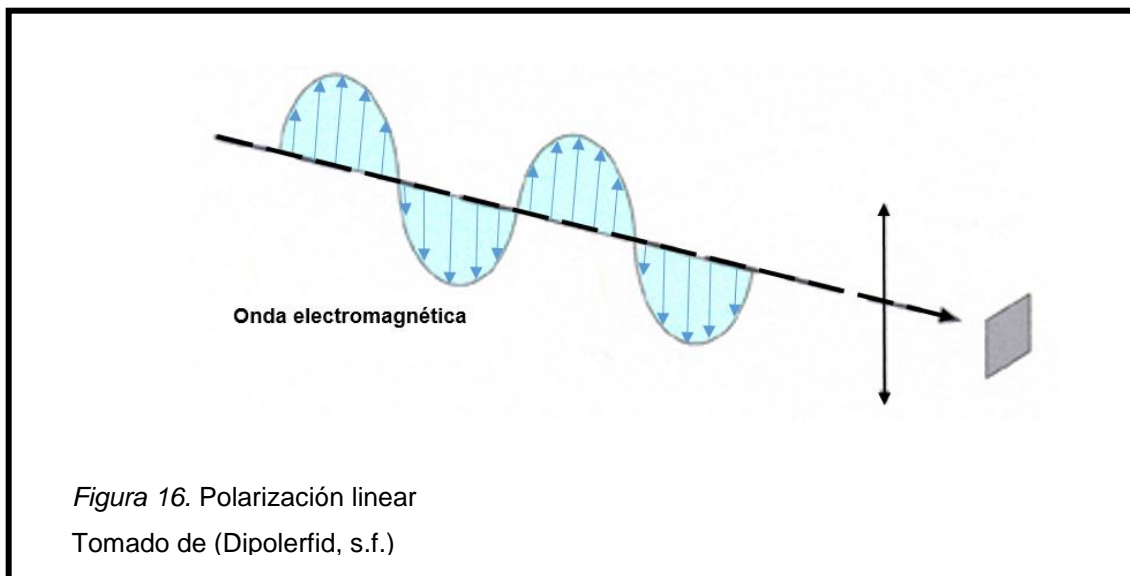
La Antena RFID es el elemento que permite la comunicación entre la etiqueta y el lector. Irradia las ondas electromagnéticas que son generadas por el lector y capta las señales de radiofrecuencia que devuelve la etiqueta. Hay casos en los que las antenas no vienen integradas al lector por lo que se las conectan a través de cables permitiendo su localización hasta 2 metros con respecto al lector.



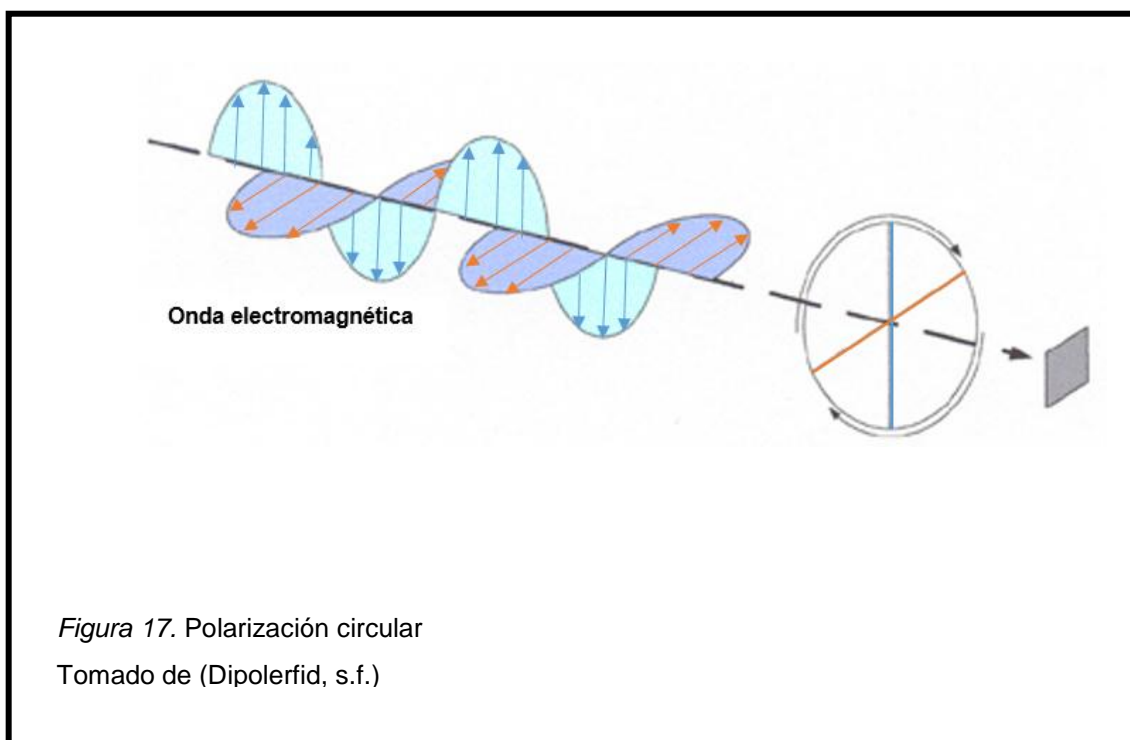
Los tamaños y diseños de las antenas son variados y diferentes, estos dependen tanto del entorno en el cual el sistema va a ser integrado así como de las necesidades y requerimientos propios para la aplicación a desarrollar.

Las 3 características fundamentales de las antenas RFID para legibilidad de las etiquetas son:

1. Rango de frecuencia: Frecuencias en las que opera la antena.
2. Ganancia y ancho de haz: Ambos son componentes eléctricos de una antena. A mayor ganancia menor será el ancho de haz. El ancho de haz está determinado por la ganancia, cuando mayor sea la ganancia más enfocado será el haz.
3. Polarización: Orientación de transmisión del campo electromagnético.
 - Polarización lineal (antenas dipolo): La onda electromagnética se propaga íntegramente en un solo plano vertical u horizontal en dirección de la señal de propagación. Esta es la mejor propagación de onda cuando la orientación de la etiqueta es conocida y fija. La antena RFID y etiqueta RFID deben coincidir en polarización para obtener los mejores índices de lectura. (Skyrfid, s.f.)



- Polarización circular (antenas hélice, dipolo cruzado y parche): La onda electromagnética se propaga en dos planos creando un efecto circular, haciendo una revolución completa en el período de tiempo de una sola longitud de onda. La polarización circular puede ser diestra o surda por lo tanto los RHCP y LHCP son opciones de antenas de polarización circular.



La funcionalidad de la antena depende en gran parte de la frecuencia de operación a la que trabaja el sistema.

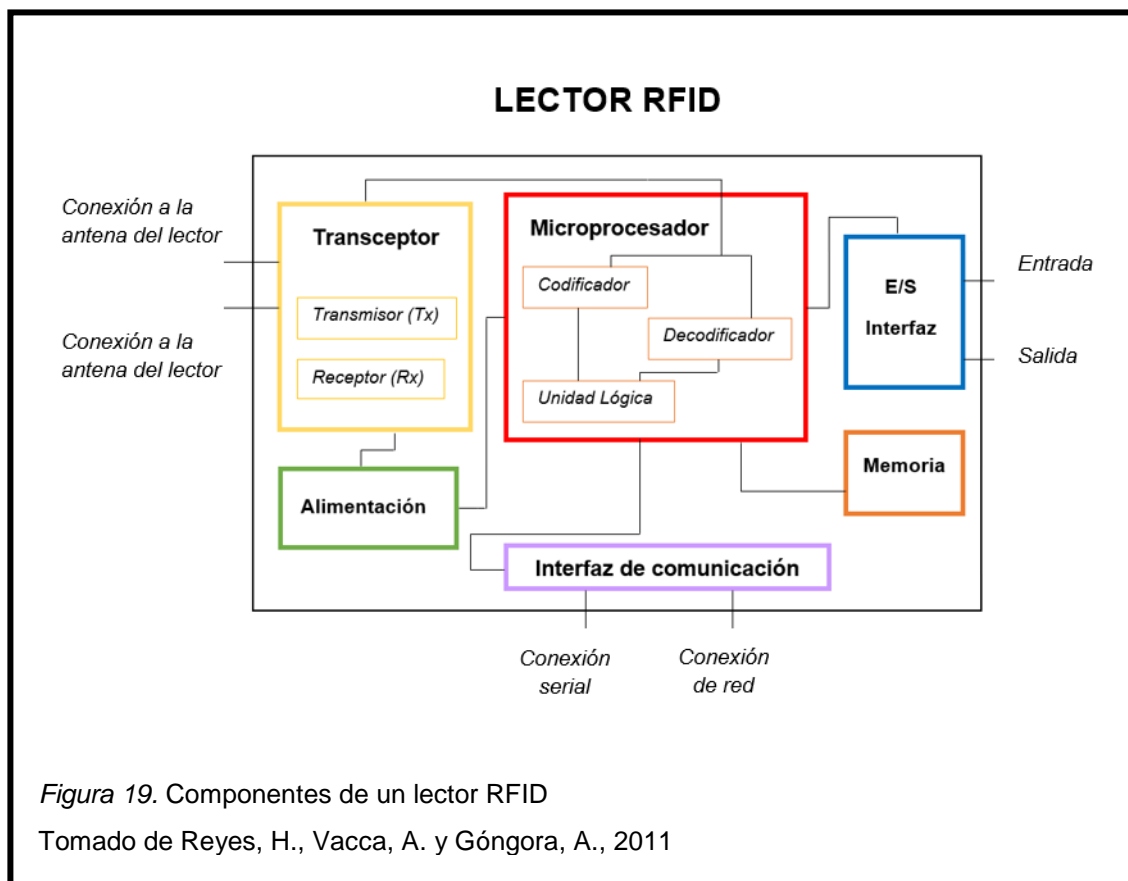
1.3.3 Lectores

El lector RFID es el dispositivo encargado de alimentar las etiquetas, una vez que recibe la información proporcionada por estas, los datos pasan por un proceso de decodificación y son transmitidos al software RFID para ser procesados y almacenados.



Figura 18. Lector RFID UHF FX-9500 Motorola
Tomado de (Instruelecom, s.f.)

Un lector está compuesto por un módulo de radiofrecuencia (consta de un transmisor y de un receptor), una unidad de control y una o más antenas que dependiendo del modelo y marca pueden ser móviles o estar integradas al lector.



La interface de comunicación del lector puede ser a través del puerto serie o por red. La comunicación del lector mediante la interfaz por puerto serie se lleva a cabo a través de la conexión con una PC mediante un módulo RS232 o RS485 dependiendo de las características propias del lector.

Por otra parte, la comunicación del lector a través de la interfaz por red se efectúa gracias a la conexión de un host por cable (Ethernet, WLAN) o inalámbricamente (WiFi, Bluetooth, etc.). Las dos interfaces de comunicación permiten que la información sea transmitida desde el lector hacia el Software RFID.

La frecuencia de operación debe ser ajustada a las normativas en el uso del espectro radioeléctrico determinado por el ente regulador de cada país.

1.3.3.1 Lectores Fijos

Los lectores fijos se encuentran ubicados en lugares específicos y estratégicos como puertas de acceso y pueden contar con una o más antenas; dependiendo de la cantidad de antenas se establece un área de lectura a cada una indistintamente.

1.3.3.2 Lectores Móviles

Los lectores móviles son dispositivos portátiles capaces de acompañar al operador a donde sea que vaya en su área de trabajo, tienen una pantalla LCD, una antena integrada dentro de una unidad portátil y un teclado para introducir datos.

1.3.4 Software RFID

Los componentes del software varían según la finalidad de la aplicación del sistema a desarrollar.

El primer componente requiere un conjunto de operaciones que permiten la conexión y funcionalidad del sistema, es decir, la lectura de la etiqueta al pasar por el lector, por lo que se lo denomina Software del Sistema, el segundo componente es el *middleware* el cual es el software encargado de la conectividad del sistema antes mencionado mediante el procesamiento y almacenamiento de la información que recibe por el lector RFID.

1.4 Clasificación RFID

La tecnología RFID se puede clasificar de acuerdo a las cuatro bandas de frecuencia que utiliza. La siguiente tabla contiene las diferentes aplicaciones en las que esta tecnología ha ido ocupando lugar.

Tabla 3. Características y ejemplos aplicables con las distintas bandas de frecuencia utilizadas en RFID.

Banda de Frecuencia	Características	Aplicación
LF (de 100 a 500 KHz.) Típico 125 a 134 KHz. Internacional	Corto Alcance. Poca velocidad de transmisión. Relativamente económico. Gran penetración en los materiales (líquidos). Trabaja bien junto a metales.	Control de acceso. Identificación de animales. Control de inventario. EAS (Antirrobo). Aplicaciones de seguridad para automóviles.
HF. Típico 13,56 MHz. Internacional	Corto/medio alcance. Velocidad de transmisión media. Puede leer a través de líquidos y en entornos húmedos. Problemático junto a metales. Moderadamente caro.	Control de acceso. Tarjetas inteligentes. EAS (Antirrobo). Inventario en bibliotecas. Gestión de almacén. Control de equipajes. Gestión de lavanderías. Identificación de pacientes.
UHF (de 400 a 1000 MHz.) Típico de 850-950 MHz.	Largo alcance. Alta velocidad de transmisión. Mecanismos de anticollisión. Problemático con líquidos y metales. Problemático en entornos húmedos. En metal genera interferencias. Moderadamente caro.	Gestión de artículos. Gestión en la cadena de suministros. Gestión de almacén. Gestión de expediciones. Trazabilidad.
Microondas de 2,4 a 5,8 GHz.	Medio alcance. Características similares a las etiquetas UHF pero con mayor velocidad de transmisión. Mayor costo.	Control ferroviario. Peajes de autopista. Localización.

Tomado de (Ruiz, 2013)

“La frecuencia de trabajo de la etiqueta y de los lectores condiciona las características físicas de propagación del campo electromagnético y, por lo

tanto, las de la transmisión de los datos: tipo de acoplamiento, distancia máxima de lectura, velocidad de transmisión, sensibilidad a los materiales. Estas características condicionan las aplicaciones comerciales para las que está destinada la tecnología RFID.” (Libera, 2010, p.7)

1.5 Regulación y Estandarización

El espectro radioeléctrico es un bien público y cada país tiene un ente encargado de regularlo con la descripción de las bandas de frecuencia que lo integran dentro de su territorio, por lo que el Ecuador no es la excepción.

Cada país puede establecer sus propias normas para la asignación de frecuencias para las etiquetas RFID, teniendo en cuenta que no todas las bandas de radio se encuentran disponibles en todos los países.

Existen dos grandes organismos internacionales para la estandarización o normalización RFID - ISO (Organización Internacional de Normalización) y *AutoID Centre* que actualmente es manejado por EPC Global (*Electronics Product Code Global Incorporated*).

1.5.1 ISO

ISO es una de las más antiguas organizaciones para estandarización y es el organismo líder mundial en materia de desarrollo de normas internacionales. En el año de 1996 se creó una comisión conjunto con IEC (*International Electrotechnical Commission*- Comisión Electrotécnica Internacional).

Las normas ISO RFID son divididas en las siguientes categorías:

- Protocolo de interfaz de aire y protocolos asociados: modo de comunicación vía radiofrecuencia entre las etiquetas y lectores.
- Contenido y formato de datos: organización de los datos que se comunican entre etiquetas y lectores.

- Pruebas de conformidad: Pruebas de que los productos cumplen con la norma.
- Aplicaciones: Cómo las aplicaciones con RFID son usadas.

ISO ha publicado las normas técnicas fundamentales para la identificación instantánea y gestión de objetos que se aplican en RFID. Una vez desarrolladas, estas normas se diseñan para su utilización en todo el mundo.

Normas de la serie ISO 18000

La ISO 18000 es una serie de normas que definen la interfaz de aire para las diferentes frecuencias de RFID en uso en todo el mundo.

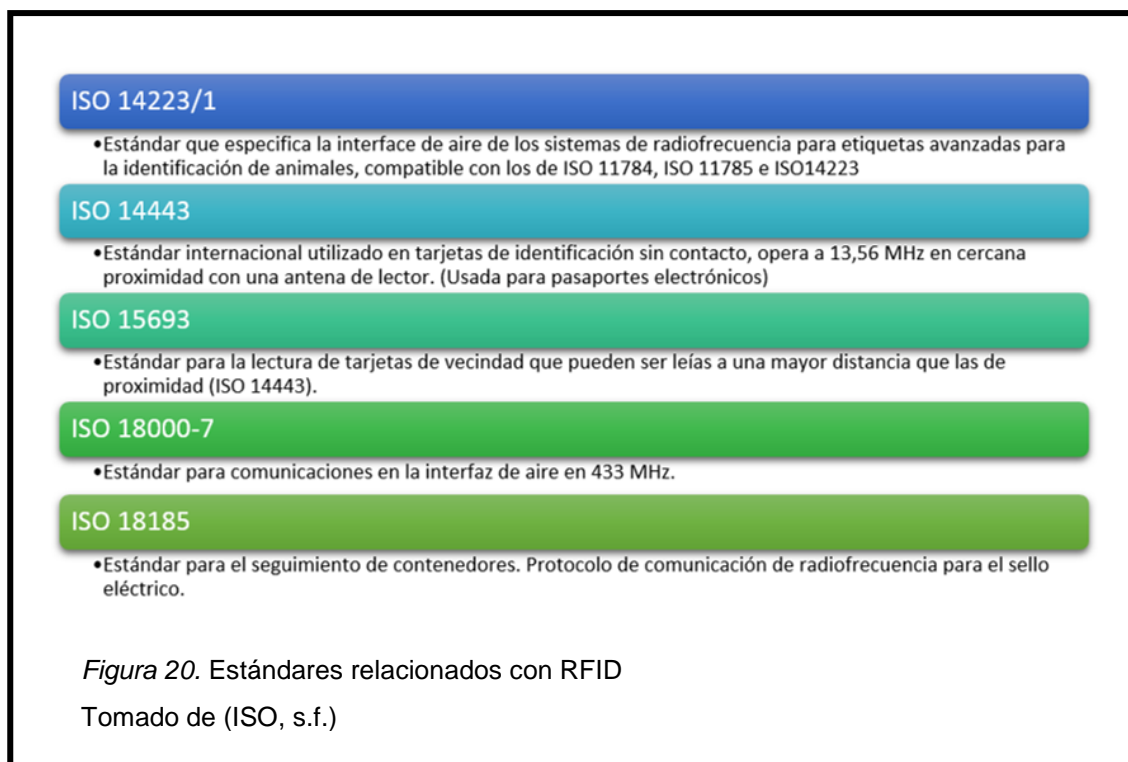
Dentro de esta serie hay un total de 7 normas.

Tabla 4. ISO- Serie 18000

Norma ISO 18000	Frecuencias
ISO 18000- Versión 1	Parámetro genérico para interfaces de aire para frecuencias mundialmente aceptadas.
ISO 18000-Versión 2	Interfaz de aire para 135 KHz
ISO 18000-Versión 3	Interfaz de aire para 13.56 MHz
ISO 18000-Versión 4	Interfaz de aire para 2.45 GHz
ISO 18000-Versión 5	Interfaz de aire para 5.8 GHz
ISO 18000-Versión 6	Interfaz de aire de 860 MHz a 930 MHz. RFID UHF Gen 2
ISO 18000-Versión 7	Interfaz de aire en 433.92 MHz

Tomado de (Libera, 2010, p.10)

Existen algunos estándares relacionados con RFID: ISO 10536, ISO 14443 e ISO 15693, pero la serie de estándares precisamente relacionada con Tecnología RFID y las frecuencias empleadas en dichos sistemas es la serie 18000.



1.5.2 EPC GLOBAL

EPCglobal es una organización que lidera el desarrollo de estándares de la industria en todo el mundo por las normas de la tecnología del Código Electrónico de Producto (EPC).

EPCglobal es una iniciativa de GS1 para innovar y desarrollar los estándares manejados por la industria para el código electrónico de producto (EPC) para apoyar el uso de la identificación por radiofrecuencia (RFID) y permitir la visibilidad global de artículos (EPCIS) en el rápido movimiento de hoy, rico en información y redes de comercio.

EPC (Código electrónico de producto), es un número único que mediante la tecnología RFID se encuentra almacenado en una etiqueta de radiofrecuencia permitiendo identificar cada producto de manera única; para su lectura emite señales de radio a dispositivos especiales que las reciben y las decodifican sin

necesidad de que el producto sea manipulado. Mediante la etiqueta se puede conocer la información detallada de cada producto.

El EPC está diseñado para incluir distintos sistemas de codificación. El más utilizado es el Sistema de Codificación Estándar GS1

TDS es el estándar de datos de etiqueta EPC de GS1, este especifica el formato de datos del código electrónico de producto y proporciona codificaciones para los esquemas de numeración incluyendo las claves de GS1 dentro de un EPC.

El Ecuador se rige a las normas de EPC, su representante es GS1 Ecuador (ECOP) la cual es la representación de GS1.

1.5.2.1 EPC Gen 2 clase 1

EPC Global de GS1 publicó el estándar Gen 2 clase 1 (EPC Gen 2) en 2004 para los protocolos de interfaz de aire, el cual define los requerimientos físicos y lógicos para un sistema RFID de lectores y etiquetas pasivas, operando en el rango UHF de 860 MHz-960 MHz. En la última década, EPC Gen2 se ha establecido como un estándar ISO con validez mundial para implementaciones UHF en múltiples sectores y es el corazón de cada vez más implementaciones de RFID, quedando reflejado finalmente como ISO 18000-6c.

El estándar Gen 2 (ISO 18000-6c) fue diseñado basado en requerimientos de desempeño y otros aportes de la comunidad de usuarios finales, incluyendo los minoristas y fabricantes de bienes de consumo. Está destinado a permitir a socios comerciales de la cadena de suministro codificar datos e interrogar los EPCs de etiquetas pasivas UHF de manera similar, así que podría compartir una infraestructura interoperable de interrogatorio y software.

2. CONSTRUCCIÓN DEL PROTOTIPO

2.1 Funcionamiento actual de la bodega de BiciQ

BiciQ tiene bajo su responsabilidad 658 bicicletas distribuidas en 25 estaciones ubicadas en el hipercentro de Quito para lo cual cuenta con pequeñas bodegas cercanas a las estaciones donde se las guardan al momento de cerrar el servicio que ofrece y una bodega principal que sirve como taller.

El personal encargado de repartir las bicicletas en cada estación se dirige a las bodegas a horas muy tempranas de la mañana ya que su servicio empieza desde las 7:00am.

El funcionamiento es muy sencillo, las bicicletas son recogidas a las 7am y se las distribuyen en cada una de las estaciones donde funciona BiciQ, de igual manera a las 7pm son retiradas de las estaciones y guardadas en las bodegas.

En las estaciones se encuentra el personal encargado de repartir las bicicletas a los usuarios así como otros colaboradores quienes son denominados “centinelas” los cuales son personas encargadas de revisar los daños menores y mayores de las bicicletas en todas las estaciones al momento de ser devueltas por los usuarios a lo largo de la semana, es decir, durante toda la semana van revisando el estado de las bicicletas en cada estación.

Los centinelas manejan diariamente un reporte de Mantenimiento de bicicletas donde detallan la bicicleta a reparar, el problema que ha sufrido y si el problema ha sido resuelto o no; si el nivel de gravedad es bajo reparan la bicicleta en la estación y si es alto los centinelas llenan el reporte explicando el motivo por el cual no se ha podido reparar la bicicleta para que sea llevada a la bodega principal la cual se encuentra ubicada en el Parque de la Carolina.

En la bodega principal llegan todas las bicicletas que no han podido ser reparadas por los centinelas y también las que se encuentran en buen estado,

en esta bodega se cuenta con un grupo de 4 mecánicos encargados de realizar los reemplazos de piezas y el mantenimiento adecuado para que la bicicleta pueda seguir usándose por los usuarios. De igual manera, llenan un reporte que registra todo lo que se le ha realizado a la bicicleta, todo este procedimiento se lo realiza manualmente por lo que la probabilidad de error se hace más alta especialmente en el control de inventario.

Actualmente la bodega de BiciQ no cuenta con un sistema automatizado que permita tanto el control de inventario que utilizan en la Bodega de la Carolina así como el control de entrada y salida de las bicicletas BiciQ que llegan a lo largo de la semana ya sea para reparar o para guardar.

2.2 Tecnología a utilizar

El control de entrada y salida de bicicletas en bodega BiciQ es un proceso netamente importante ya que garantiza el servicio que se brinda a través del control y de la gestión de stocks, este determina el estado en el que se encuentran las bicicletas al ingresar y salir al taller.

Tomando en cuenta el funcionamiento actual de la bodega de BiciQ y las problemáticas asociadas a este tales como la carencia de un control automatizado de las bicicletas que ingresan y salen y la falta de un control de inventario, es necesario que la tecnología a utilizar sea capaz de identificar estas bicicletas automáticamente de manera que se administre la información de forma inmediata y a tiempo real. La tecnología que garantiza este control se la conoce como RFID o identificación por radiofrecuencia.

2.2.1 Ventajas de la tecnología RFID frente al Código de Barras

La tecnología por identificación de radiofrecuencia presenta grandes ventajas frente a otras tecnologías de identificación automática como lo es el código de barras.

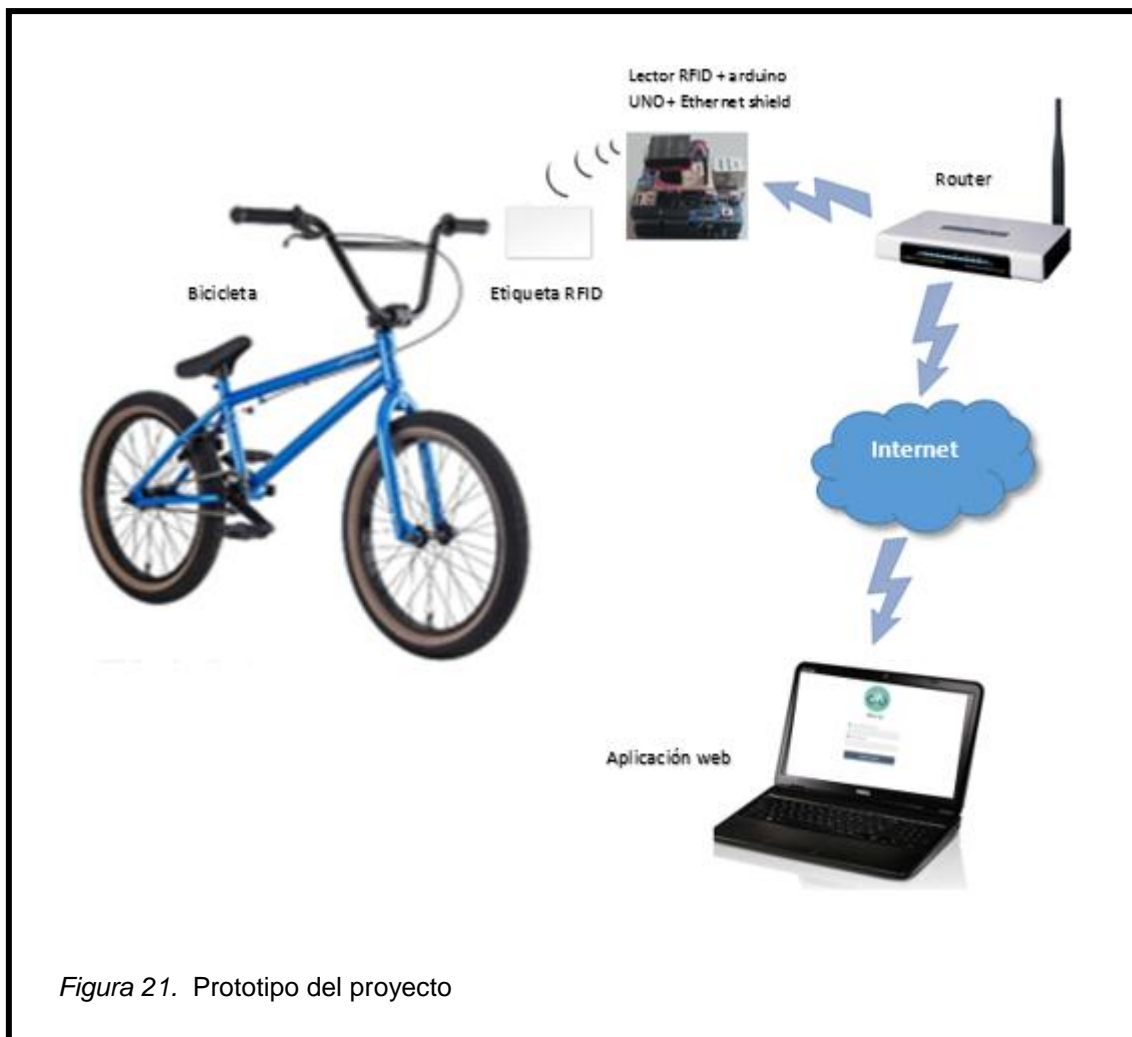
Entre las principales ventajas están las siguientes:

- Puede leer etiquetas RFID desde una mayor distancia que los códigos de barra, los códigos de barra requieren de un recurso humano para ser leído.
- Las etiquetas RFID no necesitan ser colocadas en una línea de visión con el escáner mientras que la lectura de código de barras sí requiere la línea de vista.
- La lectura de las etiquetas RFID es mucho más rápida que las de código de barra y pueden ser leídas varias al mismo tiempo. Dependiendo de las condiciones en que se encuentre el código de barras, puede dificultarse la lectura provocando errores y pérdida de tiempo.
- La lectura de la información de las etiquetas puede ser de largo alcance, dependiendo de las características del lector.
- La etiqueta puede soportar un ambiente hostil.
- A través de la tecnología RFID se realiza seguimiento de personas, animales o cosas en tiempo real, mientras que con los códigos de barras los procesos no se pueden automatizar debido a que puede variar la posición y la lectura sería nula.
- Las etiquetas son duraderas y reutilizables.

Debido a que el sistema ofrecido busca realizar un control en el seguimiento de inventario, se ha optado por la tecnología RFID y por la gran variedad de ventajas que posee ya mencionadas, cabe recalcar que dependiendo de la aplicación a realizarse, cada sistema sea RFID o código de barras tiene sus ventajas y desventajas.

2.3 Esquema del prototipo

El esquema del prototipo a desarrollar requiere de una bicicleta o varias, etiquetas (según el número de bicicletas a utilizar), el lector RFID, un cable USB, un cable de red, una PC o laptop y el software RFID.



El lector RFID trabaja a una frecuencia de 125kHz y su rango aproximado de lectura es de hasta unos 12 cm, las etiquetas RFID a usar son las tarjetas identificativas de PVC las cuales son pasivas por lo que no requieren de batería, operan a una frecuencia de 125kHz y sirven perfectamente para el prototipo a presentar. El tipo de codificación de datos de las etiquetas utilizadas en el prototipo: Manchester, PSK, FSK, NRZ o bifásico.

El lector a través de comunicación por radiofrecuencia lee el código de la etiqueta el cual es entregado en formato serial. El arduino es programado de tal forma que pueda capturar el código serial enviado por el ID-12LA y con el *Ethernet Shield* podrá conectarse a Internet.

2.4 Componentes del Hardware

El hardware de la tecnología RFID recibe la información que es enviada por la etiqueta y la transforma en bits. Los tres componentes principales utilizados para el funcionamiento del hardware son: el módulo ID-12LA, el Arduino y el *Ethernet Shield*.

Los elementos requeridos para su diseño son los siguientes:

- Simulador electrónico Proteus 7.8
- 1 Lector ID-12LA *Innovations*
- 1 placa para el lector ID12LA
- 4 Etiquetas pasivas (tarjetas identificativas de PVC)
- 1 Led
- 1 *Buzzer*
- 2 Resistencias
- 1 Transistor 2N3904 NPN 40V
- 3 Conectores
- 1 Arduino Uno
- 1 *Ethernet Shield*
- 1 Cable USB para impresora
- 1 Cable de red

2.4.1 Módulo lector ID-12LA

Para la construcción del prototipo se seleccionó el módulo lector ID12-LA ya que sus características operacionales se acoplan perfectamente al sistema propuesto, destacando su accesibilidad en el mercado debido a su bajo costo por lo que es perfecto para quienes optan por realizar proyectos a baja frecuencia.

Es un módulo RFID de pequeño tamaño que posee una antena integrada y es de fácil uso; se alimenta el módulo y al aproximarle una etiqueta RFID, se obtiene el código de identificación a través del puerto serie. (Bricogeek, s.f.)

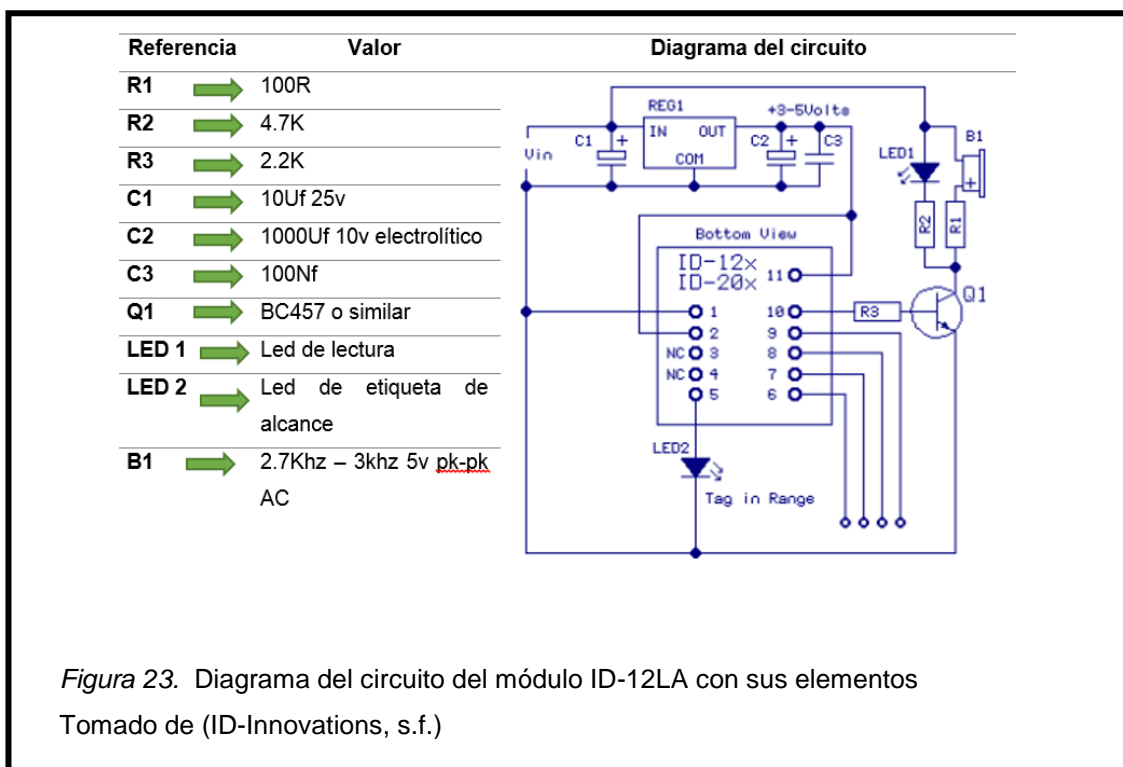
Características operacionales

- Alimentación: 2.8 – 5V
- Frecuencia de lectura: 125kHz
- Compatible con EM4001 64-bit RFID tag
- Conexión serie: 9600bps TTL y RS232
- Salida de emulación de banda magnética
- Distancia de lectura: 120mm
- Dimensiones: 24x26mm



Figura 22. Características operacionales del lector RFID ID-12LA
Tomado de (Bricogeek, s.f.)

Mediante el siguiente diagrama del circuito del módulo ID-12LA se observa de manera gráfica y simple los diferentes componentes que lo conforman.

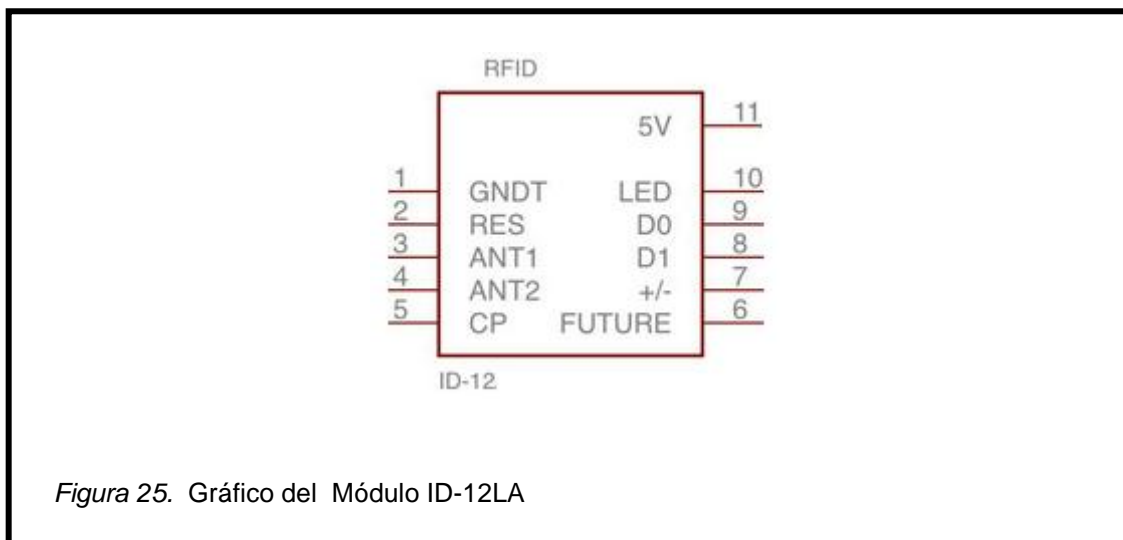


Para el adecuado uso y conexión del lector ID-12LA se detalla la vista inferior del dispositivo con la respectiva funcionalidad de cada uno de sus pines.

PIN	Descripción	ASCII	Emulación Magnética
PIN 1	→ GROUND / Cero voltios	GND 0V.	GND 0V.
PIN 2	→ Reset Bar	Barra de reinicio	Barra de reinicio
PIN 3	→ ANT(Antena)	Antena	Antena
PIN 4	→ ANT (Antena)	Antena	Antena
PIN 5	→ Card Preset (salida de emulacion de banda magnetica)	Sin función	Tarjeta preestablecida
PIN 6	→ Tag in range (etiqueta en el rango)	Etiqueta en el rango	Etiqueta en el rango
PIN 7	→ Format selector (+/-)	Atado a tierra (GND)	Atado al pin 10
PIN 8	→ Data 1	CMOS	Clock* (Reloj*)
PIN 9	→ Data 0	Datos TTL (invertida)	Datos*
PIN 10	→ Read (LED/Beeper)	Beeper/Led	Beeper/Led
PIN 11	→ DC suministro de voltaje	+2.8-5 V	+2.8-5 V

Figura 24. Descripción de los pines del módulo RFID ID-12LA y formato de datos de salida
Tomado de (ID-Innovations, s.f.)

A continuación se muestra el gráfico con los pines del módulo lector ID-12LA.



- El pin1 es el pin de suministro de cero voltios y la conexión a tierra más común en las comunicaciones.
- El pin 2 debe estar atado en todo momento al pin 11.

- El pin 3 se lo conecta sólo al usar el módulo ID-2LA y debe estar conectado a una antena externa de 1.33mH. En el módulo ID-12LA se lo debe dejar vacío.
- EL pin 4 se lo conecta únicamente al usar el módulo ID-3 y debe estar conectado a una antena externa. Debe dejárselo vacío con los módulos ID-13 Y ID-23.
- El pin 5 se lo utiliza únicamente como una tarjeta preestablecida de salida, cuando el formato de salida se establece en emulación magnética.
- El pin 6 es usado como un indicador en el rango de etiqueta. Cuando una etiqueta está en el rango el pin se ajusta en voltaje VDD de lo contrario está en 0V. El pin 6 de salida tiene una resistencia interna de 3.3K y puede ser usada para conducir un led directamente.
- El pin 7 es el selector de formato de salida de datos, el cual depende de donde está conectado (Figura 24).
- El pin 8 tiene funciones alternas. Cuando el formato de salida se lo establece en emulación magnética, el pin 8 es usado como el "Reloj". El modo alternativo está activo cuando el formato de salida se lo establece a ASCII y así el pin 8 envía los datos en ese formato. El pin 8 puede ser utilizado para conectar a una PC de entrada serial RS232.
- El pin 9 está activo cuando el formato de salida se lo establece en ASCII y este emite una salida complementaria (invertida) de salida de datos.
- El pin 10 es el pin indicador de salida de led o de sonido.
- El pin 11 es el pin de suministro de corriente continua.

2.4.2 Arduino UNO

Existe muchos otros microcontroladores disponibles en el mercado, para el desarrollo del prototipo se optó por el Arduino Uno debido a las ventajas que tiene sobre otros microcontroladores.

Entre las ventajas principales cabe destacar que es una plataforma de código y hardware abierto, es fácil de programar pues su entorno de programación es

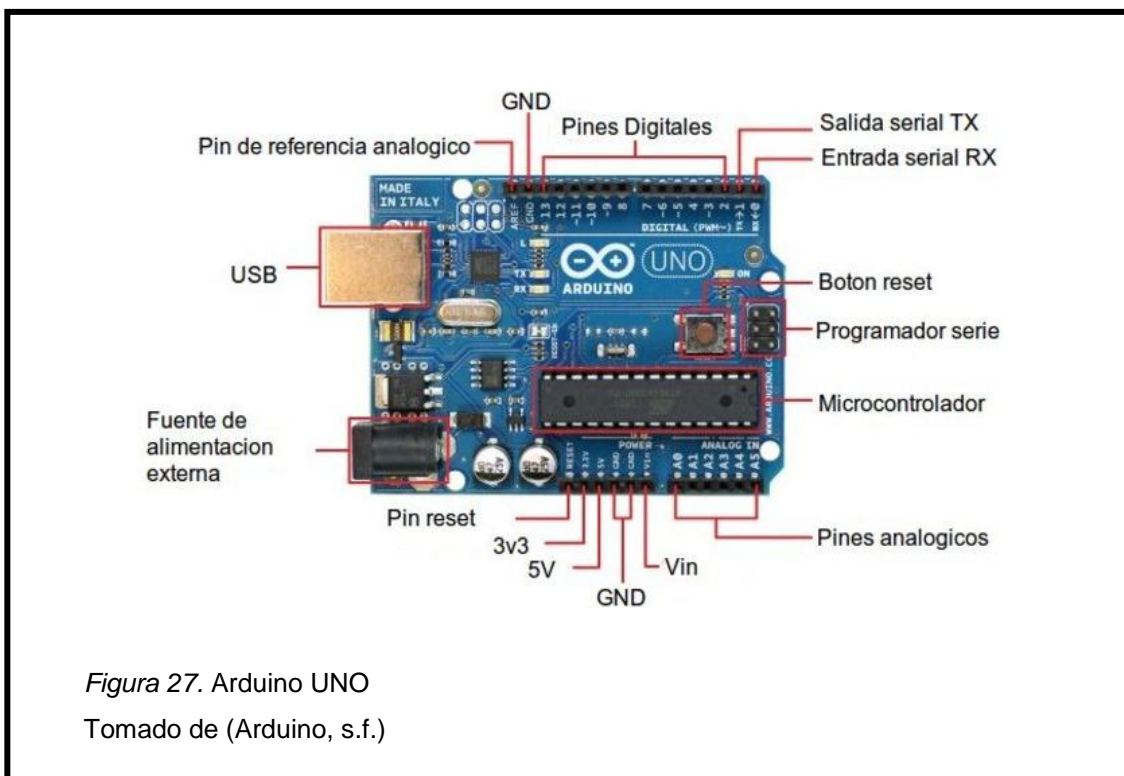
simple y directo, existen gran variedad de tutoriales que explican su uso, posee librerías para componentes externos que se requieran acoplar y es muy accesible por su costo.

El uso del Arduino permite que el lector no requiera estar conectado a un computador para que se efectúe la lectura de las etiquetas al pasar cerca de él. El arduino UNO es una placa electrónica que se basa en microprocesador ATmega328, posee una memoria flash a la que se le puede cargar un programa por USB según el fin para el cual se la requiera usar; tiene 14 pines digitales de entrada/salida en los que 6 pueden ser usados para salidas PWM (modulación por ancho de pulso), 6 entradas analógicas, un resonador cerámico de 16MHz, una conexión USB, un conector para una fuente de alimentación externa, una cabecera ICSP y un botón de reinicio.

Se lo puede alimentar a través de conexión USB a la PC o con una fuente de alimentación externa entre 9-12V.

Características Operacionales
<ul style="list-style-type: none"> • Microcontroladores: ATmega328 • Tensión de Funcionamiento: 5V • Voltaje de entrada (recomendado): 7-12V • Voltaje de salida (limites): 6-20V • Pines Digitales I/O: 14 (6 proporcionan salida PWM) • Botones de entrada analógica: 6 • Corriente DC por el pin E/O: 40 mA • Corriente DC para el pin de 3.3V: 50mA • Memoria: 32KB (ATmega328) aunque 0,5 KB se utiliza el gestor de arranque. • SRAM: 2KB (ATmega328) • EEPROM: 1KB (ATmega328) • Velocidad de Reloj: 16MHz • Largo: 68,6 mm • Ancho: 53,4 • Peso: 25g

Figura 26. Características operacionales del arduino UNO
Tomado de (Arduino, s.f.)



Los elementos que conforman los sistemas basados en Arduino: hardware, software y documentación, son de libre acceso y de código abierto. (Arduino, s.f.)

2.4.3 Arduino Ethernet Shield

El Arduino *Ethernet Shield* permite a una placa arduino conectarse a Internet, para usarla se debe simplemente montarla sobre la placa de arduino. Está basado en el chip *ethernet Wiznet W5100* que proporciona una red (IP) capaz de almacenar tanto TCP como UDP, soporta hasta 4 conexiones de sockets simultáneamente, utiliza la librería de Ethernet para escribir programas los cuales se conectan a internet usando la *shield* y tiene un conector Ethernet RJ-45 estándar con un transformador de línea integrada y una alimentación a través de Ethernet (PoE) habilitado. (Arduino, s.f.)

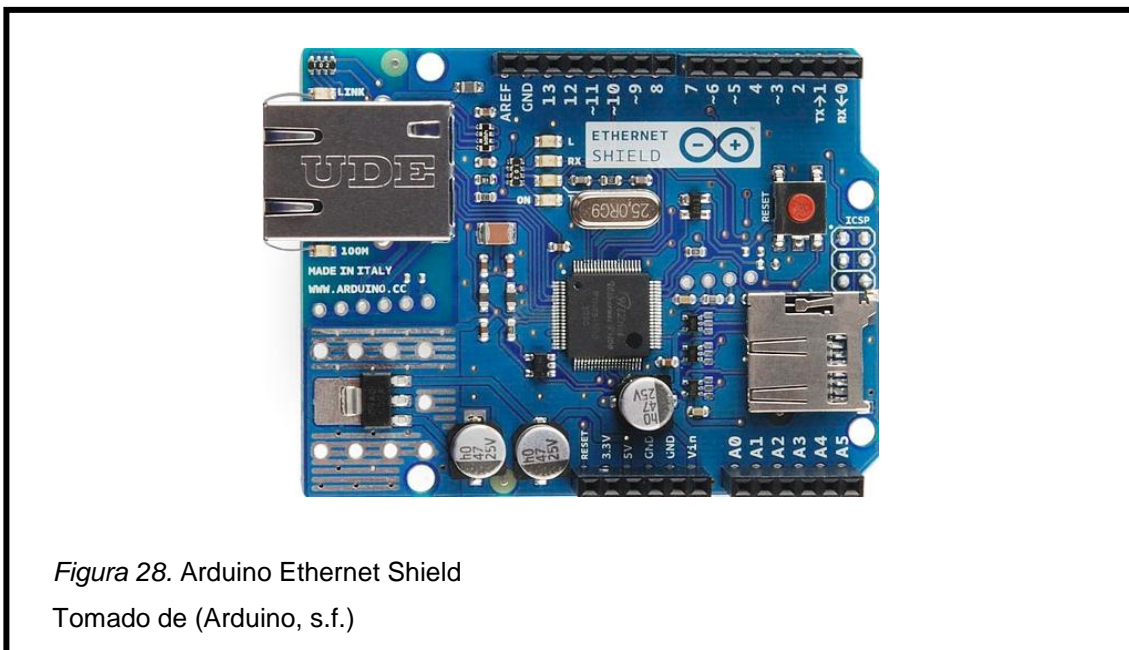
El arduino UNO utiliza los pines 10, 11, 12 y 13 (SPI) para comunicarse con el W5100 en el *Ethernet shield* a través de la cabecera ICSP.

Algunas de las características de la *Ethernet shield* son las siguientes:

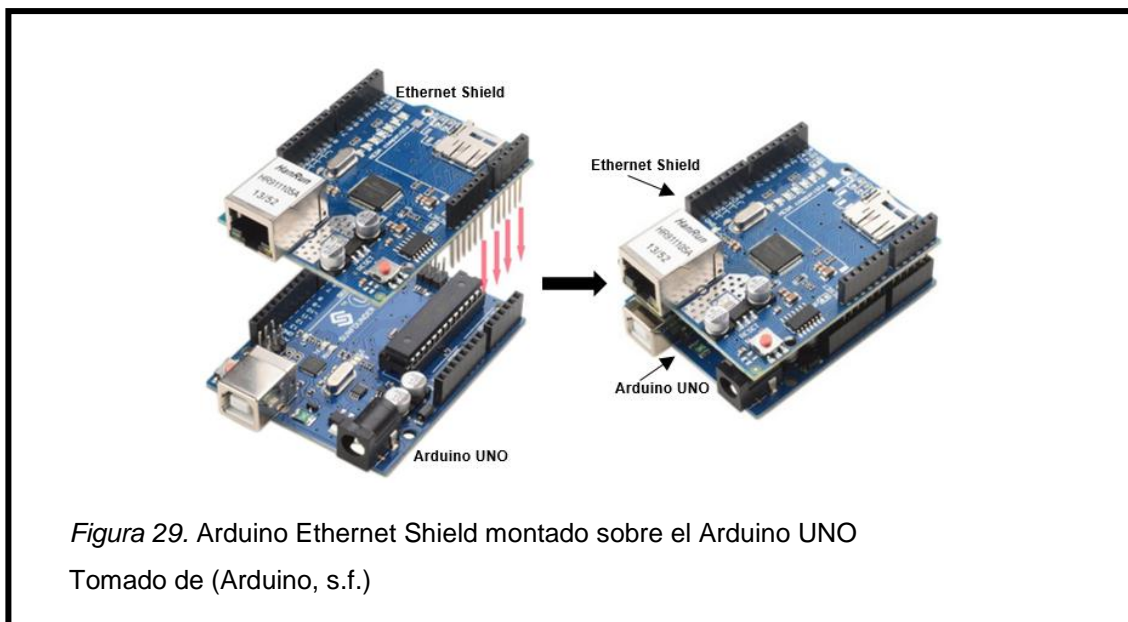
- Tensión de alimentación de 5V (suministrado por el Arduino)
- Controlador de internet: con una memoria interna de 16K
- Velocidad de conexión 10/100Mb
- Conexión con Arduino por el puerto SPI

Además la *Ethernet shield* tiene algunos LEDs informativos:

- TX: titila cuando la *shield* envía datos
- RX: titila cuando la *shield* recibe datos
- COLL: titila al existir colisiones en la red
- FULLD: indica que la conexión de red es full duplex
- 100M: indica que la conexión de red es del 100Mb/s (en lugar de 10Mb/s)
- LINK: indica la existencia de conexión de red, y titila cuando la *shield* envía o recibe datos.
- POWER: indica que el Arduino UNO y el *Ethernet shield* se encuentran alimentados.

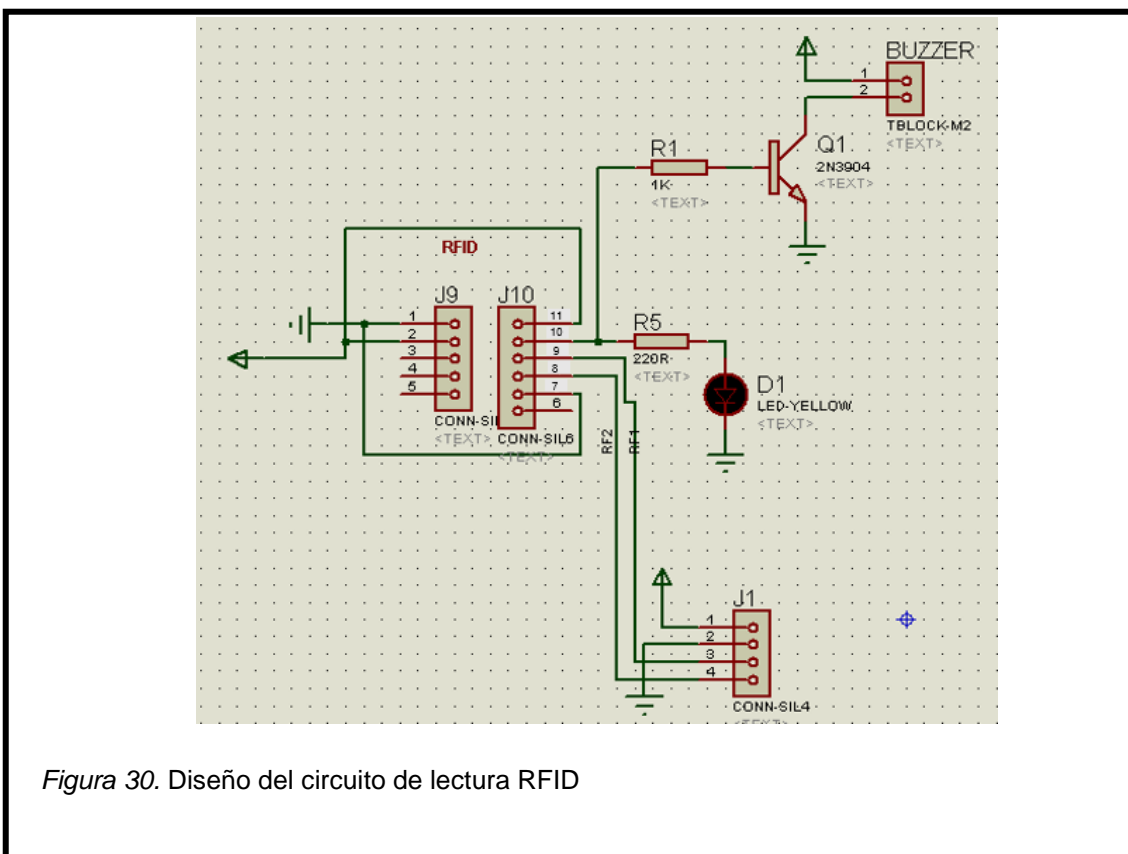


Al montar el Arduino *Ethernet Shield* sobre el Arduino UNO se obtiene el siguiente resultado:



2.5 Diseño del Lector RFID

Tomando en cuenta la funcionalidad de cada PIN de salida que posee el módulo lector ID-12LA y sus características operacionales, se procede con el diseño del circuito esquemático.



En la figura 30, el conector (J9) y (J10) representan los pines que posee el módulo ID-12LA; tomando en cuenta la información de la Figura 24 y la descripción del módulo ID12-LA se conoce detalladamente la funcionalidad de cada pin:

- Pin 1 va a conexión a tierra.
- Pin 2 es de reinicio, va a positivo debe estar atado en todo momento al pin 11 que es de 5V el cual permite la alimentación del módulo.
- Pin7 es el selector, va conectado a tierra debido al formato de datos de salida.
- Pin 8 y Pin 9 son pines de datos, para ellos se utiliza un conector (J1) el cual servirá para establecer conexión con el Arduino UNO, van conectados a los pines 4 y 3 del conector (J1) respectivamente.
- Pin 10 llamado *Buzzer* es un indicador exclusivo de sonido o led que muestra la respuesta de lectura al pasar la etiqueta por el lector. Tiene una resistencia (R5) que permite que el led (D1) funcione y se encienda sin que se quemé por exceso de corriente. Por otro lado hay una resistencia (R1) conectada a un transistor (Q1) que permite el paso de corriente en un solo sentido y de forma controlada para el funcionamiento del *Buzzer*.

El led indicador (D1) y el *BUZZER* muestran que el lector sí está mandando datos, el encendido del led y el sonido del *buzzer* garantizan la transmisión de datos.

Para determinar el valor de la resistencia (R5) que se usa para encender el led (D1) en arduino, se toma en cuenta que el arduino maneja 5V, la corriente del led va entre los 5-20mA, por lo que se necesita una resistencia (R5) para no quemar el led. A través de la ley de Ohm se realiza el cálculo de la resistencia.

La caída de voltaje que demanda el led rojo es de aproximadamente 2V.

La ley de Ohm dice que el Voltaje es igual a la corriente por la resistencia.

Entonces: $V = I * R$

Se despeja y queda $R = \frac{V}{I}$ (Ecuación 1)

$$R5 = \frac{(5-2)V}{15mA}$$

$$R5 = \frac{3V}{0.015mA} \longrightarrow R5 = 200 \text{ ohms}$$

La resistencia R5 es 200 Ohms, pero no es un valor comercial. Se busca un valor comercial que esté entre ese valor, 220 ohm o 330 ohm sirven perfectamente, ya que son las resistencias más utilizadas para encender un led.

El cálculo de la Resistencia del *buzzer* (R1) se realiza de igual manera con la ley de Ohm, para ello se utilizó una resistencia de 1k Ohm.

A través de este cálculo se garantiza el buen dimensionamiento de los componentes utilizados.

La siguiente tabla detalla técnicamente los nombres de los elementos utilizados en la Figura 30.

Tabla 5. Referencia de los elementos utilizados en el circuito

Referencia	Tipo	Valor
J1	CONN-SIL4	CONN-SIL4
J9	CONN-SIL5	CONN-SIL5
J10	CONN-SIL6	CONN-SIL6
R1	MINRES220R	1K
R5	MINRES200R	330R
Q1	2N3904 NPN	2N3904 NPN
D1	LED	LED
<i>BUZZER</i>	TBLOCK-M2	TBLOCK-M2

Al conectar correctamente los elementos para el funcionamiento del módulo lector RFID, se comprueba la funcionalidad del circuito esquemático en el simulador electrónico Proteus 7.8 y se procede a imprimir en baquelita para la construcción manual del mismo.

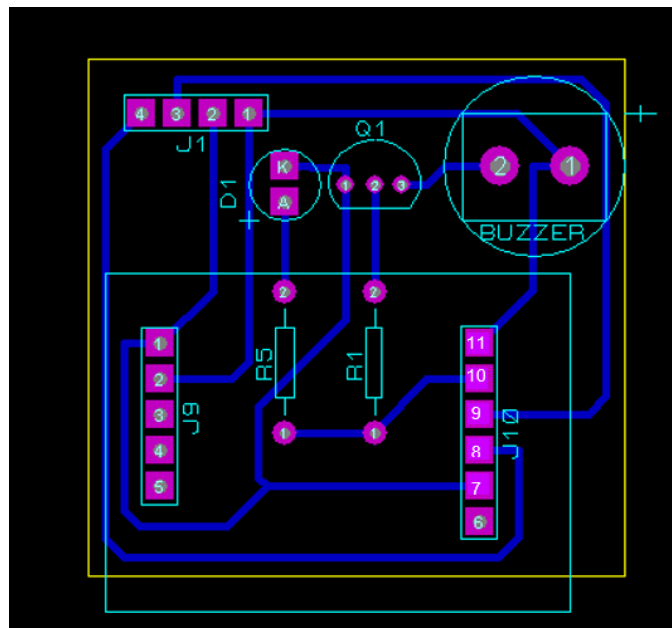


Figura 31. Diseño del circuito para la impresión en baquelita

Se puede visualizar el circuito en 3D para conocer el resultado final del circuito implementado en la baquelita.

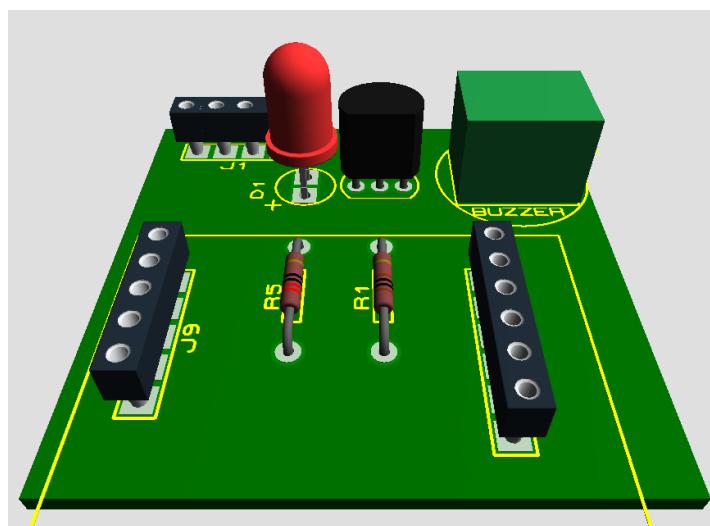


Figura 32. Visualización del circuito en 3D

Una vez impreso el circuito en la baquelita se sueldan todos los componentes según el diseño hasta obtener finalmente el circuito físico del lector RFID.

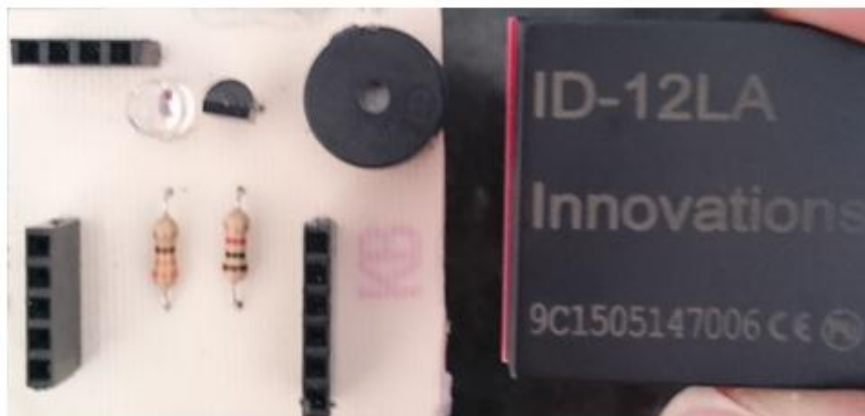


Figura 33. Lector RFID sin el modulo ID-12LA

Se procede con la conexión del módulo lector ID-12LA con el Arduino Uno y el *Ethernet Shield*.

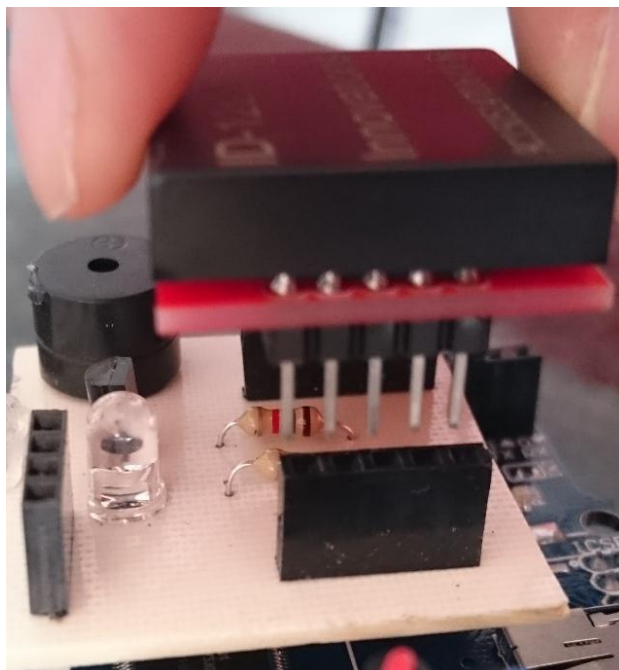
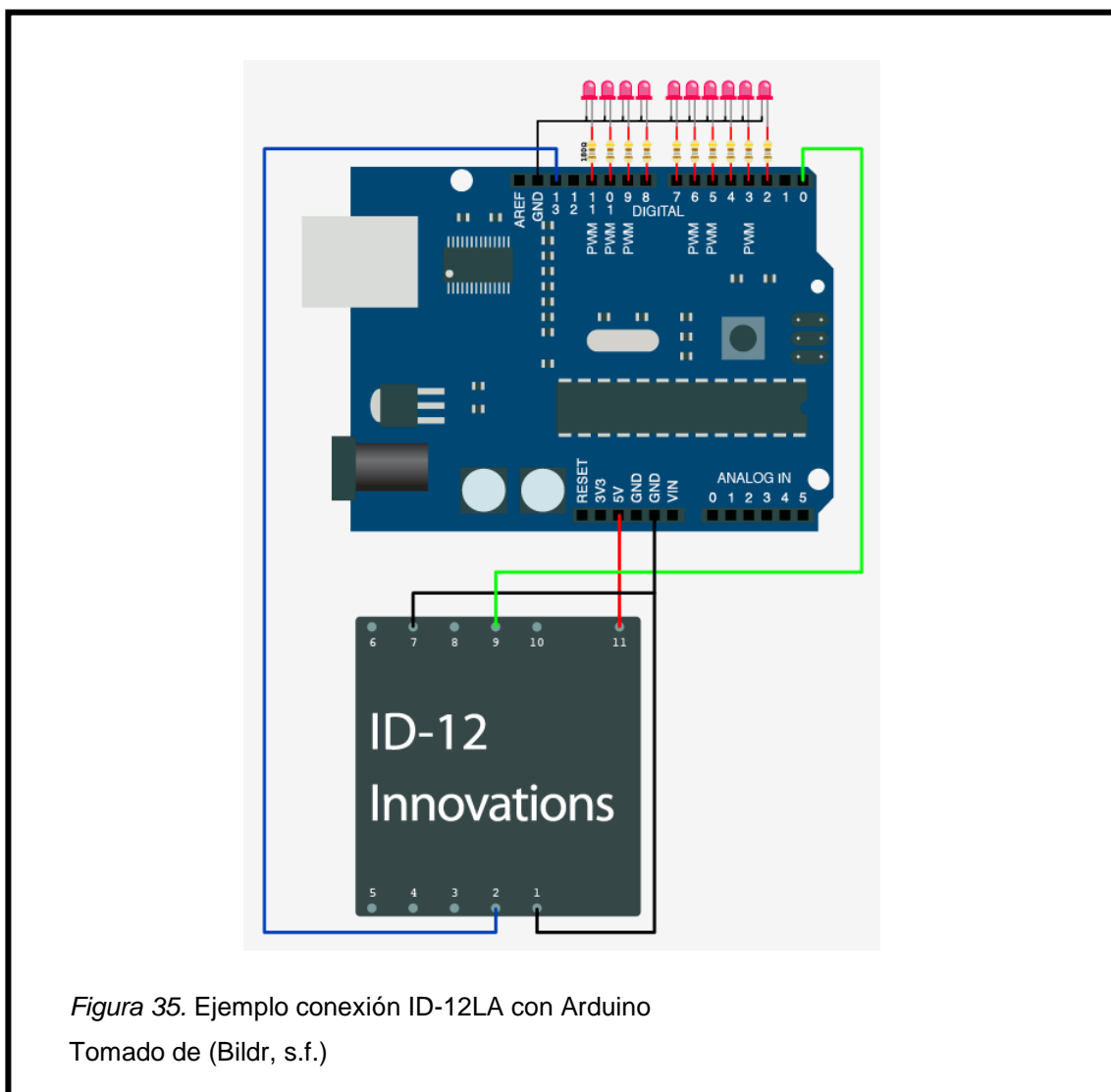


Figura 34. Lector RFID

El uso del Arduino y el *Ethernet Shield* permiten que el lector no requiera estar conectado a un ordenador para la lectura de las etiquetas una vez que el programa ha sido cargado en el Arduino.

El módulo ID-12LA mediante comunicación RFID lee el código de las etiquetas y lo entrega en formato serial, el Arduino se encarga de capturar ese código mediante comunicación serial, por lo tanto, sólo se requiere la transmisión (TX) de los datos del pin 9 del ID-12LA que va conectado al Arduino en el pin 3 del conector (J1) para la recepción (RX) de los mismos y a través del *Ethernet Shield* se conecta a Internet.

En la siguiente imagen se muestra un ejemplo de cómo controlar el Arduino sobre comunicación serial mediante módulo ID-12LA.



Por comodidad y facilidad de conexión, el circuito lector RFID es colocado sobre la *Ethernet Shield* (Figura 34). El pin 1 del conector (J1) del circuito lector RFID va conectado a través de un cable rojo con el pin de 5V del *Ethernet Shield*, el pin 2 va conectado al pin GND/Tierra a través de una cable azul y el pin 3 es el pin de salida de datos va conectado al pin 2 del arduino.

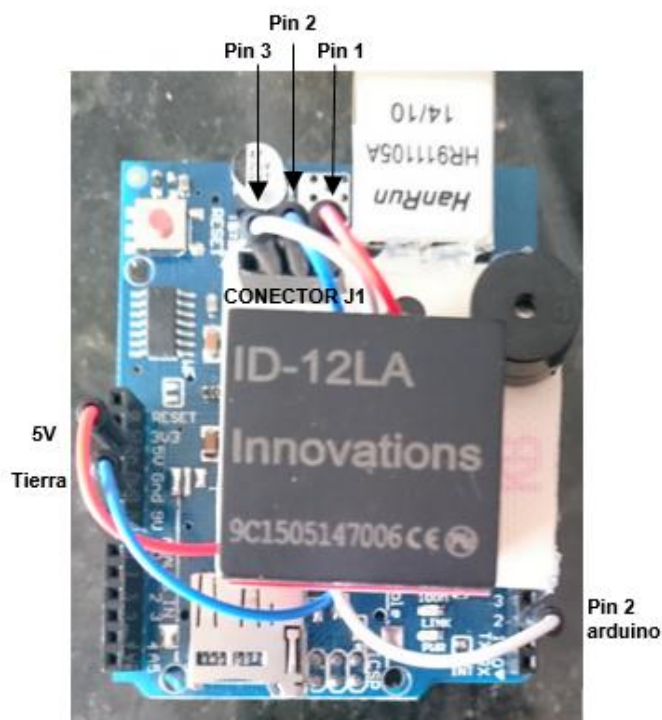


Figura 36. Conexión del lector RFID con el Arduino

3. DESARROLLO DEL SOFTWARE

3.1 Software del Arduino

El software Arduino (IDE) se lo descarga de la página www.arduino.cc, esta plataforma permitirá realizar la programación de la placa y la verificación de la lectura de datos que recibe el lector al pasar las etiquetas con el fin de comprobar el funcionamiento del hardware del sistema.

3.1.1 Lectura de Etiquetas

Se conecta la placa Arduino a la PC a través de cable USB, el led verde deberá encenderse. Al conectar el arduino también estará alimentado el lector RFID.



Figura 37. Conexión del lector y arduino a la PC

En el Administrador de dispositivos, una vez instalado los drives del software Arduino IDE, se verifica el puerto COM por el cual se ha identificado al arduino. (Arduino UNO (COM6)).

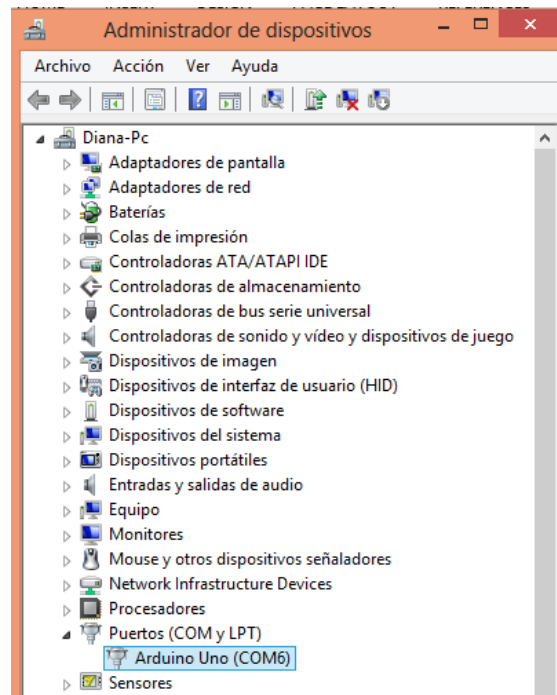


Figura 38. Puerto COM6 para arduino UNO

En el software Arduino, en la opción herramientas se selecciona el tipo de placa utilizada, en este caso, se selecciona Arduino Uno.

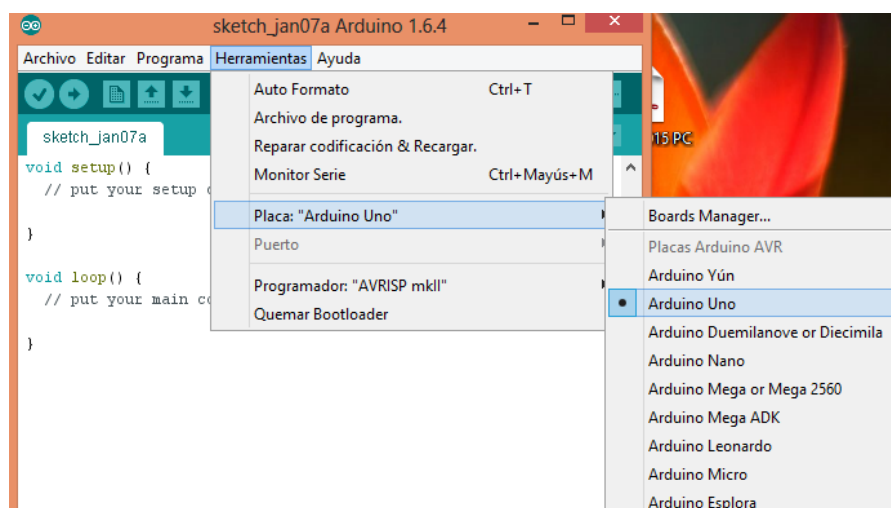


Figura 39. Selección de placa utilizada

Se elige el puerto serie COM6 asignado para el arduino.

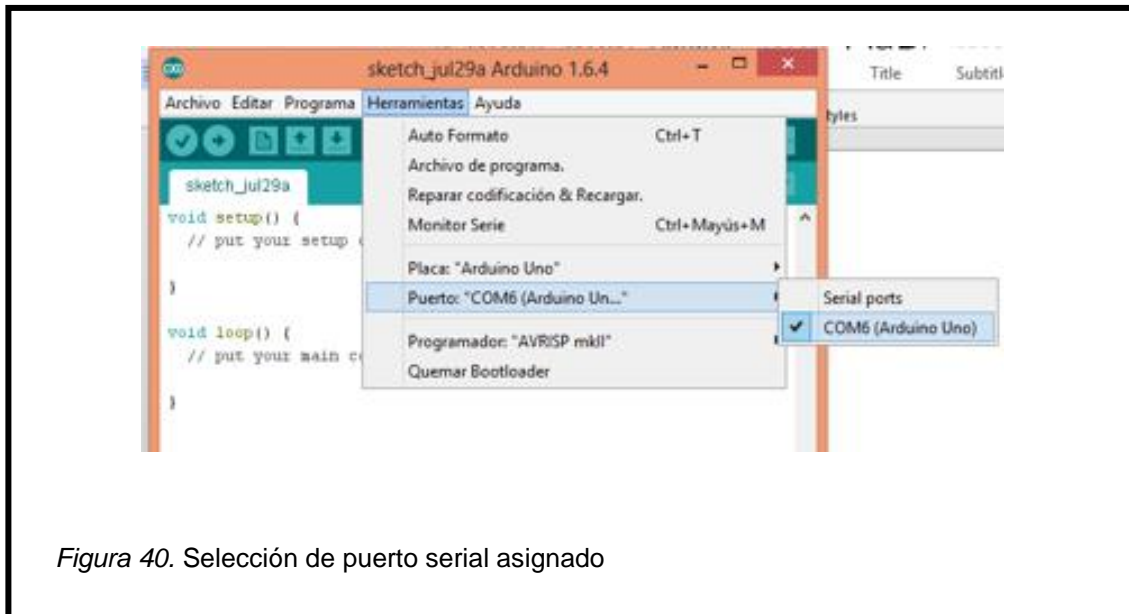


Figura 40. Selección de puerto serial asignado

Se realiza un pequeño programa que inicializará la comunicación serial permitiendo la lectura de las etiquetas al pasar por el lector RFID. Se sube el programa dando clic en flecha que se encuentra debajo de "Archivo" del menú.



Figura 41. Programa para la lectura de etiquetas

Al pasar las etiquetas sobre el lector se comprueba el funcionamiento del mismo cuando el programa recibe los datos enviados.

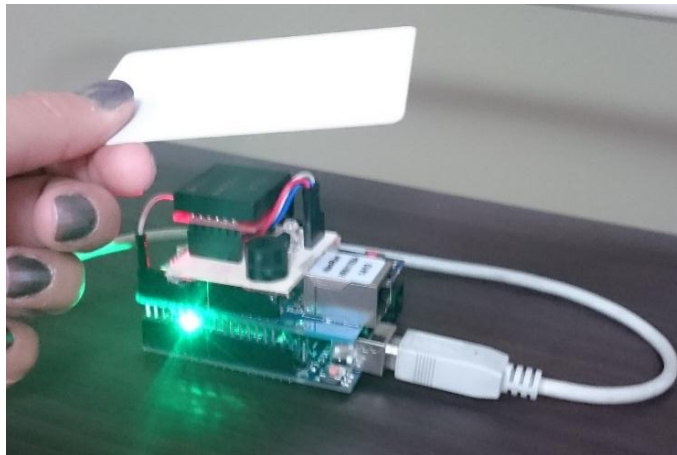


Figura 42. Paso de la etiqueta por el lector RFID

A través del monitor serial del software Arduino se observa la información enviada por las etiquetas.

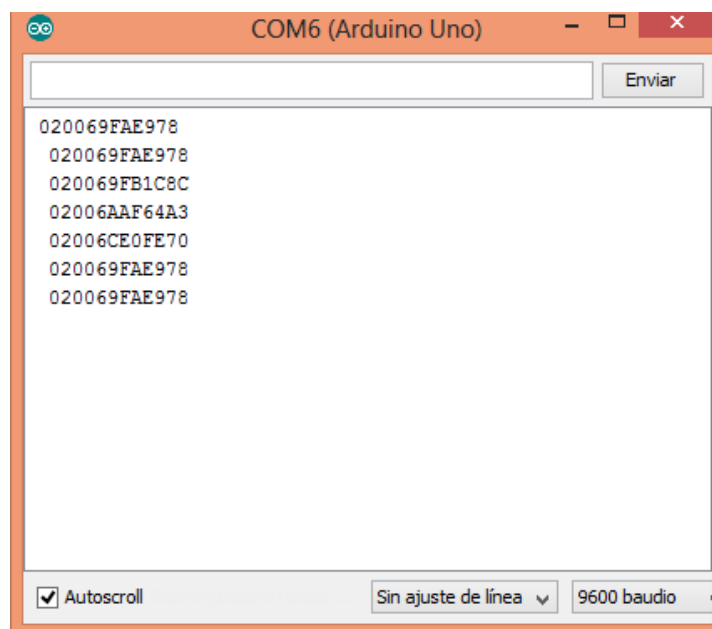
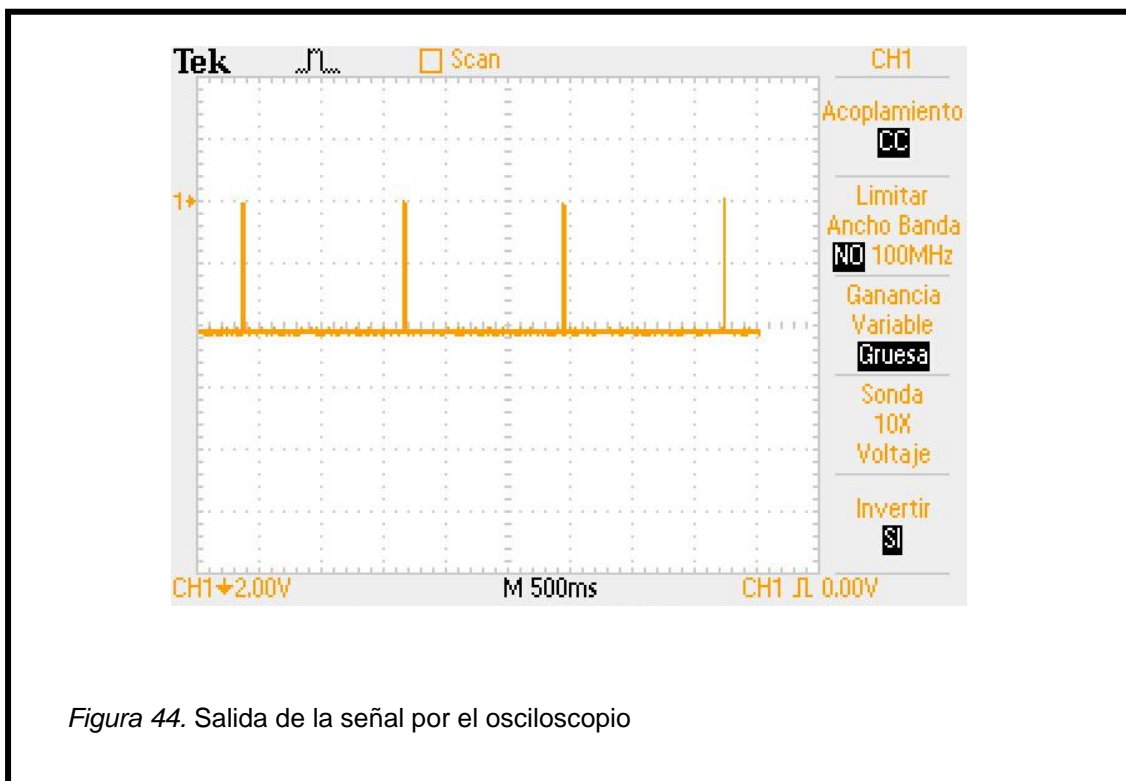


Figura 43. Información de cada etiqueta

Los datos de salida que aparecen en la pantalla se encuentran en formato ASCII de 12 bytes de longitud.

A través del osciloscopio se verifica la transmisión de datos cada vez que pasa la etiqueta por el lector, la siguiente imagen muestra la señal de salida a 500ms.



El lector RFID emite una frecuencia de 125 KHz para que la tarjeta al entrar al campo de RF se cargue y envíe la información mediante modulación. La información recibida es la que se muestra en la Figura 43.

3.1.2 Conexión del Ethernet Shield

Una vez comprobada la lectura de las etiquetas, se procede a verificar el funcionamiento y conexión con la red de la *Ethernet Shield* a través de ejemplos que vienen en el mismo software, para ello, se conecta el arduino con el cable USB y la *Ethernet shield* a un *router* con el cable de red.

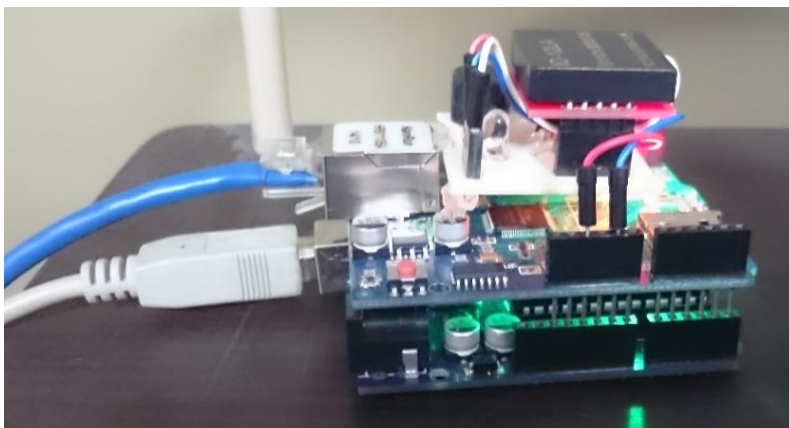


Figura 45. Conexión del arduino y de la Ethernet shield

La *Ethernet shield* cuenta con una librería Ethernet que permite a la placa arduino que se conecte a Internet. En archivo, en la opción ejemplos se selecciona *Ethernet* y escogemos la pestaña *WebClient* la cual hace una petición HTTP.

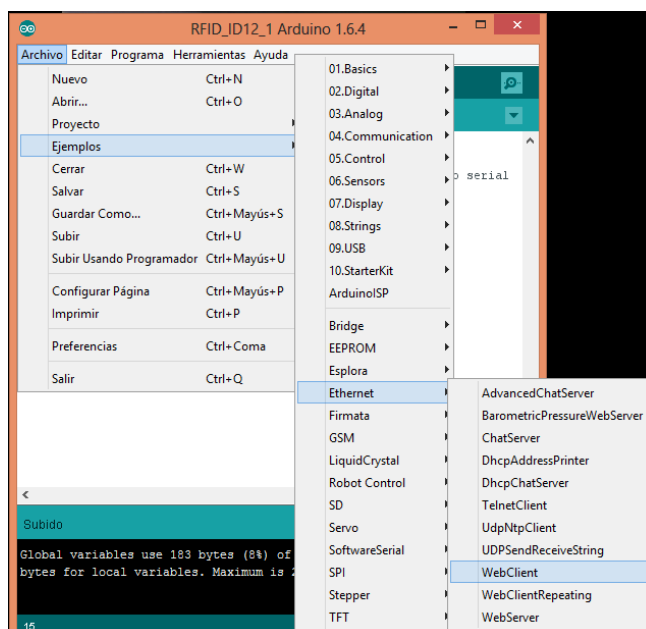


Figura 46. Selección de ejemplo Ethernet

Este ejemplo se conecta a un sitio web (<http://www.google.com>) usando la *Ethernet shield* e imprime en el puerto serie todo lo que contenga en la búsqueda la palabra Arduino una vez subido el programa.

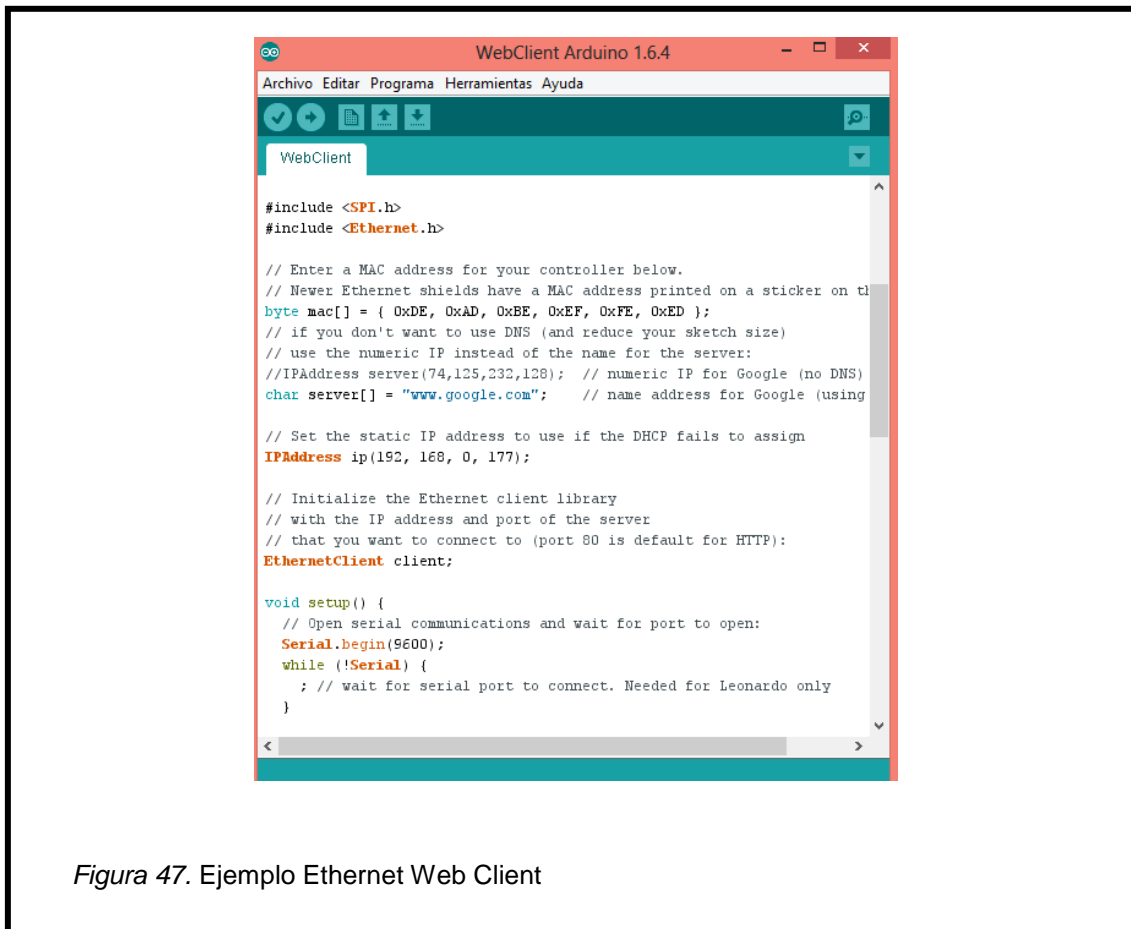


Figura 47. Ejemplo Ethernet Web Client

A través del monitor serial se verifica la conexión y el funcionamiento del programa WebClient.

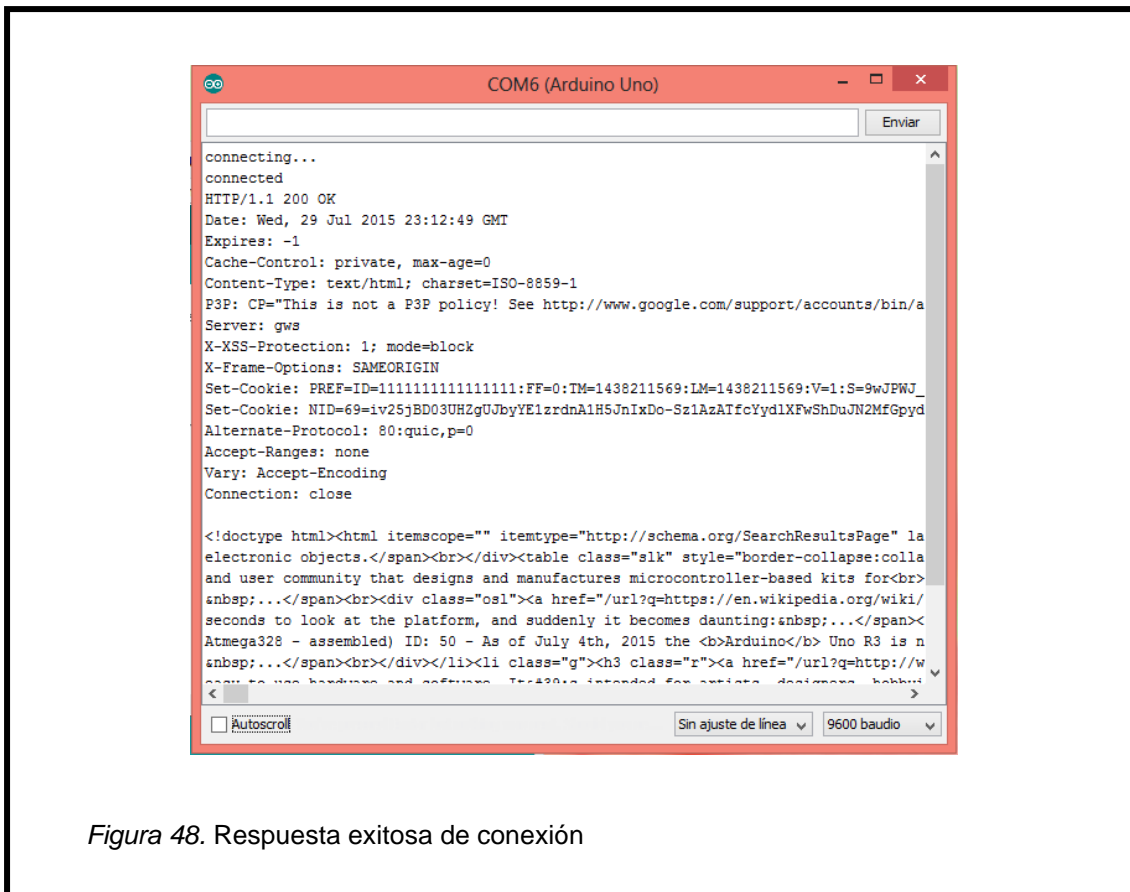
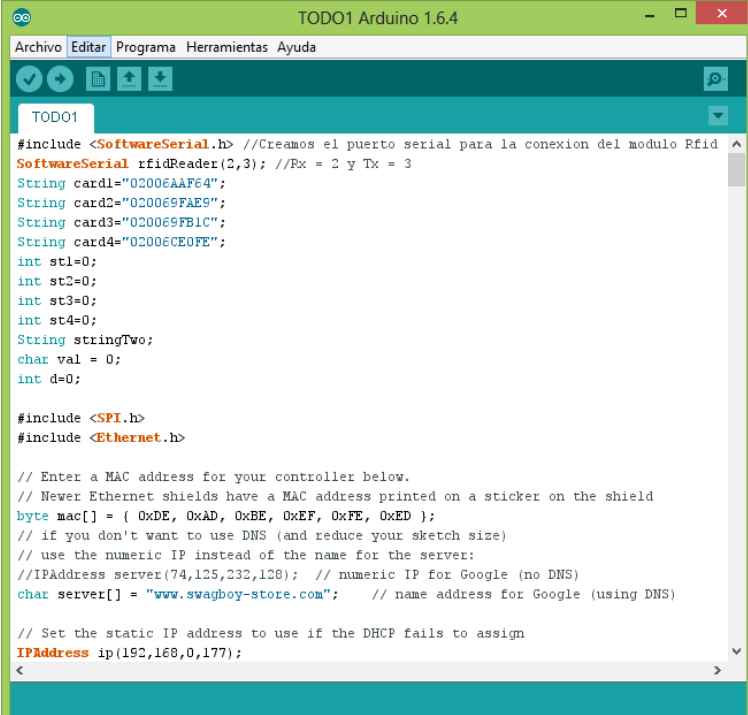


Figura 48. Respuesta exitosa de conexión

3.1.3 Programa Arduino

Una vez comprobada la lectura de etiquetas y la conexión de la *Ethernet Shield*, se realiza el programa final, el cual es subido en el Software de Arduino (IDE) para la lectura de las etiquetas al momento de pasar por el lector RFID permitiendo que la información sea recibida por el servidor donde va a ir la aplicación Web del software RFID.



```

TODO1
#include <SoftwareSerial.h> //Creamos el puerto serial para la conexion del modulo Rfid
SoftwareSerial rfidReader(2,3); //Rx = 2 y Tx = 3
String card1="02006AAF64";
String card2="020069FAE9";
String card3="020069FB1C";
String card4="02006CE0FE";
int st1=0;
int st2=0;
int st3=0;
int st4=0;
String stringTwo;
char val = 0;
int d=0;

#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// if you don't want to use DNS (and reduce your sketch size)
// use the numeric IP instead of the name for the server:
//IPAddress server(74,125,232,128); // numeric IP for Google (no DNS)
char server[] = "www.swagboy-store.com"; // name address for Google (using DNS)

// Set the static IP address to use if the DHCP fails to assign
IPAddress ip(192,168,0,177);

```

Figura 49. Programa final para el Hardware del prototipo

Para cargar los programas a la placa Arduino se utiliza el cable USB para conectarse con la PC, una vez que el programa ha sido cargado se puede conectar el arduino con una fuente de alimentación externa de 9-12V. La *Ethernet Shield* debe estar en todo momento conectada al *router* para brindarle conexión a internet al arduino.

El siguiente diagrama de flujo explica el proceso del sistema final una vez cargado el programa en la plataforma de arduino para el funcionamiento de la lectura de etiquetas al aproximarse al lector e interacción con el aplicativo web. (Ver Anexo 2)

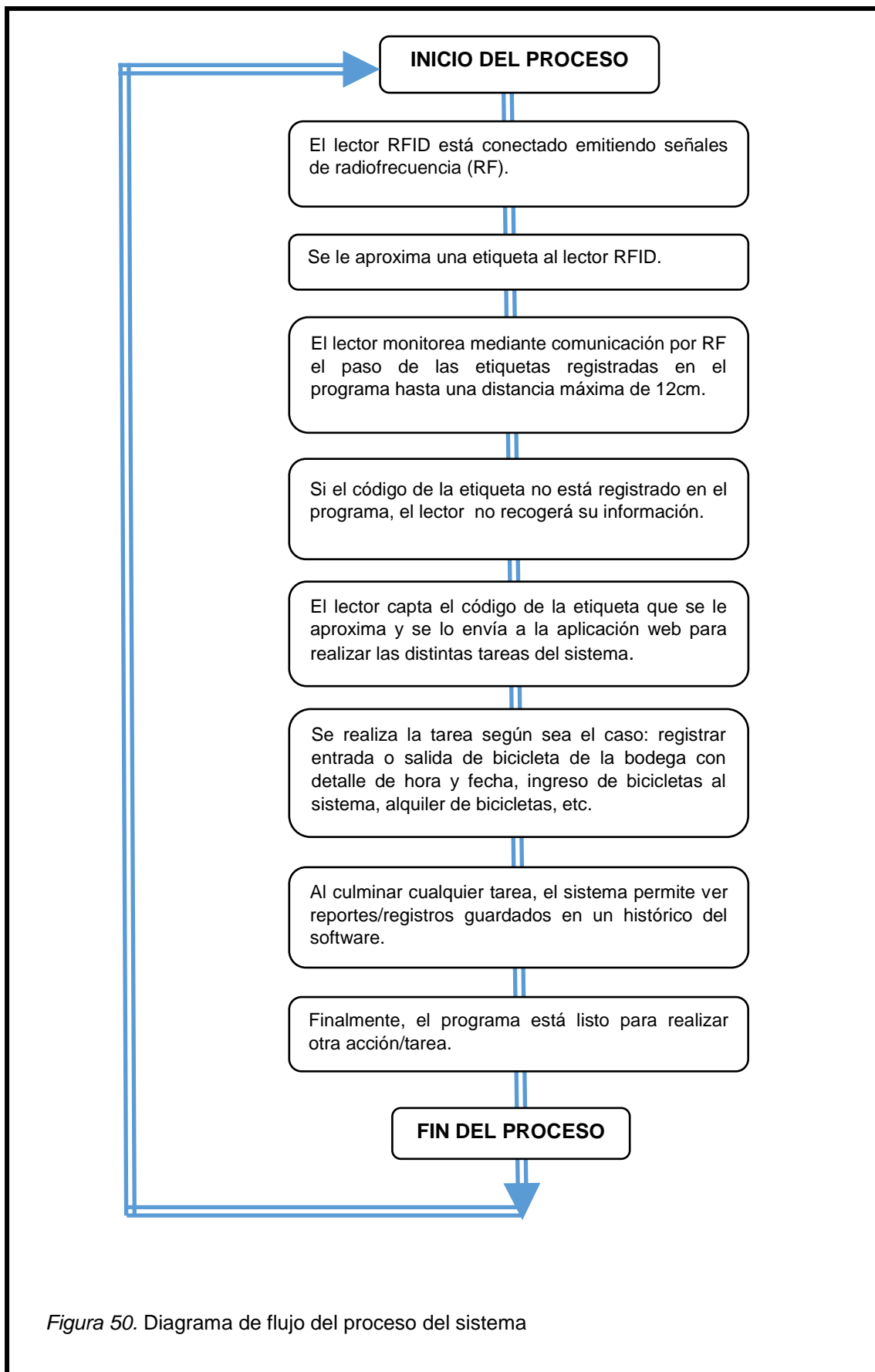


Figura 50. Diagrama de flujo del proceso del sistema

3.2 Entorno del Software del Sistema

El middleware se encarga de la conexión de los sistemas de información existentes en la aplicación para el control de inventario y de la entrada y salida de las bicicletas con el hardware RFID. Es el sistema informático de procesamiento de datos cuya principal función es gestionar y almacenar la información recibida por el hardware RFID.

La aplicación está enfocada a resolver las problemáticas asociadas al actual funcionamiento de la bodega de BiciQ, mediante el análisis de requerimientos se determinan y especifican los elementos y funciones necesarios para el proyecto.

3.2.1 Base de Datos

MySQL es el gestor de base de datos de código abierto utilizado para el desarrollo del software RFID, tiene mayor aceptación a nivel mundial, es muy popular en el desarrollo de aplicaciones web debido a que forma parte como sistema gestor de base de datos de algunas plataformas como lo es WAMP por lo que se acopla perfectamente y trabaja de manera estable con PHP

3.2.2 Lenguaje de Programación

PHP es la herramienta seleccionada para el desarrollo del software, es un lenguaje de código abierto (de uso libre y gratuito) que funciona perfectamente para la creación de páginas web con acceso a información almacenada en una base de datos, está enfocado principalmente a la programación de scripts del lado del servidor, y puede ser introducido en HTML, es decir que se pueden crear y editar páginas web con contenido dinámico de la misma forma que normalmente se crean páginas HTML, siguiendo obviamente unas reglas, por lo tanto este lenguaje se acopla y funciona perfectamente en el desarrollo de la aplicación del prototipo. (PHP, s.f.)

Al trabajar en un entorno local, se puede hacer uso del paquete WampServer que simula un servidor en la PC. WampServer es un entorno de desarrollo Web de Windows, puede crear aplicaciones web con Apache como servidor web, PHP como intérprete del lenguaje de programación y MySQL como gestor de base de datos. Por otro lado, PhpMyAdmin permite administrar fácilmente las bases de datos. (Wampserver, s.f.)

Al trabajar con un servidor remoto, existen algunos gratuitos que funcionan perfectamente, para lo cual, se requiere el registro de un hosting que ofrezca de preferencia alojamiento gratuito y a partir de ahí, con un cliente ftp, como FileZilla, se crean páginas web dinámicas. (González, E, 2015)

3.2.3 Servidor

El servidor utilizado brinda un servicio de alojamiento web fiable y rico en funciones sin publicidad perfecto para el prototipo realizado.

3.3 Análisis de Requerimientos

La obtención de los requerimientos es la primera etapa necesaria para el diseño del software a desarrollar. El análisis de requerimientos consiste en la recopilación de todos los requisitos y necesidades del cliente para el sistema, un requisito describe los servicios que debe ofrecer el sistema y sus restricciones. El análisis y la obtención de requisitos se enfocan sólo en la visión del sistema que tiene el usuario

.

Existen requerimientos funcionales como no funcionales; el funcional define lo que se espera que deba de hacer el sistema, se describe la interacción entre el sistema y el entorno, se detallan servicios y funciones que proveerá el sistema. El requisito no funcional, define cómo debe ser el sistema, describe restricciones que limitan las elecciones para construir una solución, son

atributos relacionados con la calidad como rendimiento, escalabilidad, fiabilidad, disponibilidad, mantenimiento, seguridad, etc.

En base a la constante búsqueda de información concisa del funcionamiento de la tecnología por identificación de radiofrecuencia y de la bodega de BiciQ, mediante el análisis de las necesidades del personal encargado yendo de una parte concreta del problema al todo se levantaron los requerimientos tanto funcionales como no funcionales necesarios para el desarrollo del sistema.

3.3.1 Determinación de los requerimientos de información

La bodega principal de BiciQ funciona a la vez como taller, en ella llegan bicicletas para que sean guardadas y así mismo bicicletas que necesitan reparación inmediata para que el servicio ofrecido por esta empresa pueda continuar brindándose adecuadamente por quienes lo demandan. Esta bodega o taller tiene la necesidad de contar con un sistema automatizado que le permita llevar un mejor control de las bicicletas que entran y salen de la bodega, y que a su vez sea fácil de manejar para una mejor optimización del tiempo, debido a que actualmente el registro que realizan los mecánicos es un proceso manual que está propenso a fallas.

El control consiste en llevar a cabo un registro o reporte que muestre todas las bicicletas que han ingresado y salido del taller de mecánica o bodega, el mantenimiento que se les ha realizado, el estado en el que se encuentran y los distintos repuestos que se han utilizado en las bicicletas al momento de ser reparadas.

La aplicación registrará todas aquellas actividades que han sido realizadas por el mecánico u operador de la bodega, el administrador será el único que podrá registrar, modificar, eliminar usuarios y la información principal de cada bicicleta y stock de repuestos.

El sistema permitirá realizar consultas a través del único código identificador que tiene cada etiqueta RFID en cada bicicleta. La aplicación le permitirá al usuario administrador la impresión de reportes o inventarios.

3.3.2. Diagrama de casos de uso

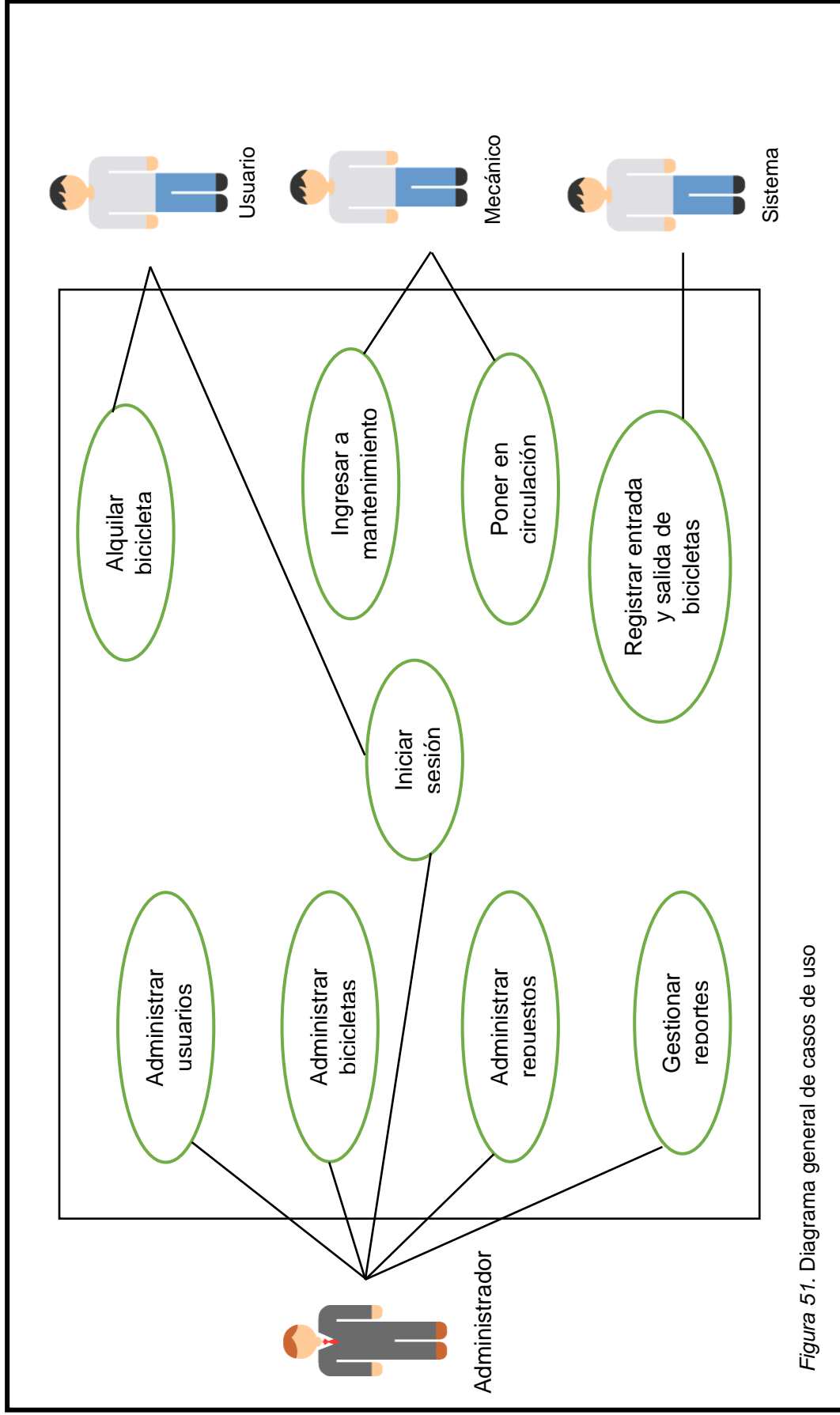


Figura 51. Diagrama general de casos de uso

3.3.3 Casos de uso

Tabla 6. Caso de uso 01: Iniciar sesión

CU-01:	Iniciar sesión	
Actor:	Administrador, usuario y mecánico	
Descripción:	El sistema deberá permitir al administrador, usuario común o mecánico ingresar a la aplicación web al insertar su usuario y contraseña según se describe en el siguiente caso de uso.	
Precondición:	Estar registrados en la base de datos del sistema.	
Secuencia Normal:	Paso	Acción
	1	El administrador, usuario o mecánico ingresan su nombre de usuario en el sistema.
	2	El administrador, usuario o mecánico ingresar su contraseña.
	3	El administrador, usuario o mecánico dan clic en el botón Iniciar Sesión.
Postcondición:	El administrador, usuario o mecánico acceden al sistema.	
Excepciones:	Paso 3	Acción Si el administrador, usuario o mecánico ingresa usuario o contraseña incorrecta el sistema mostrará el siguiente error: Sus credenciales son incorrectas, por favor inténtelo de nuevo.
	3	Si el administrador, usuario o mecánico no dejan el campo de usuario o contraseña vacío y dan clic en iniciar Sesión el sistema mostrará el siguiente aviso: Completa este campo
Prioridad:	Alta	
Frecuencia de uso:	Bajo demanda	
Comentarios:	N/A	

Tabla 7. Caso de uso 02: Administrar usuarios

CU-02:	Administrar usuarios	
Actor:	Administrador	
Descripción:	El sistema deberá permitir al administrador añadir, modificar, buscar y eliminar usuarios según se describe en el siguiente caso de uso.	
Precondición:	Iniciar sesión en el sistema teniendo permisos de administrador.	
Secuencia Normal:	Paso	Acción
	1	En el menú del sistema, el administrador selecciona la opción Usuarios.
	2	El sistema despliega la información de todos los usuarios registrados.
	3	Para agregar un nuevo usuario, el administrador da clic en el botón "Añadir usuario".
		3.1 El sistema abre una ventana para colocar la información del nuevo usuario.
		3.2 El administrador llena los campos con la información del usuario y da clic en el botón "Guardar".
		3.3 El sistema inmediatamente muestra en la lista de todos los usuarios creados, el nuevo usuario.
3.4 Los campos de la ventana para crear un nuevo usuario vacían, el administrador puede crear otro usuario o cerrar la ventana.		
4	Si el administrador quiere modificar la información del usuario, en la lista de todos los usuarios, bajo la columna	

		Acciones, debe dar clic en el gráfico de un lápiz.
		4.1 El sistema abre una ventana para modificar la información.
		4.2 El administrador modifica la información y da clic en el botón "Modificar información".
		4.3 El sistema cierra la ventana para modificar información, y despliega la información de todos los usuarios registrados.
	5	Si el administrador quiere eliminar usuarios, en la lista de todos los usuarios, bajo la columna Acciones, debe dar clic en el gráfico de un basurero. Inmediatamente se borra el usuario.
	6	Si el administrador quiere buscar un usuario en específico, debe ingresar el nombre o número de cédula del usuario y dar clic en el botón "Buscar".
Postcondición:	Los usuarios pueden ser administrados en el sistema.	
Excepciones:	Paso	Acción
	3	Si al añadir un usuario, algún campo se lo deja vacío, el sistema mostrará el siguiente mensaje: "Completa este campo".
	4	Si alguno de los campos se deja vacío al modificar usuarios, el sistema mostrará el siguiente mensaje: "Completa este campo".
Prioridad:	Alta	
Frecuencia de uso:	Bajo demanda	
Comentarios:	N/A	

Tabla 8. Caso de uso 03: Administrar bicicletas

CU-03:	Administrar bicicletas	
Actor:	Administrador	
Descripción:	El sistema deberá permitir al administrador añadir una nueva bicicleta con su número de <i>tag</i> identificador, así mismo podrá modificar, buscar y eliminar bicicletas.	
Precondición:	Iniciar sesión en el sistema teniendo permisos de administrador. Tener el número de <i>tag</i> de la bicicleta que se va a añadir.	
Secuencia Normal:	Paso	Acción
	1	En el menú del sistema, el administrador selecciona la opción "Bicicletas".
	2	El sistema despliega la información de todas las bicicletas registradas: id <i>Tag</i> , color, marca, estado y acciones.
	3	Para agregar una nueva bicicleta, el administrador da clic en el botón "Añadir bicicleta".
		3.1 El sistema abre una ventana para colocar la información bicicleta a agregar.
		3.2 El administrador llena los campos: id <i>tag</i> , color y marca de la bicicleta y da clic en el botón "Guardar".
		3.3 El sistema inmediatamente muestra en la lista la nueva bicicleta añadida.
		3.4 Los campos de la ventana para crear una nueva bicicleta se vacían, el administrador puede crear otra bicicleta o cerrar la ventana.
	4	Si el administrador quiere modificar la información de la bicicleta, en la lista de bicicletas, bajo la columna Acciones, debe dar clic en el gráfico de un lápiz.

		4.1 El sistema abre una ventana para modificar la información.
		4.2 El administrador modifica la información y da clic en el botón "Modificar información".
		4.3 El sistema cierra la ventana para modificar información, y despliega la información de todas las bicicletas.
	5	Si el administrador quiere eliminar bicicletas, en la lista de bicicletas, bajo la columna "Acciones", debe dar clic en el gráfico de un basurero. Inmediatamente se borra la bicicleta.
	6	Si el administrador quiere buscar una bicicleta en específico, debe ingresar el id del <i>tag</i> de la bicicleta y dar clic en el botón "Buscar".
Postcondición:	Las bicicletas pueden ser administradas en el sistema.	
Excepciones:	Paso	Acción
	3	Si al añadir una bicicleta, algún campo se lo deja vacío, el sistema mostrará el siguiente mensaje: "Completa este campo".
Prioridad:	Alta	
Frecuencia de uso:	Bajo demanda	
Comentarios:	N/A	

Tabla 9. Caso de uso 04: Administrar repuestos

CU-04:	Administrar repuestos	
Actor:	Administrador	
Descripción:	El sistema deberá permitir al administrador añadir nuevos repuestos al inventario con su nombre, cantidad y precio por unidad, podrá modificar, buscar y eliminar repuestos.	
Precondición:	Iniciar sesión en el sistema teniendo permisos de administrador.	
Secuencia Normal:	Paso	Acción
	1	En el menú del sistema, el administrador selecciona la opción "Repuestos".
	2	El sistema despliega la información del stock de repuestos.
	3	Para agregar un nuevo repuesto al inventario, el administrador da clic en el botón "Añadir repuesto".
		3.1 El sistema abre una ventana para añadir los datos del nuevo repuesto.
		3.2 El administrador llena los campos: descripción, stock y precio por unidad del repuesto y da clic en el botón "Guardar".
	3.3 El sistema inmediatamente muestra en el inventario el repuesto añadido.	
	4	Si el administrador quiere modificar la información del repuesto, en el inventario, bajo la columna Acciones, debe dar clic en el gráfico de un lápiz.
		4.4 El sistema abre una ventana para modificar la información.
		4.5 El administrador modifica la información y da clic en el botón "Modificar información".
4.6 El sistema cierra la ventana para modificar información, y despliega el inventario de todos los repuestos con las modificaciones realizadas.		

	5	Si el administrador quiere eliminar repuestos del inventario, bajo la columna "Acciones", debe dar clic en el gráfico de un basurero. Inmediatamente se borra el repuesto seleccionado.
	6	Si el administrador quiere buscar un repuesto en específico, debe ingresar la descripción/nombre del repuesto y dar clic en el botón "Buscar".
Postcondición:	Los repuestos pueden ser administrados en el sistema.	
Excepciones:	Paso	Acción
	3	Si al añadir un repuesto, algún campo se lo deja vacío, el sistema mostrará el siguiente mensaje: "Completa este campo".
	4	Si al modificar un repuesto, algún campo se lo deja vacío, el sistema mostrará el mensaje del paso 3.
Prioridad:	Alta	
Frecuencia de uso:	Bajo demanda	
Comentarios:	N/A	

Tabla 10. Caso de uso 05: Gestionar reportes

CU-05:	Gestionar reportes	
Actor:	Administrador	
Descripción:	El sistema deberá permitir al administrador obtener e imprimir la información detallada de usuarios, bicicletas y repuestos para coordinar las actividades realizadas a través de reportes.	
Precondición:	Iniciar sesión en el sistema teniendo permisos de administrador.	
Secuencia Normal:	Paso	Acción
	1	Si el administrador desea el listado de los usuarios, en el menú del sistema, selecciona la opción "Usuarios".
		1.1 El sistema despliega la información de los usuarios registrados.
		1.2 El administrador da clic en "Imprimir Reporte".
		1.3 El sistema descarga un documento ".pdf" sobre el listado de los usuarios.
		1.4 El administrador imprime el reporte descargado.
	2	Si el administrador desea información de un usuario en específico sobre las bicicletas que ha utilizado, en el usuario deseado bajo la columna "Acciones" da clic en la imagen de una lupa.
		2.1 El sistema muestra un reporte con fecha y hora de todas las bicicletas usadas por el usuario.
		2.2 El administrador da clic en "Imprimir Reporte".
		2.3 El sistema descarga un documento ".pdf" sobre el detalle de uso de unidades.
		2.4 El administrador imprime el reporte descargado.
	3	Si el administrador desea el listado de las bicicletas, en el menú del sistema, selecciona la opción "Bicicletas".
		3.1 El sistema despliega la información de las bicicletas registradas.
		3.2 El administrador da clic en "Imprimir Reporte".
		3.3 El sistema descarga un documento ".pdf" sobre el listado de bicicletas.
		3.4 El administrador imprime el reporte descargado.
	4	Si el administrador desea información de una bicicleta en específico, en la bicicleta deseada bajo la columna "Acciones" da clic en la imagen de una lupa.

		4.1 El sistema muestra un reporte con fecha y hora de todos los usuarios que han utilizado esa bicicleta.
		4.2 El administrador da clic en "Imprimir Reporte".
		4.3 El sistema descarga un documento ".pdf" sobre el listado de usuarios que usaron esa unidad.
		4.4 El administrador imprime el reporte descargado.
	5	Si el administrador desea el listado del stock de repuestos, en el menú del sistema, selecciona la opción "Repuestos".
		5.1 El sistema despliega el inventario de repuestos.
		5.2 El administrador da clic en "Imprimir Reporte".
		5.3 El sistema descarga un documento ".pdf" sobre el listado de repuestos.
		5.4 El administrador imprime el reporte descargado.
Postcondición:	El administrador obtiene información detallada de usuarios, bicicletas y repuestos del sistema la cual puede ser impresa.	
Excepciones:	Paso	Acción
	N/A	N/A
Prioridad:	Alta	
Frecuencia de uso:	Bajo demanda	
Comentarios:	N/A	

Tabla 11. Caso de uso 06: Alquilar bicicleta

CU-06:	Alquilar bicicleta	
Actor:	Usuario	
Descripción:	El sistema deberá permitir al usuario reservar una bicicleta al ingresar a la aplicación web.	
Precondición:	Estar registrados en la base de datos del sistema. Iniciar sesión en el sistema.	
Secuencia Normal:	Paso	Acción
	1	El usuario da clic en el botón "Alquilar unidad".
	2	El sistema abre una ventana, para seleccionar la unidad.
	3	El usuario selecciona su nombre, y la bicicleta que desea alquilar y da clic en el botón "registrar".
	4	El sistema muestra información sobre la bicicleta reservada por el usuario.
	5	Si el usuario quiere cancelar la reserva, en la columna "Acciones", da clic logo de un basurero.
	6	El sistema elimina la reserva.
Postcondición:	El usuario visualiza todas las unidades que ha alquilado.	
Excepciones:	Paso	Acción
	N/A	N/A
Prioridad:	Media	
Frecuencia de uso:	Bajo demanda	
Comentarios:	N/A	

Tabla 12. Caso de uso 07: Alquilar bicicleta

CU-07:	Ingresar a mantenimiento	
Actor:	Mecánico	
Descripción:	El sistema deberá permitir al mecánico seleccionar la bicicleta que no se encuentra en buen estado para enviarla a mantenimiento.	
Precondición:	Estar registrados en el sistema con rol de bodega para que tenga permisos de mecánico.	

	Iniciar sesión en el sistema.	
Secuencia Normal:	Paso	Acción
	1	El mecánico ingresa al menú "Bicicletas"
	2	El sistema muestra un listado de todas las bicicletas con su id <i>tag</i> , marca, color, estado y acciones que se pueden realizar en ellas.
	3	El mecánico selecciona la bicicleta que va a ingresar a mantenimiento, en "Acciones" da clic en el logo de una llave inglesa.
	4	El sistema cambia el estado de esa bicicleta a "en reparación"
Postcondición:	El mecánico visualiza el listado de todas las bicicletas que han ingresado a mantenimiento.	
Excepciones:	Paso	Acción
	N/A	N/A
Prioridad:	Alta	
Frecuencia de uso:	Alta	
Comentarios:	N/A	

Tabla 13. Caso de uso 08: Poner en circulación

CU-08:	Poner en circulación	
Actor:	Mecánico	
Descripción:	El sistema deberá permitir al mecánico una vez que la bicicleta sea reparada ponerla en circulación para habilitarla a los usuarios según se describe en el siguiente caso de uso.	
Precondición:	Estar registrados en el sistema con rol de bodega para que tenga permisos de mecánico. Iniciar sesión en el sistema.	
Secuencia Normal:	Paso	Acción
	1	El mecánico ingresa al sistema, al menú "Bicicletas"
	2	El sistema muestra el listado de todas las bicicletas que se encuentran en reparación, habilitadas y alquiladas.
	3	El mecánico selecciona la bicicleta que ha sido reparada y que va a poner en funcionamiento, en "Acciones" da clic en el logo de un visto.
	4	El sistema abre una ventana para detallar el mantenimiento que se le ha hecho.
	5	El mecánico selecciona los repuestos utilizados en la bicicleta para que esté habilitada y da clic en el botón "poner en circulación"
	6	El sistema actualiza el estado de la bicicleta que estaba en reparación a habilitada.
Postcondición:	El mecánico visualiza el listado de todas las bicicletas con su estado actual.	
Excepciones:	Paso	Acción
	5	Si no se seleccionan los repuestos utilizados para que la bicicleta vuelva a circulación, el sistema mostrará el siguiente mensaje: "Seleccione un elemento de la lista"
Prioridad:	Alta	
Frecuencia de uso:	Alta	
Comentarios:	N/A	

Tabla 14. Caso de uso 09: Poner en circulación

CU-09:	Registrar entrada y salida de bicicletas	
Actor:	Sistema	
Descripción:	El sistema deberá permitir registrar la fecha y hora en la que las bicicletas entran o salen de la bodega.	
Precondición:	Las bicicletas deberán tener instalado un <i>tag</i> o etiqueta de identificación RFID. Los códigos de las etiquetas deberán estar registradas en el sistema.	
Secuencia Normal:	Paso	Acción
	1	El sistema lee la información de la etiqueta instalada en la bicicleta al pasar por el lector RFID
	2	El sistema registra la siguiente información de la bicicleta: la fecha, el id de la etiqueta o <i>tag</i> , la hora de salida y la hora de entrada.
	3	El sistema muestra un reporte sobre la información de las bicicletas.
Postcondición:	Se crean reportes para que puedan ser gestionados por el administrador del sistema.	
Excepciones:	Paso	Acción
	1	Si la bicicleta no pasa por el lector con su etiqueta, el sistema registrará ninguna información.
	1	La distancia de lectura entre el lector y la etiqueta debe ser de máximo 12cm para que se efectúe el intercambio de información.
Prioridad:	Alta	
Frecuencia de uso:	Alta	
Comentarios:	N/A	

3.4 Descripción de la Base de Datos

Las bases de datos están diseñadas con el fin de ofrecer un mecanismo organizado para almacenar, gestionar y recuperar gran cantidad de información de manera estructurada, con la menor redundancia permitiendo que la búsqueda y recuperación de los datos sea mucho más fácil y eficiente.

Entre los tantos sistemas gestores de bases de datos, MySQL funciona perfectamente para el desarrollo del prototipo, es ideal para la crear bases de datos para cualquier solución profesional que consista en guardar datos, con la posibilidad de ejecutar múltiples y rápidas consultas. (Cobo, 2005, pág. 339)

MySql funciona y trabaja perfectamente y de manera estable con PHP el cual es el lenguaje de programación utilizado para el desarrollo del sistema del prototipo de la aplicación web. La base de datos de este prototipo almacena toda la información de usuarios, bicicletas, reparaciones realizadas, repuestos

utilizados y bicicletas alquiladas con los campos con los que se relacionan para la administración de la aplicación web.

Tabla 15. Tabla usuarios

Tabla usuarios		
Clave	Columna	Tipo
Primaria	<u>id_usuario</u>	int(11)
	cedula	varchar(10)
	nombres	varchar (200)
	teléfono	varchar(15)
	email	varchar(120)
	rol	int(11)
	nickname	varchar(15)
	psw	varchar(10)
	estado	int(11)

Tabla 16. Tabla bicicletas

Tabla bicicletas		
Clave	Columna	Tipo
Primaria	<u>id_bicicleta</u>	int(11)
	id_tag	varchar(50)
	color	varchar (30)
	marca	varchar(30)
	estado	int(2)

Tabla 17. Tabla reparaciones

Tabla reparaciones		
Clave	Columna	Tipo
Primaria	<u>id_reparacion</u>	int(11)
	id_tag	varchar(50)
	Id_repuesto	int (11)
	cantidad	int (11)

Tabla 18. Tabla repuestos

Tabla repuestos		
Clave	Columna	Tipo
Primaria	<u>id_repuesto</u>	int(11)
	nombre	varchar(250)
	stock	Int(4)
	precio	varchar(5)
	estado	int(2)

Tabla 19. Tabla alquilar

Tabla alquilar		
Clave	Columna	Tipo
Primaria	<u>id_alquilar</u>	int(11)
	id_usuario	int(11)
	id_tag	varchar(50)
	fecha	varchar(10)
	hora_salida	varchar(10)
	hora_entrada	varchar(10)
	estado	int(2)

3.4.1 Diagrama general de la base de datos

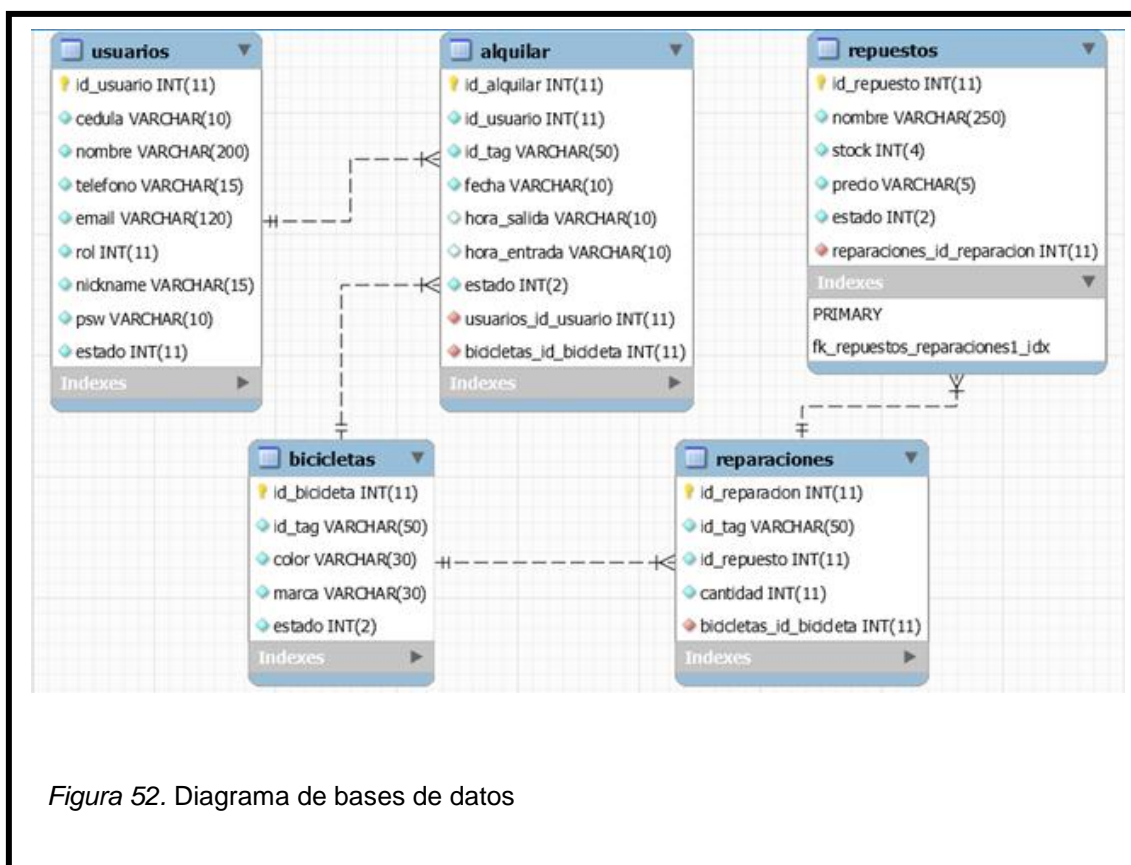


Figura 52. Diagrama de bases de datos

3.5 Desarrollo de interfaz de usuario

La interfaz web consta de una serie de elementos gráficos que facilitan al usuario el acceso, navegación e interacción eficaz con los contenidos.

El diseño de la interfaz se basa en ayudar al usuario a realizar una tarea dada como navegar por diferentes páginas del sitio, acceder a los contenidos, etc., lo más simple y eficiente como sea posible. El aspecto y la sensación es de innegable importancia, pero el núcleo de una gran interfaz de usuario es su función: en términos de navegación, esta deberá ser intuitiva hasta el punto de ser invisible. Si el usuario se pierde o no puede interactuar con ella, la interfaz de usuario ha fallado. (Carson, 2014)

Para el desarrollo de la aplicación web se ha tomado en cuenta el rol principal de los actores de la bodega de BiciQ, por lo que se han desarrollado tres tipos de interfaces en la cuales unas poseen más contenido que otras según los permisos que se les otorgue. Las tres interfaces mencionadas son para: el administrador del sistema, el mecánico de la bodega de BiciQ y para los usuarios que deseen alquilar una bicicleta. Se realizar pruebas con el usuario para verificar que las interfaces son factibles y de fácil uso para garantizar el éxito de la aplicación final. El ingreso al sistema debe ser a través de un inicio de sesión en el que se digita el nombre de usuario y la contraseña.

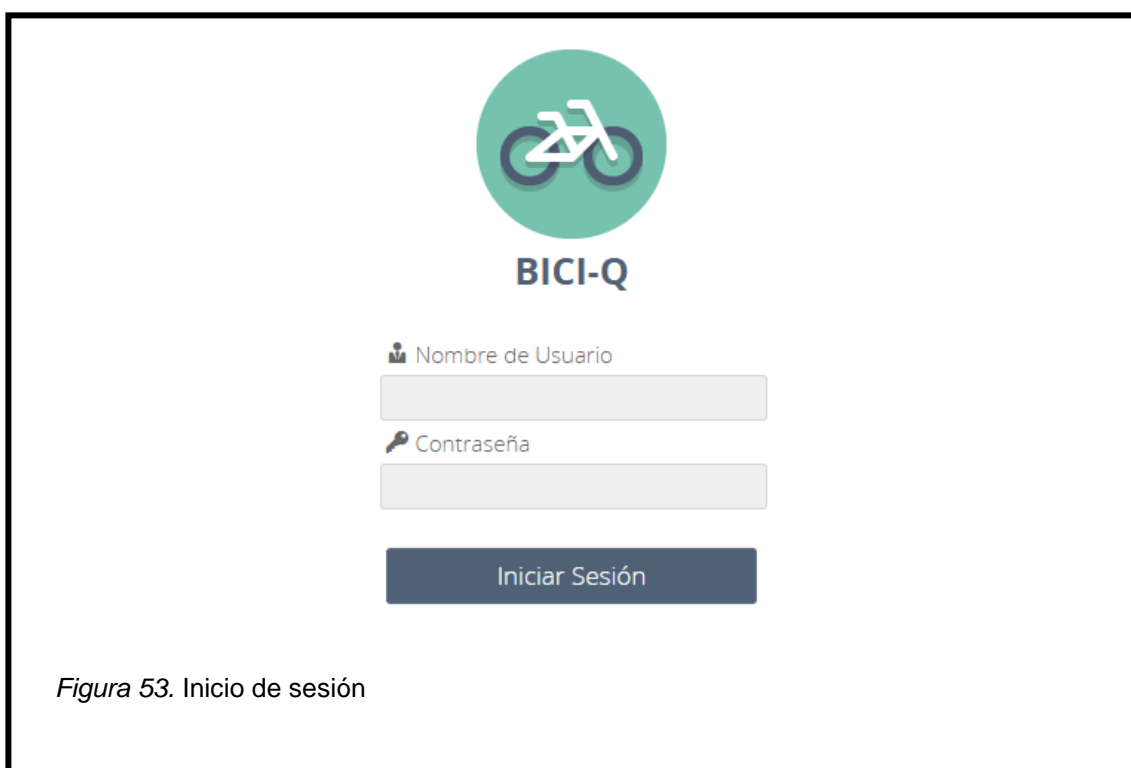


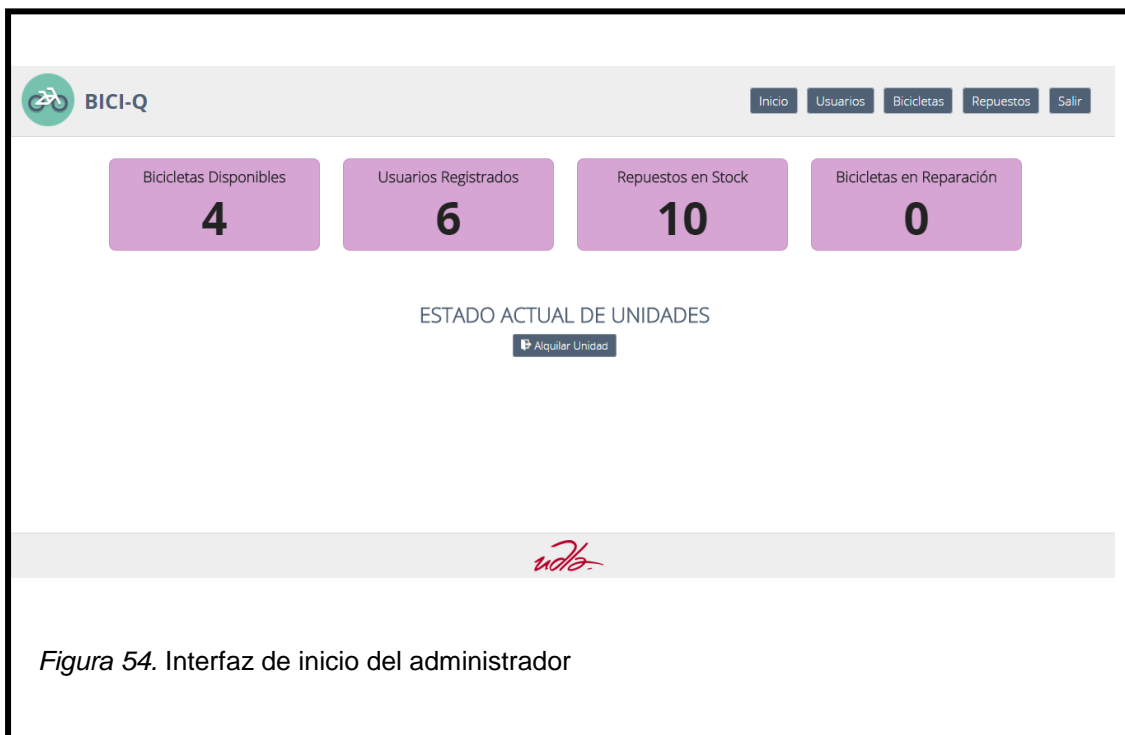
Figura 53. Inicio de sesión

Después de iniciar sesión, aparece la interfaz de inicio, la cual es diferente para el administrador, mecánico y usuario general.

3.5.1 Interfaz de Administrador

El administrador tiene privilegios que cualquier otro usuario no posee, estos privilegios le permiten gestionar y conservar la información que es ingresada al

sistema de BiciQ para el control de entrada y salida de bicicletas en bodega y de inventario.



Al iniciar sesión, la pantalla de “Inicio” del menú es la primera interfaz que visualiza el administrador, en ella se detallan las bicicletas que se encuentran disponibles en la bodega, el número de usuarios registrados en el sistema, número de repuestos en stock, y el número de bicicletas que se encuentran en reparación.

Adicionalmente el administrador puede ver el estado actual de las unidades (bicicletas) cuando un usuario alquila una desde el sistema; ya cuando la bicicleta sale de la bodega registra la hora de salida y de entrada a través de la etiqueta RFID que posee. Automáticamente la información de las bicicletas disponibles se actualiza.

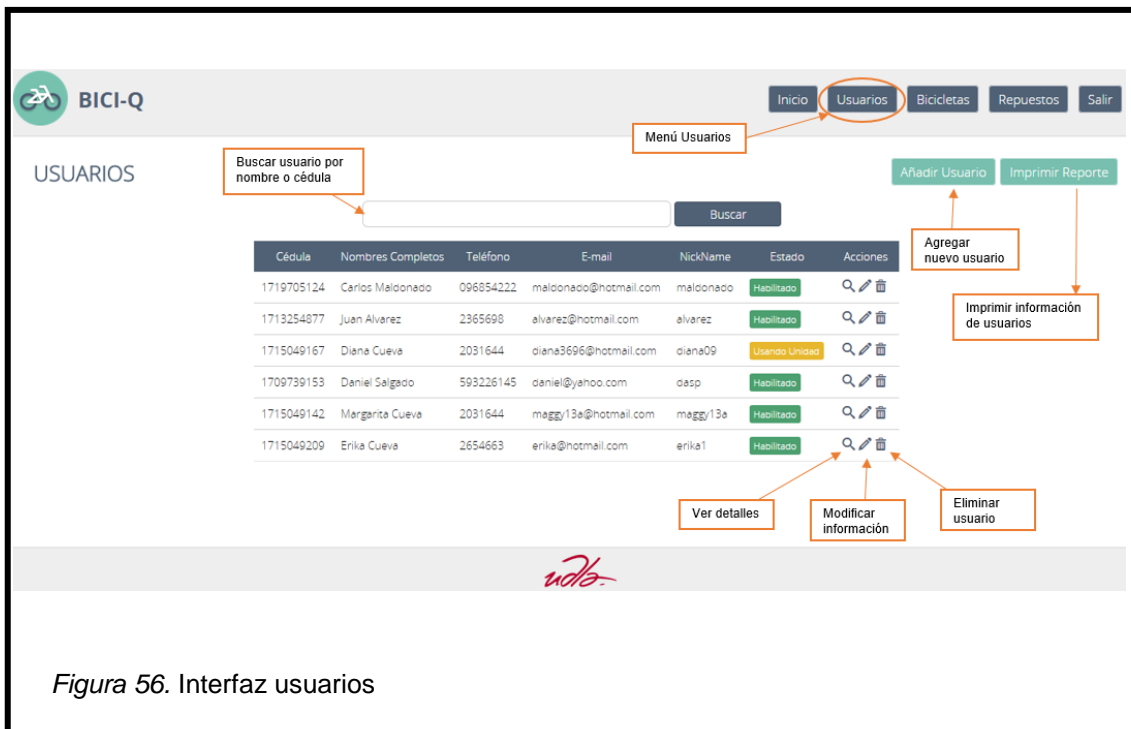


El menú del administrador cuenta con los botones para administrar usuarios, bicicletas, repuestos y salir del sistema, además del inicio que es la primera pantalla que se despliega al iniciar sesión.

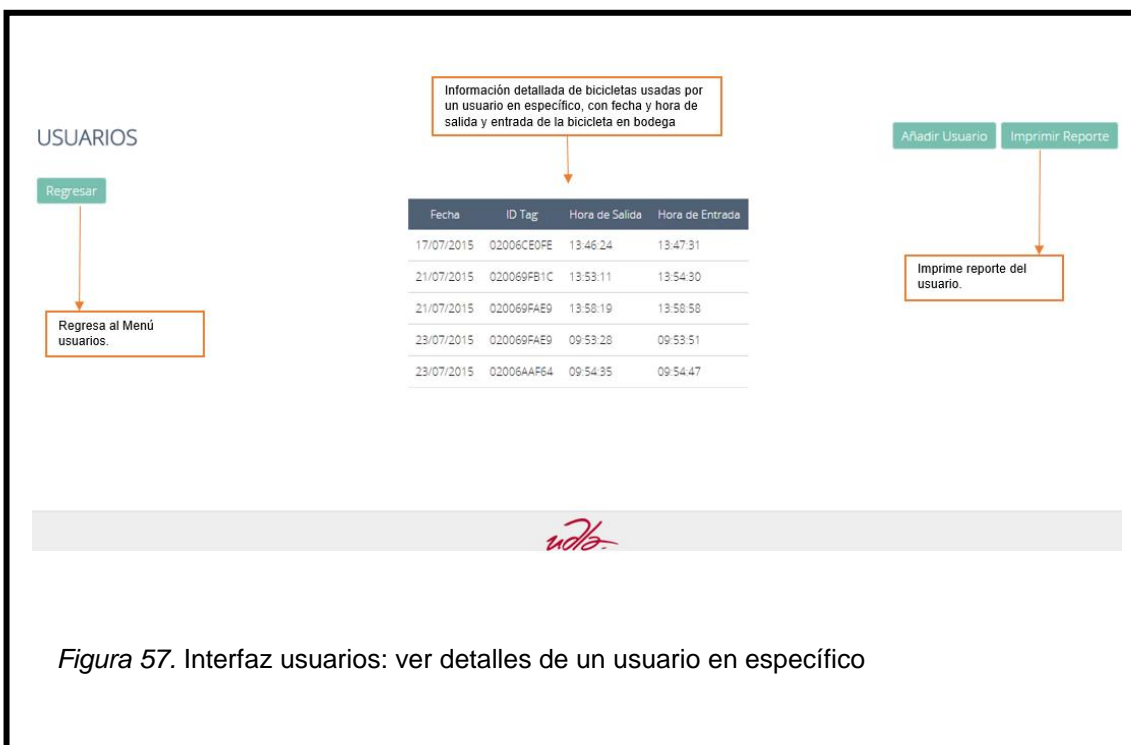
MENÚ USUARIOS

En el botón “Usuarios” el administrador visualiza el reporte de todos los usuarios que se encuentran registrados en el sistema. Se puede añadir, buscar, modificar y eliminar usuarios, imprimir reportes de todos los usuarios. Al dar clic en “Imprimir Reporte” el sistema genera un documento en formato “.pdf” para que pueda ser impreso.

El administrador al añadir un nuevo usuario, especifica el rol que le dará para el ingreso al sistema: administrador, mecánico o usuario general.



Si se quiere tener información de un usuario en específico, se da clic en “Ver detalles”. El sistema mostrará la información de las bicicletas utilizadas por el con la fecha y hora tanto de salida como de entrada en bodega.



MENÚ BICICLETAS

En el botón “Bicicletas” el administrador visualiza el reporte de todas las bicicletas registradas en el sistema y que operan para el servicio que ofrece BiciQ a sus usuarios. El administrador es el único que tiene permisos para añadir, buscar, modificar y eliminar bicicletas e imprimir reportes de las mismas. Al dar clic en “Imprimir Reporte” el sistema genera un documento en formato pdf para que pueda ser impreso.

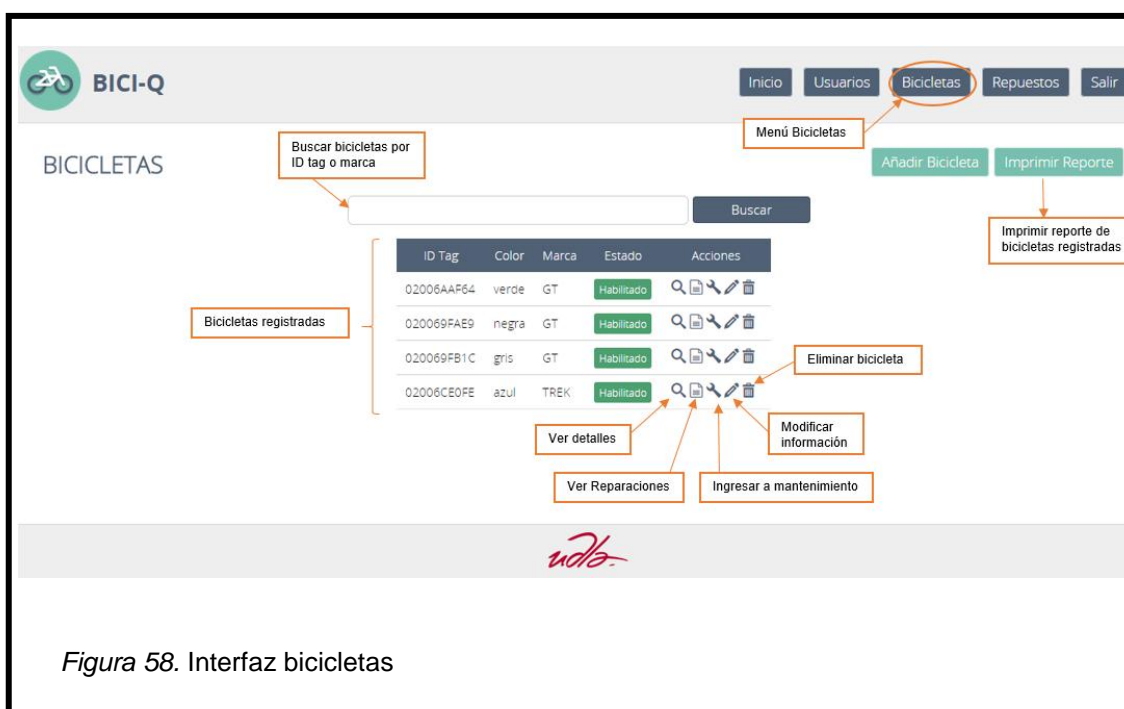
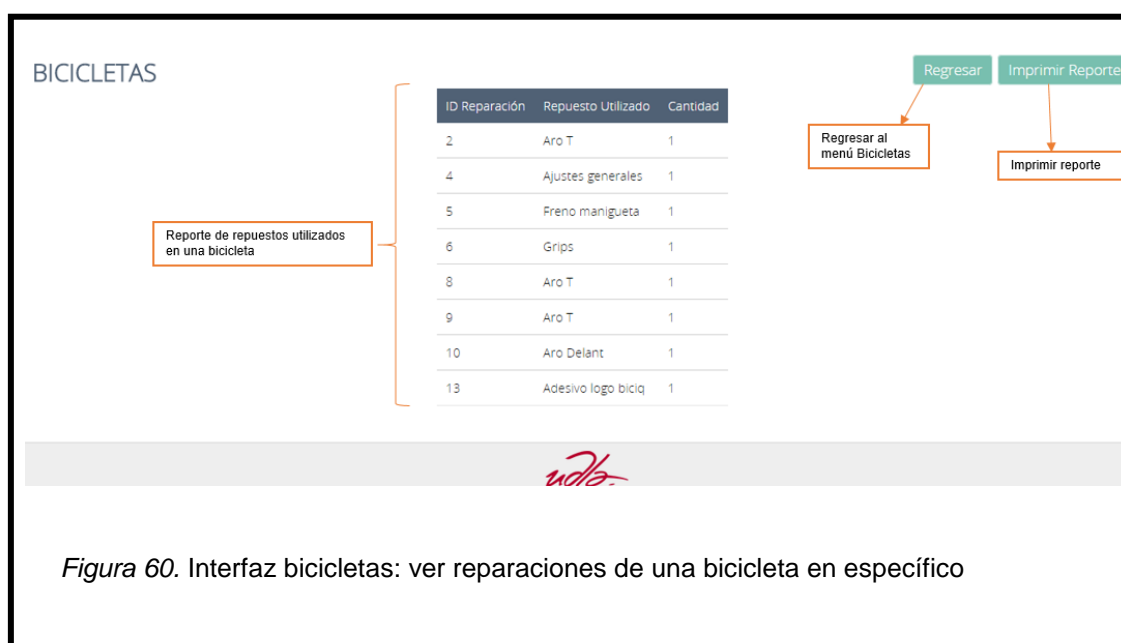


Figura 58. Interfaz bicicletas

Adicionalmente, el administrador cuenta con la posibilidad de ver imprimir la información detallada de cada bicicleta al dar clic en “Ver detalles”, se despliega una pantalla con un registro/reporte de todos los usuarios que han utilizado dicha bicicleta.



Si el administrador da clic en “Ver reparaciones”, se despliega una nueva pantalla con un reporte de todos repuestos utilizados en esa bicicleta para que pueda volver a funcionamiento, de igual manera se puede imprimir el reporte. La opción de “Ingresar a mantenimiento” es una función del mecánico, pero está habilitada al administrador ya que éste posee todos los privilegios de administración.



MENÚ REPUESTOS

En el botón “Repuestos” el administrador visualiza el reporte de todos los repuestos registrados en el sistema con su descripción, stock y precio, gracias

a ello le permite conocer cuánto producto tiene en stock en bodega. Se puede añadir, buscar, modificar y eliminar repuestos e imprimir reportes de los mismos. Al dar clic en “Imprimir Reporte” el sistema genera un documento en formato pdf para que pueda ser impreso.

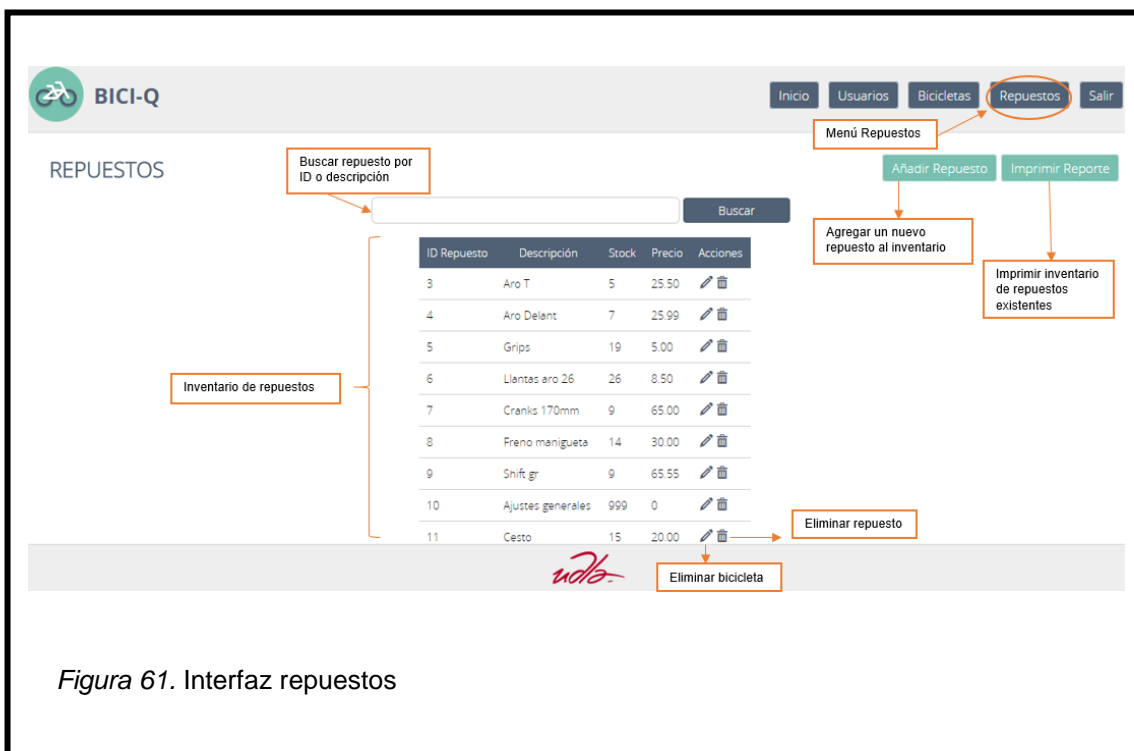


Figura 61. Interfaz repuestos

3.5.2 Interfaz de Mecánico

Al contrario del administrador, el mecánico tiene habilitado solamente privilegios que van de la mano con la reparación de alguna bicicleta que se encuentra en mal estado. El inicio de sesión es normal para todos los que ingresan al sistema, deben digitar su nombre de usuario y su contraseña.

Al iniciar sesión, la pantalla de “Inicio” del menú es la primera que visualiza el mecánico. En ella se detallan el número de bicicletas que se encuentran disponibles, número de registro de repuestos en el inventario y el número de bicicletas que se encuentran en reparación. Así mismo, en la parte inferior de la pantalla, se visualiza un pequeño reporte con información detallada de todas las unidades que se encuentran en reparación.



Figura 62. Interfaz de Inicio del mecánico

El menú del mecánico cuenta con los botones: bicicletas, repuestos y salir del sistema, además del inicio que es la primera pantalla que se despliega al iniciar sesión.

MENÚ BICICLETAS

Cuando una bicicleta llega a la bodega de BiciQ en malas condiciones y necesita que le hagan alguna reparación, el mecánico ingresa al menú “Bicicletas”, ahí aparece la lista de todas las bicicletas que se encuentran registradas en el sistema de BiciQ, busca y selecciona la bicicleta que va a ingresar a mantenimiento, inmediatamente el estado de la bicicleta cambia a “en reparación”.

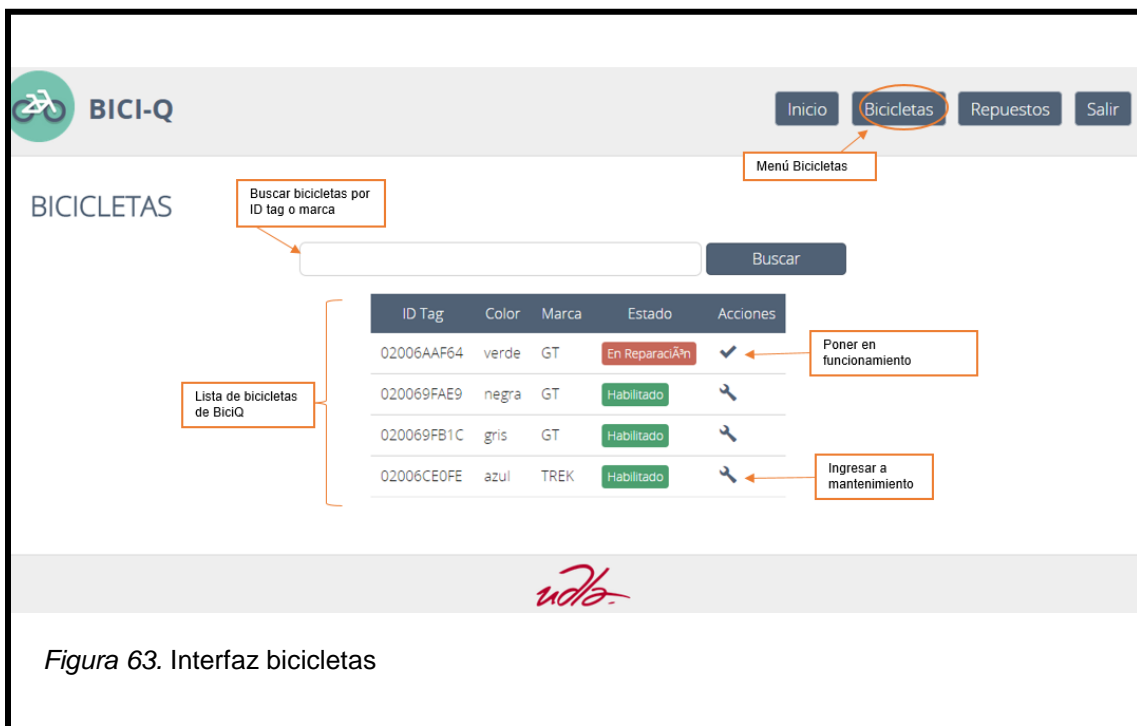


Figura 63. Interfaz bicicletas

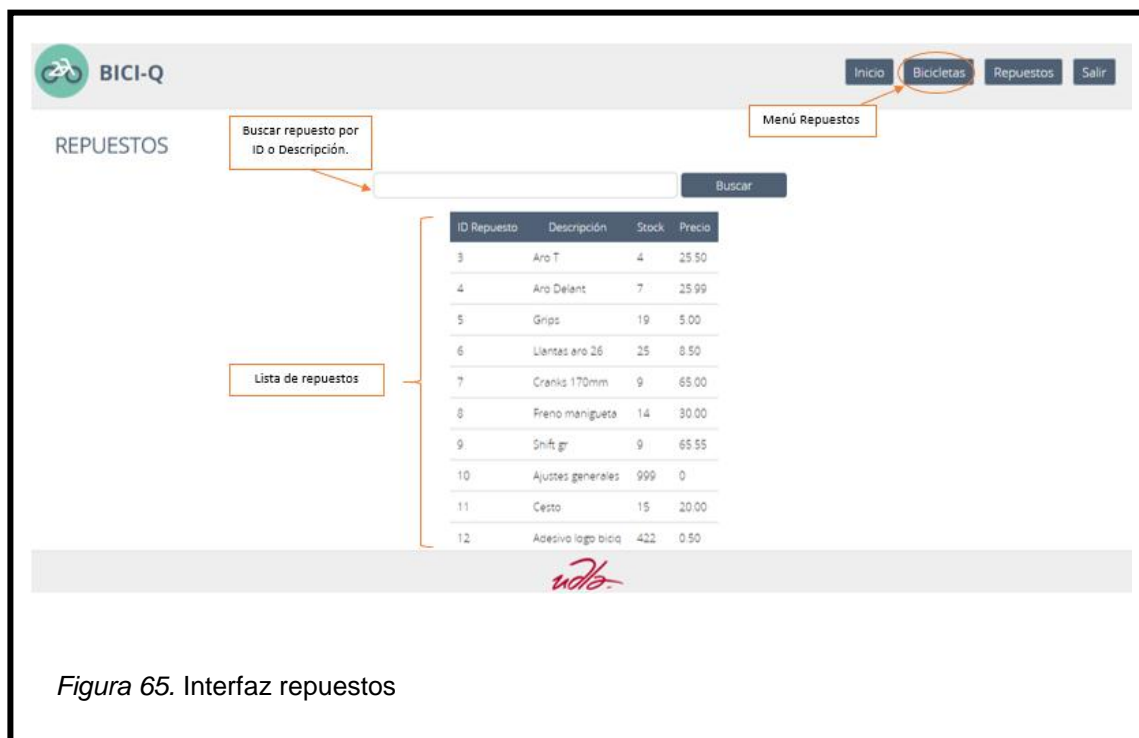
Cuando la bicicleta ya está reparada, el mecánico busca y selecciona la bicicleta que reparó, y da clic en “poner en funcionamiento”. Le aparece ventana para detallar los repuestos utilizados en esa bicicleta para que se encuentre habilitada.



Figura 64. Interfaz bicicletas: poner en funcionamiento

MENÚ REPUESTOS

En este menú, el mecánico puede visualizar todos los repuestos que se encuentran en el inventario, así mismo puede conocer su stock y precio pero no puede modificar su información ni borrarlos ya que de la administración de inventarios se encarga el administrador del sistema.



3.5.3 Interfaz de Usuario

La interfaz de usuario está creada netamente para facilitar al usuario el alquiler/reserva de una bicicleta que se encuentra disponible en el sistema y en la bodega de BiciQ.

Al iniciar sesión, se le despliega una pantalla en la que le muestra el número de bicicletas disponibles y el número de bicicletas que se encuentran en reparación.

Para reservar una bicicleta, el usuario da clic en el botón "Alquilar unidad".



En el lado izquierdo de la pantalla le aparece una ventana para seleccionar la bicicleta que se desea reservar. La descripción de cada bicicleta (unidades disponibles) consta del id del *tag* (identificador de la etiqueta) y la marca de bicicleta. Al seleccionar la unidad, el usuario finaliza su reserva al dar clic en el botón "Registrar".



Una vez reservada la bicicleta, en la parte inferior de la pantalla, el usuario puede ver el estado actual de la unidad reservada, y si desea, también puede eliminar la reserva realizada.

The screenshot displays the BICI-Q user interface. At the top left is the BICI-Q logo, and at the top right is a 'Salir' button. Below the header are three purple boxes showing the status of units: 'Bicicletas Disponibles' with the number 3, 'Repuestos en Stock' with the number 10, and 'Bicicletas en Reparación' with the number 0. The main section is titled 'ESTADO ACTUAL DE UNIDADES' and includes a 'Alquilar Unidad' button. Below this is a table with the following data:

Fecha	Nombre del Usuario	Tag Unidad	Hora de Salida	Estado
19/08/2015	Diana Cueva	02006AAF64	00:00:00	Reservada

Annotations in the image include a box labeled 'Detalle de bicicleta reservada' pointing to the first row of the table, and another box labeled 'Eliminar reserva' pointing to the trash icon next to the 'Reservada' status.

Figura 68. Interfaz usuario: estado actual de reserva

4. PRUEBAS Y COSTO DEL PROTOTIPO

Al finalizar la construcción del prototipo, se realizan pruebas de funcionamiento tanto del hardware como del software para comprobar que los resultados obtenidos cumplen y satisfacen los objetivos y requisitos funcionales del sistema.

4.1 Pruebas de Hardware

A cada bicicleta se le acopla una etiqueta que posee un código único para que al salir o entrar de la bodega el lector RFID capte la información que ésta le envía, para ello, al lector se lo sitúa estratégicamente tomando en cuenta las características de funcionalidad para que la distancia de lectura con las etiquetas sea válida, aproximadamente 12 cm.

Como al lector se lo diseñó con un Arduino, no requiere estar conectado a un computador, simplemente se lo conecta con una fuente de alimentación externa de 12V, pero si se desea conectarlo al computador, se lo debe alimentar con un cable USB para impresora. Además, gracias al *Ethernet Shield* conectado al *router* con un cable de red, se conecta a internet para almacenar la información recibida la cual se la visualiza en la aplicación web desarrollada.

Tabla 20. Pruebas y resultados del hardware del prototipo

Pruebas	Resultados	Observaciones
Conexión del lector con alimentación externa 12V.	Al conectar el lector se encendió normalmente, se le aproximó una etiqueta para la lectura y recibió la información deseada.	N/A
Conexión del lector al computador mediante cable USB	El lector recibe la información de las etiquetas, dicha información puede visualizarse en la consola de la plataforma de arduino.	N/A

Lectura de etiquetas (tarjeta PVC)	<p>Son efectivas en la lectura para el lector diseñado.</p> <p>El lector lee el código de las etiquetas hasta una distancia máxima de 12cm.</p>	<p>Operan a una frecuencia de 125 KHz, por lo tanto son perfectas con respecto a las características propias del lector.</p> <p>Si la distancia entre la etiqueta o el lector supera los 12 cm la lectura no es posible.</p> <p>Se realiza un pequeño programa y se lo carga en el arduino para verificar la lectura de las etiquetas.</p>
Funcionamiento del programa cargado en el Arduino.	<p>El programa captó la información que emitieron las etiquetas al lector independientemente cual sea la etiqueta de pvc que se le aproximó.</p>	<p>Se conecta el lector al computador para cargar el programa final en el arduino.</p>

4.2 Pruebas de Software

El desarrollo del software del prototipo realizado permite tener un sistema online de inventario y un control automatizado de la entrada y salida en bodega de las bicicletas de BiciQ a través de la tecnología por identificación por radiofrecuencia. Las interfaces web cumplen con la simplicidad y coherencia necesaria para el manejo del sitio web por parte del usuario logrando que se sienta cómodo y satisfecho cada vez que ingresa al sistema.

A continuación se detallan las pruebas ejecutadas en la aplicación web basadas en los casos de uso realizadas para la elaboración del sistema.

Tabla 21. Pruebas y resultados del software del prototipo

Caso de uso / situación a probar	Descripción de la prueba	Resultado esperado	Resultado	
			Inicial	final
CU-01: Iniciar Sesión	Pantalla de inicio de sesión.	Al ingresar al dominio aparece la pantalla de inicio de sesión.	✓	✓
	Muestra mensaje al no ingresar un campo necesario.	Aparece mensaje: Completa este campo.	x	✓
	Ingresa al sistema sin importar el navegador que se utilice.	Funciona en Internet Explorer, Mozilla Firefox y Chrome	✓	✓
	Muestra mensaje de error al ingresar datos inválidos.	No ingresa al sistema y muestra el Mensaje: Sus credenciales son inválidas. Por favor, inténtelo de nuevo	✓	✓
CU-02,03,04: Administrar usuarios, bicicletas y repuestos	Administrador crea, modifica, elimina, busca usuarios, bicicletas y repuestos.	El botón agregar usuario, bicicleta y repuesto permite agregarlos en el sistema inmediatamente. Las opciones para eliminar, buscar y modificar cumplen su propósito perfectamente.	x	✓
	Mensajes de error al agregar o modificar un usuario, bicicleta o repuesto.	Aparece mensaje: Completa este campo y no se agrega o modifica hasta que el campo este completo.	x	✓
CU-05: Gestionar Reportes	Le permite al administrador imprimir reportes de bicicletas usadas, usuarios que usaron bicicletas, lista	Al dar clic en el botón Imprimir reporte se genera un documento “.pdf” para imprimirlo.	✓	✓

	de repuestos, etc.			
CU-06: Alquilar bicicleta	Un usuario ingresa al sistema y reserva una bicicleta para utilizarla.	El usuario alquila una bicicleta, le aparecen las bicicletas disponibles y reserva la bici que desea usar.	✓	✓
		El usuario puede eliminar la reserva.	x	✓
CU-07: Ingresar a mantenimiento	El mecánico al ingresar al sistema. Le aparece la opción para mandar una bicicleta dañada a reparación, da clic ahí y la bicicleta cambia de estado a “en reparación”	Le aparece una llave de tuercas que manda a reparación a la bicicleta.	✓	✓
CU-08: Poner en circulación	El mecánico habilita la bicicleta en reparación y selecciona los repuestos utilizados.	Le aparece un visto que abre una ventana para seleccionar los repuestos utilizados y poner la bicicleta en circulación.	✓	✓
CU-09: Poner en circulación	El sistema registra el paso de la bicicleta por la bodega	Se registra automáticamente la fecha y hora en la que sale y entra la bicicleta de la bodega.	x	✓
Función del sistema en la UDLA	Se prueba el prototipo en la universidad y al pasar las etiquetas por el lector se registra la información.	El prototipo funciona con la red de la universidad.	x	✓
Interfaces de usuario	Las interfaces de la aplicación web son amigables y de fácil uso para el usuario.	Interfaces sencillas, funcionales y amigables para el usuario	✓	✓

4.3 Resultados índice de Satisfacción

Mediante una pequeña encuesta realizada a alrededor de 76 personas se evaluó el índice de satisfacción del cliente sobre el servicio que ofrece el prototipo creado.

El desarrollo de esta encuesta evaluó la satisfacción del cliente con respecto a la calidad y funcionalidad del servicio ofrecido lo que permitirá tener una mejor administración y control de logística a la bodega de BiciQ. Antes de realizar la encuesta, los encuestados tuvieron una breve capacitación del funcionamiento del sistema lo que les permitió poder responder las preguntas.

Tabla 22. Encuesta sobre índice satisfacción

Preguntas	Respuestas	Porcentaje
1.- ¿Qué tan importante es el conocer el estado de las bicicletas públicas de BiciQ?	Extremadamente importante: 32 Muy importante: 42 Un poco importante: - Ligeramente importante: 2 Nada importante: -	42% 55% 0% 3% 0%
2.- ¿Qué tan importante es realizar consultas sobre el historial de una bicicleta?	Extremadamente importante: 28 Muy importante: 32 Un poco importante: 16 Ligeramente importante: - Nada importante: -	37% 42% 21% 0% 0%
3.- ¿Qué tan útil considera usted que son las herramientas tecnológicas con RFID ofrecidas para el control de inventario y entrada y salida de bicicletas?	Extremadamente útil: 24 Muy útil: 46 Un poco útil: 6 Ligeramente útil: - Nada útil: -	32% 60% 8% 0% 0%
4.- Al ingresar al sistema ¿Considera a la interfaz de usuario de su agrado, intuitiva y de fácil uso?	Si: 76 No: -	100% 0%
5.- Califique la calidad general del sistema ofrecido	Excelente: 48 Muy buena: 28 Regular: - Pobre: -	63% 37% 0% 0%

6.- ¿En general, cómo ha sido su experiencia utilizando nuestro sistema?	Excelente: 38 Muy buena: 38 Regular: - Pobre: -	50% 50% 0% 0%
7.- ¿Considera que el sistema ayuda a optimizar tiempos y recursos empleados para las actividades cotidianas en bodega de BiciQ?	Si: 76 No: -	100% 0%

A continuación se muestran los resultados obtenidos en la encuesta a través de los siguientes gráficos de torta o circulares.

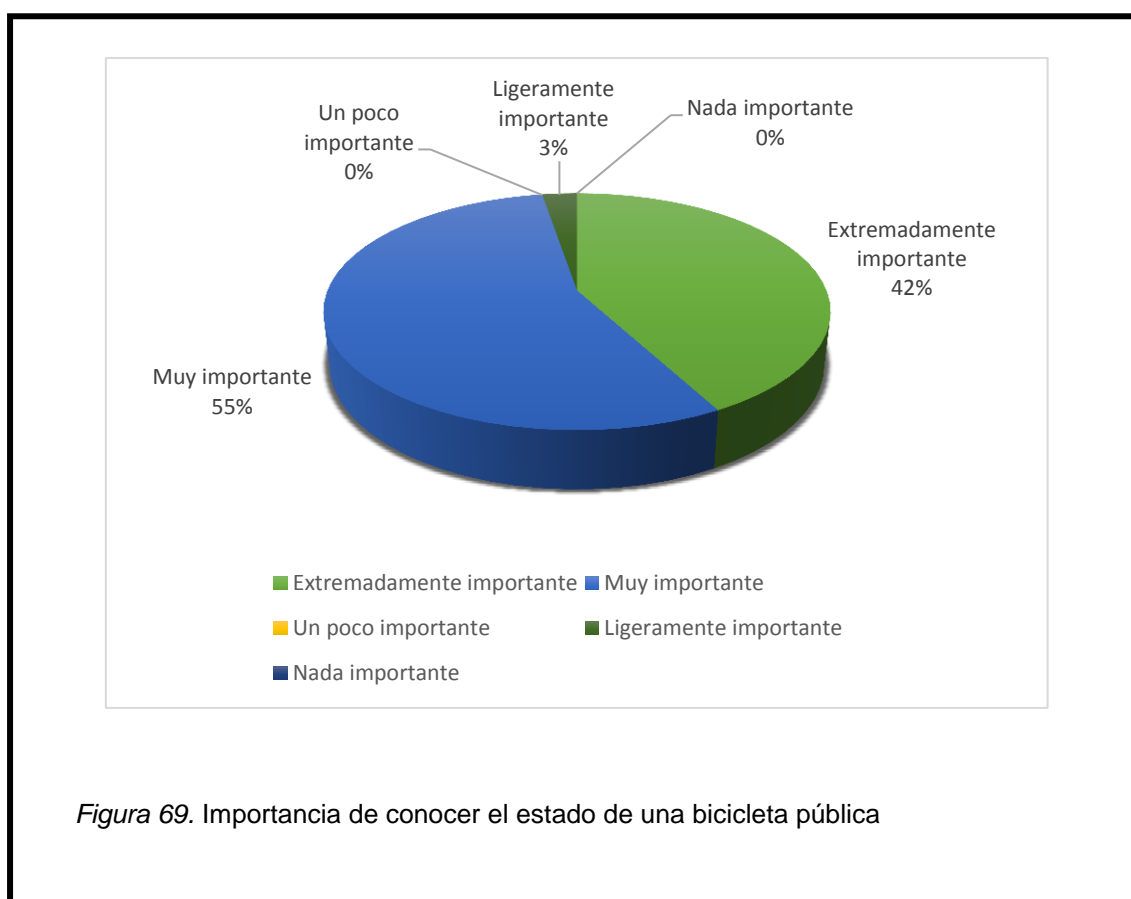




Figura 70. Importancia de realizar consultas sobre el historial de una bicicleta

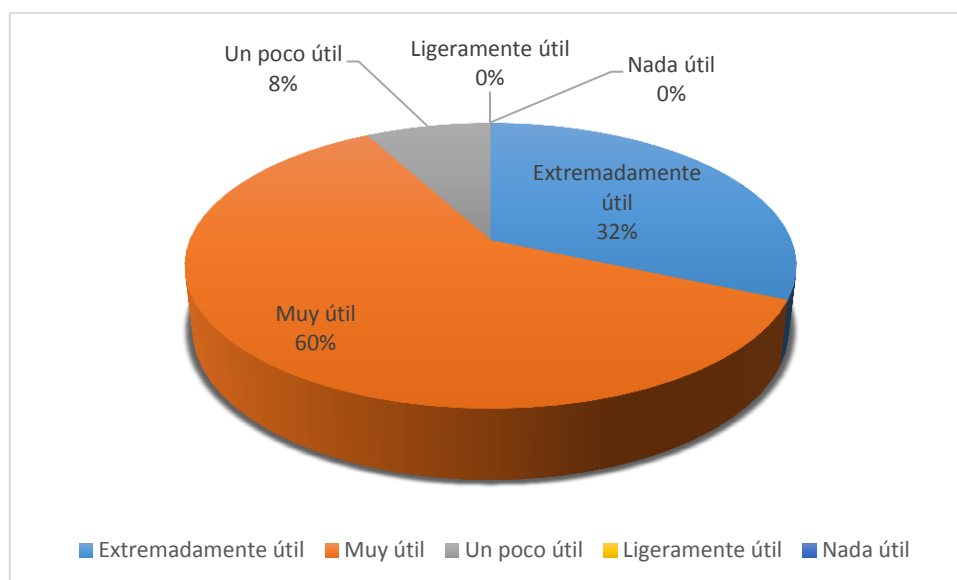


Figura 71. Grado de utilidad de las herramientas tecnológicas con RFID ofrecidas para el control de inventario y entrada y salida de bicicletas

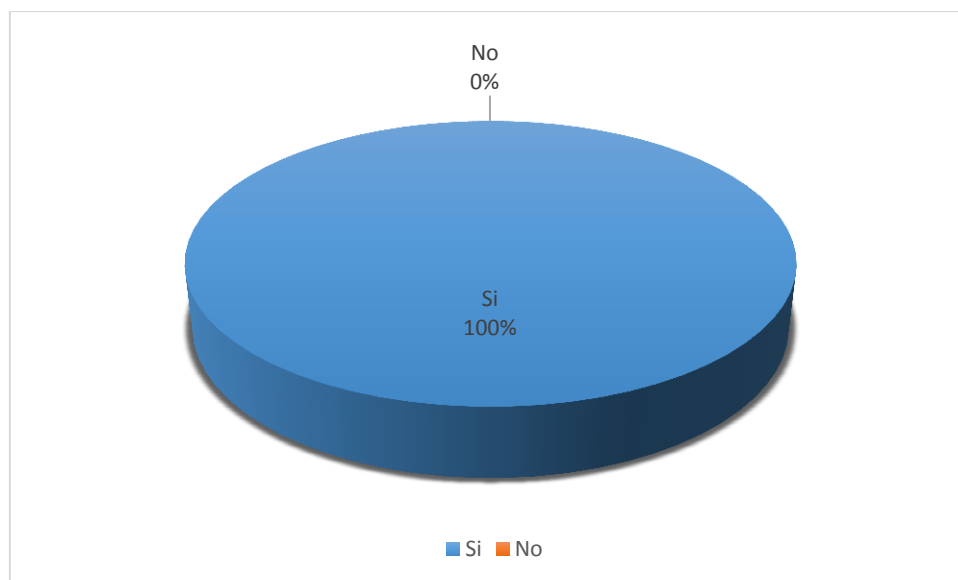


Figura 72. Interfaz de usuario agradable, intuitivo y de fácil uso

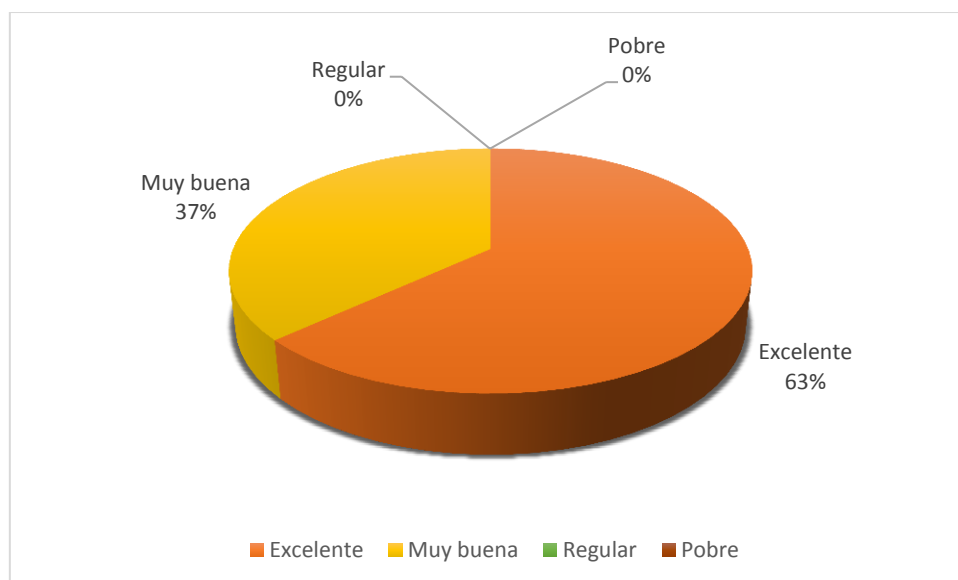


Figura 73. Calidad del sistema ofrecido

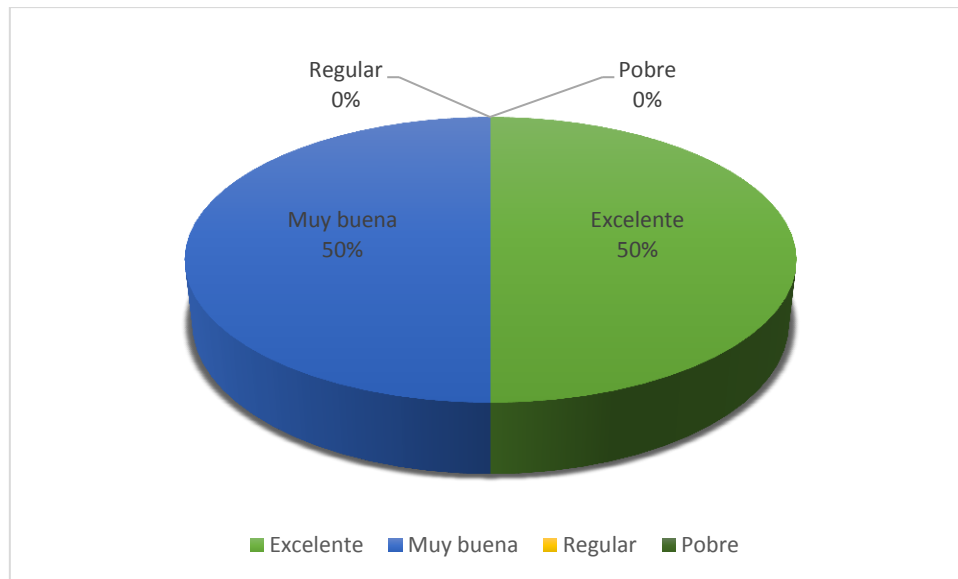


Figura 74. Experiencia de encuestados al usar el sistema

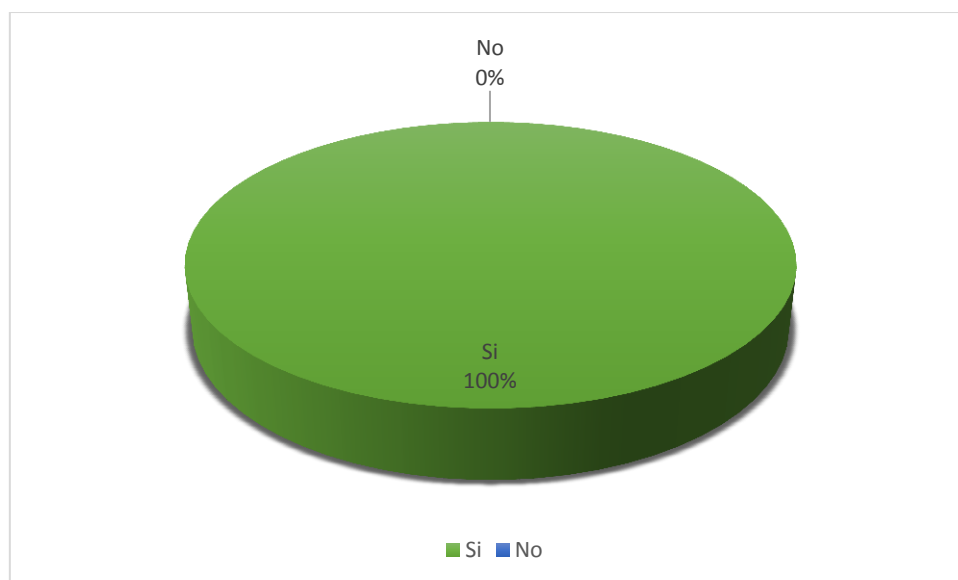


Figura 75. El sistema optimiza tiempos y recursos empleados para las actividades cotidianas en bodega de BiciQ

De acuerdo a los resultados obtenidos en la encuesta realizada, se confirma favorablemente que gran parte de los encuestados están satisfechos con el prototipo propuesto para el control de inventario online y de entrada y salida de bicicletas públicas BiciQ en bodega mediante tecnología por identificación de radiofrecuencia y consideran en su totalidad que este servicio ayudará a la optimización de tiempos y recursos empleados en la logística de la empresa lo que garantiza el éxito de este proyecto.

4.4 Análisis de Costos

Las siguientes tablas detallan el costo en dólares para la elaboración del prototipo de BiciQ, los cuales se los ha dividido en dos partes:

- Costos del prototipo
- Costos del software de administración

En los costos del prototipo se detallan todos los materiales necesarios y recursos empleados para la elaboración del prototipo.

Tabla 23. Costo del prototipo

Elemento/Item	Cantidad	Precio unitario	Precio total
Lector ID-12LA Innovations	1	\$ 42,86	\$ 42,86
Placa para el lector ID12LA	1	\$ 1,43	\$ 1,43
Etiquetas pasivas (tarjetas identificativas de PVC)	4	\$ 2,14	\$ 8,56
Led	1	\$ 0,09	\$ 0,09
Buzzer	1	\$ 0,58	\$ 0,58
Resistencias	2	\$ 0,02	\$ 0,04
Transistor 2N3904 NPN 40V	1	\$ 0,07	\$ 0,07
Conectores	3	\$ 0,15	\$ 0,45
Arduino Uno	1	\$ 22,00	\$ 22,00
Ethernet Shield	1	\$ 25,00	\$ 25,00
Cable USB impresora	1	\$ 6,00	\$ 6,00
Cable de red	1	\$ 2,50	\$ 2,50
Adaptador pared	1	\$ 10,00	\$ 10,00
Mano de obra	1	\$ 354,00	\$ 354,00
Total			\$ 473,58

En los costos del software de administración se colocan todos los costos por producción de software de control de datos que envía el prototipo.

Tabla 24. Costo del software de administración

Item	Cantidad	Precio unitario	Precio total
Servicio WEB/dominio	1	\$ 200,00	\$ 200,00
Servidor Virtual/hosting	1	\$ 600,00	\$ 600,00
Conectividad	1	\$ 200,00	\$ 200,00
Total			\$ 1.000,00

Adicionalmente a estas dos partes principales que se han tomado en cuenta para realizar el análisis de costos del sistema, se realiza la tabla de costos de instalación del lector y del sistema.

Tabla 25. Costo de instalación

Item	Precio total
Configuración del lector por punto	\$ 320,00
Puesta en marcha del sistema	\$ 340,00
Total	\$ 660,00

Tomando en cuenta que el sistema se lo implementase a escala real para un total de 658 bicicletas, se tomaría ese número de bicicletas para la compra de etiquetas más el 20% de reserva o prevención, lo que da un total de 790 etiquetas detalladas en la siguiente tabla.

Tabla 26. Costo del sistema para todas las bicicletas

Item	Cantidad	Precio unitario	Precio total
Etiquetas (658) + 20% backup	790	\$ 2,14	\$ 1.690,60
Precio total			\$ 1.690,60

El mantenimiento del sistema se lo realizaría cada 6 meses y se detalla el costo en la siguiente tabla.

Tabla 27. Costo de mantenimiento cada 6 meses

Periodo	Cantidad	Precio
cada 6 meses	1	\$ 250,00
Total		\$ 250,00

La siguiente tabla detalla el costo del soporte del sistema que se le brindaría a la empresa, se define el valor del monitoreo tomando en cuenta 24 horas por los 7 días a la semana por 365 días al año a \$1,5 la hora y el soporte en sitio considerando que se lo necesite 5 veces al mes a \$10 por visita.

Tabla 28. Costo de soporte

Item	Horas	Precio unitario	Horas x Precio unitario	Total al mes
Monitoreo	62000	\$ 1,50	\$ 93.000,00	\$ 129,17
Soporte en sitio	5	\$ 10,00	\$ 50,00	\$ 250,00
Total				\$ 379,17

Finalmente se desea ganar un 45% sobre el costo de producción del prototipo, por lo que la siguiente tabla resume el costo final del proyecto con RFID para BiciQ.

Tabla 29. Resumen de costos del prototipo para BiciQ

Item	Costo	45% de ganancia	Total
Costo del prototipo	\$ 473,58	\$ 213,11	\$ 686,69
Costo del software de administración	\$ 1.000,00	\$ 450,00	\$ 1.450,00
Costo del sistema para todas las bicicletas	\$ 1.690,60	\$ 760,77	\$ 2.451,37
Costo de instalación	\$ 660,00	\$ 297,00	\$ 957,00
Costo de soporte	\$ 379,17	\$ 170,63	\$ 549,79
Costo mantenimiento cada 6 meses	\$ 250,00	\$ 112,50	\$ 362,50

El proyecto tendría una inversión inicial de un solo pago de \$4.095,06 que corresponden a la suma del costo del prototipo, costo del sistema para todas las bicicletas y costo de instalación, mientras que el costo del software de administración y de soporte hacen referencia a un costo mensual de aproximadamente \$2000 y finalmente el costo de mantenimiento que anualmente se reflejaría en \$725.

Por lo tanto, el costo del proyecto incluyendo los \$4.095,06 dólares en el primer año sería de \$28.817,56 mientras que el segundo y tercer año sería de \$24.722,50 que corresponden a los costos del sistema de administración, soporte y mantenimiento ya antes mencionado.

Tabla 30. Costo anual de Sistema, soporte y mantenimiento

Concepto	Costo Mensual	Costo Anual
Sistema + Soporte	\$ 1.999,79	\$ 23.997,50
Mantenimiento	\$ 0,00	\$ 725,00
total		\$ 24.722,50

4.4.1 Cuadro comparativo de costos

El Municipio del distrito metropolitano de Quito destina anualmente a la administración de BiciQ un presupuesto aproximado de \$3.276.870 dólares, de los cuales utiliza \$7.228,26 mensualmente para una parte de la administración de la logística la cual se detalla a continuación.

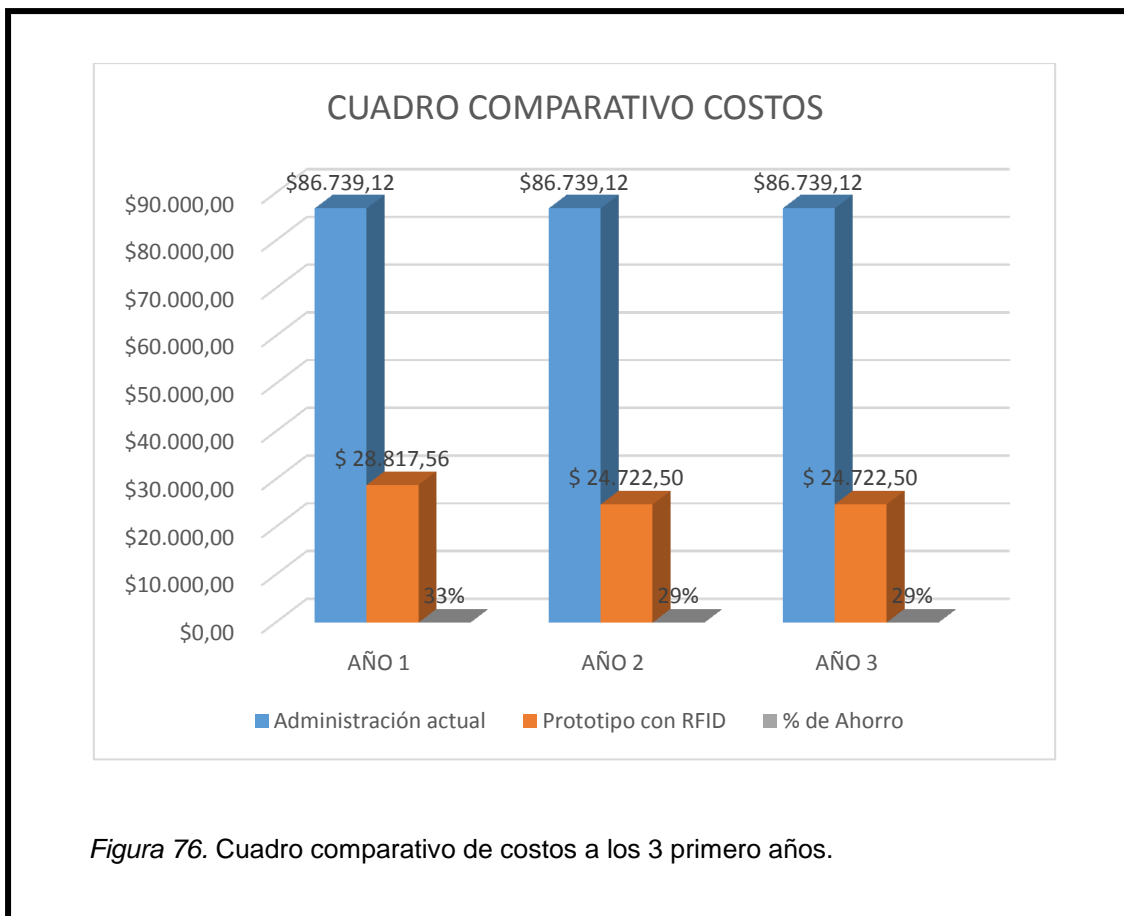
Tabla 31. Administración actual sin RFID

Concepto	Costo Mensual	Costo Anual
Suministros de oficina	\$ 500,00	\$ 6.000,00
Contador	\$ 600,00	\$ 7.200,00
Fotos, copias y reproducciones	\$ 80,00	\$ 960,00
Hosting/dominio	\$ 1.200,00	\$ 14.400,00
Herramientas y repuestos	\$ 4.848,26	\$ 58.179,12
Total presupuesto	\$ 7.228,26	\$ 86.739,12

A continuación, un cuadro comparativo entre el presupuesto asignado en la actualidad en el año (Tabla 31), y el presupuesto propuesto del prototipo para BiciQ con RFID.

Tabla.32 Cuadro comparativo de costos

	CUADRO COMPARATIVO COSTOS		
	Administración actual	Prototipo con RFID	% de Ahorro
AÑO 1	\$ 86.739,12	\$ 28.817,56	33%
AÑO 2	\$ 86.739,12	\$ 24.722,50	29%
AÑO 3	\$ 86.739,12	\$ 24.722,50	29%



Lo que nos permite evidenciar que el porcentaje de ahorro en el primer año que propone el prototipo con relación a gastos de la administración actual es de un 33%, lo que permite hacer un análisis de los materiales como suministros de oficina, herramientas y repuestos que se utilizan actualmente. Y del 29% en el costo de disminución o ahorro a partir del 2do año en adelante.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El prototipo realizado permite realizar un seguimiento online de las bicicletas que salen y entran de la bodega a través de la tecnología por Identificación de radiofrecuencia RFID, lo que da como resultado un mayor control de stock de inventario, de repuestos utilizados en las bicicletas al encontrarse en reparación, una mejor administración de la logística de la bodega, entre otros.
- El proyecto desarrollado resulta ser mucho más aplicable, 100% automatizable, funcional y así mismo costoso al elaborarlo en escala real con equipos de mejores características operacionales capaces de realizar la lectura de las etiquetas a una distancia mucho más efectiva que los 12cm en los que opera el prototipo realizado.
- Las ventajas del prototipo al poseer la tecnología de identificación por radiofrecuencia frente a otras tecnologías ya existentes para la identificación y seguimiento de objetos como lo es el código de barras son mucho más evidentes debido a que la captura de los datos se la realiza de forma automática brindando mayores beneficios a las empresas como la reducción de costos, de tiempo y errores de registros de entrada, de inventarios físicos, reducción de pérdidas y extravíos de mercadería, etc, incrementando los niveles de control y manejo avanzado de la información.
- El prototipo desarrollado en este proyecto de tesis es escalable pues flexiblemente se adapta a las condiciones variantes que supone un sistema que constantemente está en crecimiento como lo es el servicio de transporte que ofrece BiciQ, ya que tranquilamente se puede incrementar el número de bicicletas a través de nuevas etiquetas registrando el id en el sistema.

- El módulo lector ID-12LA satisface las necesidades referentes a protocolos de comunicación y funciona perfectamente para la elaboración del prototipo.
- La elaboración de interfaces web sencillas y amigables para el usuario en el software garantizan el éxito del sistema elaborado pues permiten una manipulación fácil de la información y contenido que se maneja en la aplicación web.
- La constante ejecución de pruebas tanto técnicas como del funcionamiento de la aplicación web garantizan la efectividad del trabajo realizado permitiendo la corrección de errores presentados inicialmente.
- Es mucho más rentable tener un sistema automatizado para el control de las bicicletas públicas ya que optimiza recursos y reduce exigencias humanas sensoriales y mentales.
- El correcto uso de herramientas de software libre como PHP y la base de datos MySQL permiten el desarrollo de un sistema web dinámico beneficiando el costo del prototipo gracias al ahorro del licenciamiento.

5.2 Recomendaciones

- Este prototipo es un pequeño módulo que podría ser integrado a un sistema de control y automatización completo que abarque todas las actividades que maneja BiciQ.
- Optar por un lector de mejores características operacionales que cubra la zona y distancia de lectura que requiera el sistema implementándolo a escala real para la captura de información.

- Realizar un análisis de requerimientos adecuado que abarquen todas las necesidades del cliente con respecto al sistema a desarrollar con el fin de que garantice el resultado deseado.
- Para una mejor manipulación de los materiales a utilizar en la elaboración del lector del prototipo se recomienda leer la información de datasheet con el fin de evitar errores y gastos innecesarios.
- Realizar constantes pruebas al prototipo para asegurar el correcto funcionamiento del mismo.
- El prototipo realiza la lectura de 4 bicicletas a través de las etiquetas de PVC, para aumentar el número de bicicletas en el sistema se debe agregar al programa cargado en el arduino líneas de código fuente que validen el identificador único de las nuevas etiquetas a adicionar.
- Se recomienda que la distancia de lectura entre la etiqueta y el lector no sobrepase los 12 cm debido a las características propias del lector con el fin de evitar errores en la transmisión de los datos.
- Se recomienda que el administrador del sistema asigne correctamente los roles y permisos a los usuarios que también interactuarán en el sistema con el fin de evitar accesos no autorizados a la información ya que no debe ser manipulada ni visualizada por cualquier persona.

REFERENCIAS

- Arduino. (s.f.). *ARDUINO UNO*. Recuperado el 28 de julio del 2015 de <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- BICIQ. (s.f.). *¿Qué es BiciQ?*. Recuperado el 15 de abril de 2015 de: <http://www.biciquito.gob.ec/index.php/info/que-es.html>
- Bil提高. (s.f.). *ID-20 Lector RFID + Arduino*. Recuperado el 18 de agosto del 2015 de <http://bildr.org/2011/02/rfid-arduino/>
- Bricogeek. (s.f.). *RFID Reader ID-12LA* . Recuperado el 15 de mayo del 2015 de <http://tienda.bricogeek.com/modulos-radiofrecuencia/262-rfid-reader-id-12.html>
- Carson, N. (2014). *11 Great UI designs*. Recuperado el 19 de agosto del 2015 de <http://www.creativebloq.com/web-design/examples-ui-design-7133429>
- Cobo, Á. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. España: Editorial Díaz de Santos, S.A.
- Cortoc. (s.f.). *ARDUINO. Pines de Arduino* Recuperado el 28 de julio del 2015 de <http://www.cortoc.com/p/arduino.htm>
- Dataflows. (s.f.). *AN OVERVIEW OF RFID TECHNOLOGY*. Recuperado el 25 de marzo del 2015 de http://www.dataflows.com/RFID_Overview.shtml
- Datamark. (s.f.). *RADIO FRECUENCIA RFID*. Recuperado el 26 de marzo del 2015 de <http://www.datamark.es/identificacion-por-radio-frecuencia-rfid.html>
- Dipolerfid. (s.f.). *Antenas RFID*. Recuperado el 24 de septiembre de 2015 de: <http://www.dipolerfid.es/Productos/Antenas-RFID/Default.aspx>
- García, J., Escamilla, L., Álvarez, O. y Mayoral, M. (2013). *Protocolos Anticolisión en RFID*. Recuperado el 6 de enero de 2016 de: <http://revistatelematica.cujae.edu.cu/index.php/tele/article/viewFile/84/84>
- Glover, B. y Bhatt, H. (2006). *RFID Essentials*. (1ª ed.) Estados Unidos: O'Reilly Media, Inc.

- González, E. (2015). *Tutorial básico del programador web: PHP desde cero..* Recuperado el 29 de julio del 2015 de http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=515:icomo-escribir-codigo-php-y-crear-webs-dinamicas-remoto-o-local-wamp-xampp-lamp-cu00807b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193
- ID-Innovations. (s.f.). *ID-2LA/12LA/20LA Series Datasheet*. Recuperado el 16 de mayo del 2015 de <http://cdn.sparkfun.com/datasheets/Sensors/ID/ID-2LA,%20ID-12LA,%20ID-20LA2013-4-10.pdf>
- Impinj. (s.f.). *The different types of RFID Systems*. Recuperado el 15 de octubre de 2015 de: <http://www.impinj.com//resources/about-rfid/the-different-types-of-rfid-systems/>
- Instruelecom. (s.f.). *Lector RFID UHF FX-9500 Motorola*. Recuperado el 24 de septiembre de 2015 de: <http://www.instruelecom.com/portal/index.php/productos/rfid/lectoras/23-lector-fx9500>
- ISO. (s.f.). *Standards*. Recuperado el 16 de mayo del 2015 de <http://www.iso.org/iso/home/standards.htm>
- Libera. (2010). *RFID: Tecnología, Aplicaciones y Perspectivas*. Recuperado el 26 de Marzo del 2015 de http://www.libera.net/uploads/documents/whitepaper_rfid.pdf
- Lin, I., Hsu, H. y Cheng, C. (2014). *A Cloud-based Authentication Protocol for RFID Supply Chain Systems*. Springer US, p. 978-997
- Montañana, R. (s.f.). *Redes locales*. Recuperado el 15 de octubre de 2015 de: <http://slideplayer.es/slide/106511/>
- Mundonfc. (s.f.). *Diferencias entre NFC y RFID*. Recuperado el 26 de marzo del 2015 de <https://mundonfc.wordpress.com/2012/02/08/diferencia-entre-nfc-y-rfid/>
- Ozguven, E.E. y Ozbay, K. (2012). *An RFID-based inventory management framework for emergency relief operations*. IEEE, p. 1274-1279

- PC-Doctor. (s.f.). *La tecnología de Identificación por Radio Frecuencia (RFID)*. Recuperado el 15 de abril de 2015 de: [http://www.pc-doctor.com.mx/Radio%20Formula/temas/RFID%20ETIQUETAS%20DE L%20FUTURO.htm](http://www.pc-doctor.com.mx/Radio%20Formula/temas/RFID%20ETIQUETAS%20DE%20L%20FUTURO.htm)
- Person.wanadoo. (s.f.). *La comunicación serie*. Recuperado el 18 de mayo del 2015 de <http://perso.wanadoo.es/pictob/comserie.htm>
- PHP. (s.f.). *QUE PUEDE HACER PHP?*. Recuperado el 29 de julio del 2015 de <http://php.net/manual/es/intro-whatcando.php>
- Porras, J. (2012). *Espectro electromagnético y ondas electromagnéticas*. Recuperado el 26 de marzo del 2015 de <https://grupo2radiobiologiayradioproteccion2.wordpress.com/category/un-categorized/page/5/>
- Portillo, J., Bermejo, A. y Bernardos, A. (2013) *Tecnología de identificación por radiofrecuencia (RFID): Aplicaciones en el ámbito de la salud*. Madrid: CEIM Confederación Empresarial de Madrid – CEOE
- Priority1design. (s.f.). *EM4100 Protocol description*. Recuperado el 15 de octubre de 2015 de: http://www.priority1design.com.au/em4100_protocol.html
- Prototypexpress. (s.f.). *RFID Technology White Paper*. Recuperado el 26 de marzo del 2015 de <http://www.prototypexpress.com/rfidwhitepaper.htm>
- Reyes, H., Vacca, A. y Góngora, A. (2011). *Diseño de un sistema de identificación electrónica y su potencial uso en la trazabilidad de la carne bovina*. Recuperado el 26 de Marzo del 2015 de http://www.scielo.org.co/scielo.php?pid=S0121-37092011000100010&script=sci_arttext
- Ruíz, F. (2013). *LA TECNOLOGÍA RFID Y EL “INTERNET DE LOS OBJETOS”*. Recuperado el 25 de marzo del 2015 de <http://ferranruiz.net/la-tecnologia-rfid-y-el-internet-de-los-objetos/>
- Skyrfid. (s.f.). *Antenas RFID Reader Antenna Tutorial*. Recuperado el 24 de septiembre de 2015 de: http://skyrfid.com/RFID_Antenna_Tutorial.php
- Toaha, A., Tariqul, M. y Misran, N. (2011). *RFID Technology: Perspectives and Technical Considerations of Microstrip Antennas for Multi-Band RFID*

Reader Operation, Current Trends and Challenges in RFID. Recuperado el 26 de marzo del 2015 de <http://www.intechopen.com/books/current-trends-and-challenges-in-rfid/rfid-technology-perspectives-and-technical-considerations-of-microstrip-antennas-for-multi-band-rfid>

Transcore. (2001). *SHROUDS OF TIME THE HISTORY OF RFID. The decades of RFID.* Recuperado el 25 de marzo del 2015 de <https://www.transcore.com/sites/default/files/History%20of%20RFID%20White%20Paper.pdf>

Vellingiri, S., Ray, A. y Kande, M. (2013). *Wireless Infrastructure for Oil and Gas Inventory Management.* IEEE, p. 5461-5466

Visualstudio. (s.f.). *VISUAL STUDIO EXPRESS.* Recuperado el 22 de mayo del 2015 de <https://www.visualstudio.com/es-es/products/visual-studio-express-vs.aspx>

Wampserver. (s.f.). *WAMPSSERVER. A Windows web development environment.* Recuperado el 29 de julio del 2015 de <http://www.wampserver.com/en/>

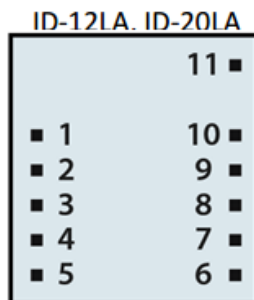
ANEXOS

ANEXO 1. Datasheet Módulo lector ID-12LA

1. Overview

ID2-LA, ID12-LA and the ID20-LA series are small footprint 2.8-5.0volt reader modules that support ASCII, Wiegand26 and Magnetic ABA Track2 data formats. The modules are pin and function compatible with the ID2/12/20 series.

2. Pin Out for ID12-LA and ID20-LA



Bottom View

1. GND
2. RES (Reset Bar)
3. NC
4. NC
5. CP
6. Tag in Range
7. Format Selector
8. D1 (Data Pin 1)
9. D0 (Data Pin 0)
10. Read (LED / Beeper)
11. +2.8V thru +5.0V



3. Device Operational Characteristics

Parameter	ID-2LA, ID-12LA, ID-20LA
Frequency	125 kHz nominal
Card Format	EM 4001 or compatible
Read Range ID3	Up to 30 using suitable antenna using ID-Innovations clamshell card @5v
Read Range ID13	Up to 12cm using ISO card, up to 18cm using ID-Innovations clamshell card @5v
Read Range ID23	Up to 18cm using ISO card, up to 25cm using ID-Innovations clamshell card @5v
Encoding	Manchester 64-bit, modulus 64
Power Requirement	+2.8 VDC thru +5 VDC @ 35mA ID-12LA, 45mA ID-20LA
RF I/O Output Current	+/- 200mA PKPK

4. Data Formats

Output Data Structure - ASCII - 9600 Baud, No Parity, 1 stop bit.

Output = CMOS (Push Pull) 0-Vdd

STX (02h)	DATA (10 ASCII)	CHECK SUM (2 ASCII)	CR	LF	ETX (03h)
-----------	-----------------	---------------------	----	----	-----------

Example for Calculation of Checksum for ASCII

Suppose the output Data is 0C000621A58E

Here the actual data is 0C,00,00,06,21,A5 and the checksum is 6E

Using binary we Exclusive OR the bit columns

0C = 00001100
 00 = 00000000
 06 = 00000110
 21 = 00100001
 A5 = 10100101
 CHECKSUM 10001110 (8E)

Output Data Structure - Wiegand26 – 1mS repeat, 50uS pulse. Open Drain

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
P	E	E	E	E	E	E	E	E	E	E	E	E	O	O	O	O	O	O	O	O	O	O	O	O	P
Even parity (E)													Odd parity (O)												

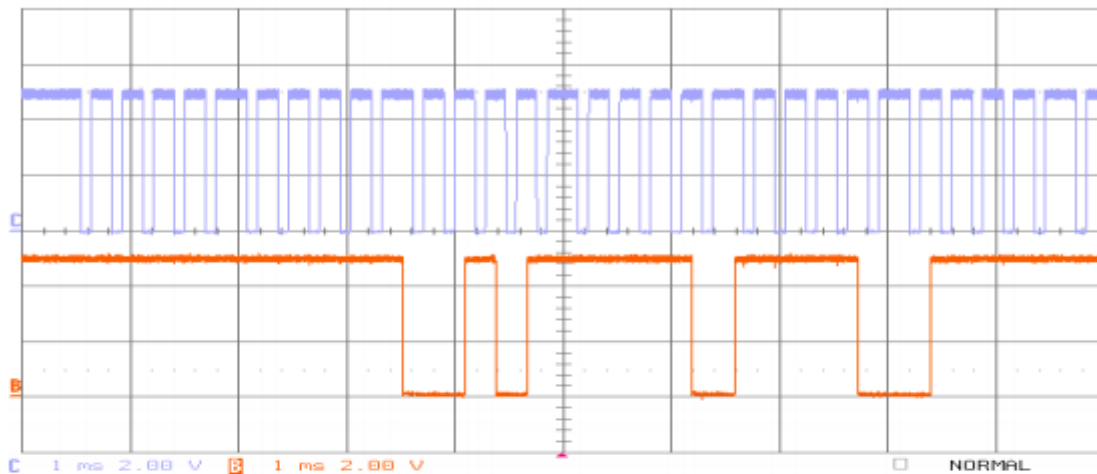
P = Parity start bit and stop bit

Output Data Magnetic ABA Track2 – At Approx. 80cm/sec. Open Drain

10 Leading Zeros	SS	Data	ES	LCR	10 Ending Zeros
------------------	----	------	----	-----	-----------------

[SS is the Start Character of 11010, ES is the end character of 11111, and LRC is the Longitudinal Redundancy Check.]

5. Magnetic Emulation Waveforms



Blue = Clock, Brown = Data

Fig. 3

Start and End Sequences for Magnetic Timing

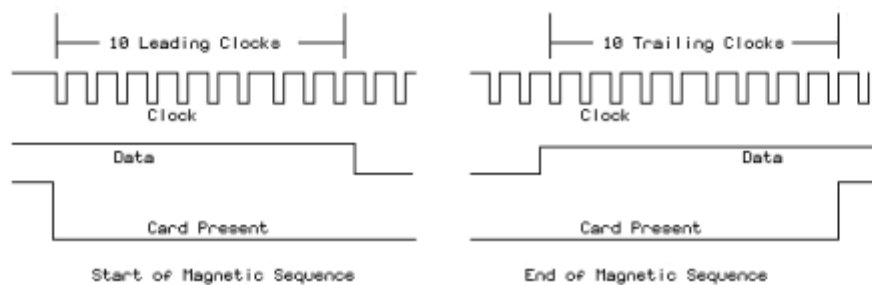


Fig. 4

Data Timings for Magnetic Emulation

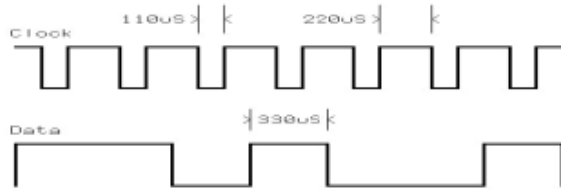


Fig. 5

The magnetic Emulation Sequence starts with the Card Present Line going active (down). There next follows 10 clocks with Zero '0' data. At the end of the 10 leading clocks the start character (11010) is sent and this is followed by the data. At the end of the data the end character is sent followed by the LCR. Finally 10 trailing clocks are sent and the card present line is raised.

The data bit duration is approximately 330µs. The approximate clock duration is 110µs. Because of the symmetry data can be clocked off either the rising or falling edge of the clock.

6. Pin Description & Output Data Formats

Pin #	Description	ASCII	Magnet Emulation	Wiegand26
Pin 1	Zero Volts	GND 0V	GND 0V	GND 0V
Pin 2	Strap to Pin11	Reset Bar	Reset Bar	Reset Bar
Pin 3	To External Antenna ID-2LA only	Antenna	Antenna	Antenna
Pin 4	To External Antenna ID-2LA only	Antenna	Antenna	Antenna
Pin 5	Card Present	No function	Card Present*	No function
Pin 6	Tag in Range (Future)	Tag in Range	Tag in Range	Tag in Range
Pin 7	Format Selector (+/-)	Strap to GND	Strap to Pin 10	Strap to +5V
Pin 8	Data 1	CMOS	Clock*	One Output*
Pin 9	Data 0	TTL Data (inverted)	Data*	Zero Output*
Pin 10	3.1 kHz Logic	Beeper / LED	Beeper / LED	Beeper / LED
Pin 11	DC Voltage Supply	+2.8 thru 5V	+2.8V thru 5V	+2.8V thru 5V

* Requires 4K7 Pull-up resistor to +5V

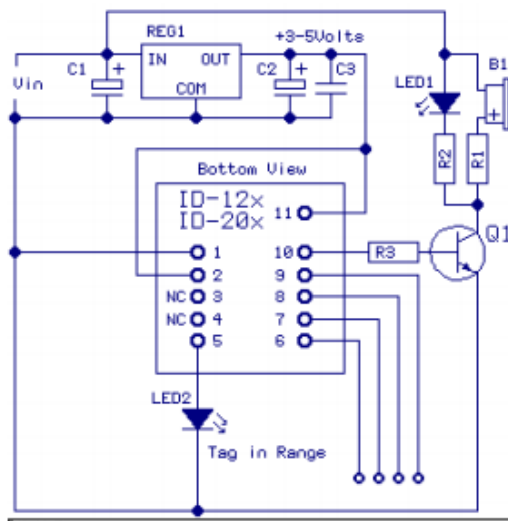
7. Absolute Maximum Ratings

Maximum voltage applied to Pin 2	(Vcc)	5.5volt
Maximum voltage applied to Pin 2	(Reset)	Vcc + 0.7v, -0.7v
Maximum current drawn from Pin 3 (Antenna)	+/- 75mA
Maximum 125 KHz RF Voltage at Pin 4	(Antenna)	+/- 80volt Peak
Maximum current drawn from Pin 5	(Card Present)	+/- 5mA
Maximum current drawn from Pin 6	(Tag in Range)	+/- 5mA
Maximum Voltage at Pin 7	(Format Selector)	Vcc + 0.7v, -0.7v
Maximum current drawn from Pin 8	(Data1)	+/- 5mA
Maximum current drawn from Pin 9	(Data0)	+/- 5mA
Maximum current drawn from Pin 10	(Beeper)	+/- 10mA
Additionally, Pins 5, 6, 7, 8, 9 & 10 may not have a voltage exceeding		Vcc + 0.7v, -0.7v

These ratings are absolute maximums and operation at or near the maximums may cause stress and eventual damage or unpredictable behaviour.

8. Circuit Diagram

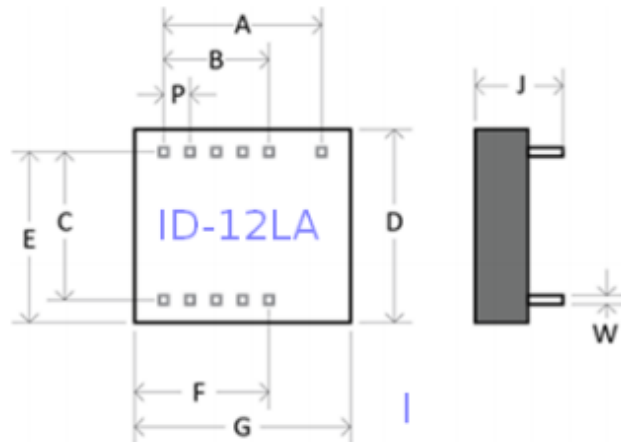
8.1 Circuit Diagram for ID-12LA, ID-20LA



Parts List	
Part #	Value
R1	100R
R2	4K7
R3	2K2
C1	10uF 25v electrolytic
C2	1000uF 10v electrolytic
C3	100nF
Q1	BC457 or similar
LED1	Read LED
LED2	Tag In Range LED
B1	2.7kHz – 3kHz 5v PKPK AC

9. Dimensions (mm)

ID-12LA			
	Nom.	Min.	Max.
A	12.0	11.6	12.4
B	8.0	7.6	8.4
C	15.0	14.6	15.4
D	25.3	24.9	25.9
E	20.3	19.8	20.9
F	16.3	15.8	16.9
G	26.4	26.1	27.1
P	2.0	1.8	2.2
H	6.0	5.8	6.6
J	9.9	9.40	10.5
W	0.66	0.62	0.67



10. Connection direct to a computer

Direct connection to a computer RS232 can be made by connecting Pin8 to a 1k series resistor and connecting the other end of the resistor to the computer RS232 input. The mode is called pseudo RS232. On a standard D9 socket, connect module Pin8 via the series 1k to pin2 of the D-type. Connect the ground to Pin5 on the D-type. Leave the TX pin3 open. See "Useful Information" below for free terminal download information.

Note that a +2.8v rail will result in the data outputs having a lower swing and may not be suitable for all computers.

10.1 Connection to a Processor UART

Direct connection to UART is made by connecting Pin9 to the UART Rx in pin

10.2 Connecting a Read LED

Sometimes the user may not want to drive a beeper but may still need to drive an LED. In this case a driver transistor may not be necessary because the Beeper Output Pin can supply 5mA continuously. Connect a 1k5 resistor to the Beeper Pin. This will limit the current. Connect the other end of the resistor to the LED anode and connect the cathode to ground.

11. Useful information

For general testing we suggest the user downloads a terminal program free from the internet. Here is one particularly good one to consider:

<http://braypp.googlepages.com/terminal> - Truly an excellent piece of software, the best terminal we have ever seen.

If you have any technical queries please contact your local distributor, they have all the technical resources to help you and support you. Where no local distributor exists, our technical helpline may be contacted by writing to help@ID-Innovations.com

ANEXO 2. Líneas de código del programa final cargado en el arduino.

```
BiciQudla $
#include <SoftwareSerial.h> //Creamos el puerto serial para la conexion del modulo Rfid
SoftwareSerial rfidReader(2,3); //Rx = 2 y Tx = 3
String card1="02006AAF64";
String card2="020069FAE9";
String card3="020069FB1C";
String card4="02006CE0FE";
int st1=0;
int st2=0;
int st3=0;
int st4=0;
String stringTwo;
char val = 0;
int d=0;

#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// if you don't want to use DNS (and reduce your sketch size)
// use the numeric IP instead of the name for the server:
//IPAddress server(74,125,232,128); // numeric IP for Google (no DNS)
char server[] = "www.biciquidla.com"; // name address for Google (using DNS)

// Set the static IP address to use if the DHCP fails to assign
IPAddress ip(192,168,0,177);

// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to (port 80 is default for HTTP):
EthernetClient client;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  rfidReader.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
}

void loop()
{
  // if there are incoming bytes available
  // from the server, read them and print them:
  if (client.available()) {
    char c = client.read();
    Serial.print(c);
  }
}
```



```

stringTwo="";
while (rfidReader.available () > 0)
{
    val = rfidReader.read();
    // d=val;
    stringTwo=stringTwo+val;
    //Serial.write(val);
    Serial.print(stringTwo);
    //Serial.print(d);

    if (stringTwo.substring(1) ==card1) {
        Serial.println(" TAG 1");
        if (st1==0){
            Serial.println("SALIDA");
            st1=1;
            sendtag1s();
        }else{
            st1=0;
            Serial.println("INGRESO");
            sendtag1i();
        }

        //delay(2000);
        Serial.flush();
        rfidReader.flush();
        stringTwo="";
        break;
    }

    if(stringTwo.substring(1)==card2) {
        Serial.println(" TAG 2");
        if (st2==0){
            Serial.println("SALIDA");
            st2=1;
            sendtag2s();
        }else{
            st2=0;
            Serial.println("INGRESO");
            sendtag2i();
        }
        delay(2000);
        Serial.flush();
        rfidReader.flush();
        stringTwo="";
        break;
    }
}

```

```

    if(stringTwo.substring(1)==card3) {
Serial.println(" TAG 3");
    if (st3==0){
        Serial.println("SALIDA");
        st3=1;
        sendtag3s();
    }else{
        st3=0;
        Serial.println("INGRESO");
        sendtag3i();
    }
    delay(2000);
    Serial.flush();
    rfidReader.flush();
    stringTwo="";
    break;
}

    if(stringTwo.substring(1)==card4) {
Serial.println(" TAG 4");
    if (st4==0){
        Serial.println("SALIDA");
        sendtag4s();
        st4=1;
    }else{
        st4=0;
        Serial.println("INGRESO");
        sendtag4i();
    }
    delay(2000);
    Serial.flush();
    rfidReader.flush();
    stringTwo="";
    break;
}

// if the server's disconnected, stop the client:
if (!client.connected()) {
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();
    //delay(1000);

    // do nothing forevermore:
}
//delay(1000);
}

```

```

void sendtagis(){
// start the Ethernet connection:
if (Ethernet.begin(mac) == 0) {
  Serial.println("Failed to configure Ethernet using DHCP");
  // no point in carrying on, so do nothing forevermore:
  // try to configure using IP address instead of DHCP:
  Ethernet.begin(mac, ip);
}
// give the Ethernet shield a second to initialize:
delay(1000);
Serial.println("connecting...");

// if you get a connection, report back via serial:
if (client.connect(server, 80)) {
  Serial.println("connected");
  // Make a HTTP request:
  client.println("GET /bici/php/salida.php?tag=02006AAF64 HTTP/1.1");
  client.println("Host: www.biciquidla.com");
  client.println("Connection: close");
  client.println();
}
else {
  // if you didn't get a connection to the server:
  Serial.println("connection failed");
}
delay(5000);
}
void sendtagli(){
// start the Ethernet connection:
if (Ethernet.begin(mac) == 0) {
  Serial.println("Failed to configure Ethernet using DHCP");
  // no point in carrying on, so do nothing forevermore:
  // try to configure using IP address instead of DHCP:
  Ethernet.begin(mac, ip);
}
// give the Ethernet shield a second to initialize:
delay(1000);
Serial.println("connecting...");

// if you get a connection, report back via serial:
if (client.connect(server, 80)) {
  Serial.println("connected");
  // Make a HTTP request:
  client.println("GET /bici/php/ingreso.php?tag=02006AAF64 HTTP/1.1");
  client.println("Host: www.biciquidla.com");
  client.println("Connection: close");
  client.println();
}
else {
  // if you didn't get a connection to the server:
  Serial.println("connection failed");
}
delay(5000);
}

```

```

void sendtag2s(){
// start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // no point in carrying on, so do nothing forevermore:
    // try to configure using IP address instead of DHCP:
    Ethernet.begin(mac, ip);
  }
// give the Ethernet shield a second to initialize:
delay(1000);
Serial.println("connecting...");

// if you get a connection, report back via serial:
if (client.connect(server, 80)) {
  Serial.println("connected");
  // Make a HTTP request:
  client.println("GET /bici/php/salida.php?tag=020069FAE9 HTTP/1.1");
  client.println("Host: www.biciquadla.com");
  client.println("Connection: close");
  client.println();
}
else {
  // kf you didn't get a connection to the server:
  Serial.println("connection failed");
}
delay(5000);
}

void sendtag2i(){
// start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // no point in carrying on, so do nothing forevermore:
    // try to configure using IP address instead of DHCP:
    Ethernet.begin(mac, ip);
  }
// give the Ethernet shield a second to initialize:
delay(1000);
Serial.println("connecting...");

// if you get a connection, report back via serial:
if (client.connect(server, 80)) {
  Serial.println("connected");
  // Make a HTTP request:
  client.println("GET /bici/php/ingreso.php?tag=020069FAE9 HTTP/1.1");
  client.println("Host: www.biciquadla.com");
  client.println("Connection: close");
  client.println();
}
else {
  // kf you didn't get a connection to the server:
  Serial.println("connection failed");
}
delay(5000);
}
}

```

```

void sendtag3s() {
// start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // no point in carrying on, so do nothing forevermore:
    // try to configure using IP address instead of DHCP:
    Ethernet.begin(mac, ip);
  }
// give the Ethernet shield a second to initialize:
delay(1000);
Serial.println("connecting...");

// if you get a connection, report back via serial:
if (client.connect(server, 80)) {
  Serial.println("connected");
  // Make a HTTP request:
  client.println("GET /bici/php/salida.php?tag=020069FB1C HTTP/1.1");
  client.println("Host: www.biciqudla.com");
  client.println("Connection: close");
  client.println();
}
else {
  // if you didn't get a connection to the server:
  Serial.println("connection failed");
}
delay(5000);
}

void sendtag3i() {
// start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // no point in carrying on, so do nothing forevermore:
    // try to configure using IP address instead of DHCP:
    Ethernet.begin(mac, ip);
  }
// give the Ethernet shield a second to initialize:
delay(1000);
Serial.println("connecting...");

// if you get a connection, report back via serial:
if (client.connect(server, 80)) {
  Serial.println("connected");
  // Make a HTTP request:
  client.println("GET /bici/php/ingreso.php?tag=020069FB1C HTTP/1.1");
  client.println("Host: www.biciqudla.com");
  client.println("Connection: close");
  client.println();
}
else {
  // if you didn't get a connection to the server:
  Serial.println("connection failed");
}
delay(5000);
}

```

```

void sendtag4s(){
// start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // no point in carrying on, so do nothing forevermore:
    // try to configure using IP address instead of DHCP:
    Ethernet.begin(mac, ip);
  }
  // give the Ethernet shield a second to initialize:
  delay(1000);
  Serial.println("connecting...");

  // if you get a connection, report back via serial:
  if (client.connect(server, 80)) {
    Serial.println("connected");
    // Make a HTTP request:
    client.println("GET /bici/php/salida.php?tag=02006CE0FE HTTP/1.1");
    client.println("Host: www.biciquidla.com");
    client.println("Connection: close");
    client.println();
  }
  else {
    // kf you didn't get a connection to the server:
    Serial.println("connection failed");
  }
  delay(5000);
}

void sendtag4i(){
// start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // no point in carrying on, so do nothing forevermore:
    // try to configure using IP address instead of DHCP:
    Ethernet.begin(mac, ip);
  }
  // give the Ethernet shield a second to initialize:
  delay(1000);
  Serial.println("connecting...");

  // if you get a connection, report back via serial:
  if (client.connect(server, 80)) {
    Serial.println("connected");
    // Make a HTTP request:
    client.println("GET /bici/php/ingreso.php?tag=02006CE0FE HTTP/1.1");
    client.println("Host: www.biciquidla.com");
    client.println("Connection: close");
    client.println();
  }
  else {
    // kf you didn't get a connection to the server:
    Serial.println("connection failed");
  }
  delay(5000);
}

```