



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:
INGENIERO EN ELECTRÓNICA DIGITAL Y
TELECOMUNICACIONES**

TEMA:

**PROTOTIPO DE UN SISTEMA AUTOMÁTICO DE FACTURACIÓN CON
TECNOLOGÍA RFID Y UNA APLICACIÓN ANDROID PARA AGILIZAR EL
PROCESO DE PAGO EN LAS CAJAS DE UN SUPERMERCADO.**

AUTOR:

CORELLA TITUAÑA MARCELA CRISTINA

TUTOR:

ING. FIDEL DAVID PARRA BALZA (Ph.D)

QUITO, ECUADOR

2018

UNIVERSIDAD ECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación “**PROTOTIPO DE UN SISTEMA AUTOMÁTICO DE FACTURACIÓN CON TECNOLOGÍA RFID Y UNA APLICACIÓN ANDROID PARA AGILIZAR EL PROCESO DE PAGO EN LAS CAJAS DE UN SUPERMERCADO**”, presentado por la Srta. Corella Tituaña Marcela Cristina, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. agosto del 2018

TUTOR

.....

Ing. Fidel David Parra Balza (Ph.D)

AGRADECIMIENTO

En primer lugar quiero agradecer a papito Dios por estar siempre junto a mí y nunca soltar mi mano, por brindarme salud y vida.

A la Universidad Tecnológica Israel por haberme acogido en sus instalaciones y brindarme la oportunidad de continuar mis estudios superiores en tan prestigiosa institución.

A los docentes de la Universidad Tecnológica Israel por su incansable labor diaria con sus estudiantes, por brindarnos sus conocimientos y amistad.

Agradezco también a mi asesor de tesis Ing. Fidel Parra (Ph.D) por haber aceptado dirigir mi tesis y ser un docente que inspira a sus estudiantes; así como también por haberme tenido la paciencia necesaria para guiarme en el desarrollo de mi tesis

A mi familia por todo su apoyo a lo largo de toda mi vida, de mis años de estudio, siendo mí soporte en las horas de flaqueza, y a la vez ejemplo de esfuerzo, amor y trabajo.

A mis amigos gracias por siempre han estado junto a mí apoyando y alentándome a cumplir mis metas y seguir hacia adelante.

Marcela Cristina Corella Tituaña

DEDICATORIA

Este trabajo está dedicado al ser más importante en mi vida mi hijo con quien he aprendido que la vida no tiene un manual para ser padres pero hacemos el mejor intento por fórjales un mejor futuro y verlos felices y que a su vez es mi inspiración y motivo de superación.

Ariel Nikolai

A mi madre por ser una fuente contaste de alegría, ejemplo de tenacidad y constancia, por nunca rendirse ante las adversidades y por su infinito amor y comprensión hacia su familia y hacia los demás.

Janet Patricia

A mi padre por ser un ejemplo de trabajo y responsabilidad, quien me inculco el valor del trabajo honrado, el respeto y el amor por la música.

Efrén Tarsicio

A mis hermanos que como en toda familia no estamos de acuerdo en todo y tenemos nuestras pequeñas diferencias pero siempre contamos el uno con el otro.

Ruddy

A mami Cris y papi Miguel mis abuelitos, mis tíos y tías, primos y primas que son muchos por eso no los nombro a cada uno de ustedes pero saben que los quiero y a toda mi familia por siempre estar junto a mí apoyándome, alentándome a seguir siempre superándome.

Y de manera especial a unos seres muy queridos para mí, que aunque hoy no se encuentren junto a mí siempre estarán en mi corazón y en mi memoria porque alguien muere cuando se le olvida, los amare por siempre.

Mami Juanita y Papi Toño

RESUMEN

El presente trabajo se lo realiza para buscar una mejor experiencia de compras de los clientes en los supermercados y de la misma manera apoyar a los supermercados con sistema de control de sus inventarios más exacto.

El avance constante de la tecnología obliga a estar permanentemente actualizados y en la actualidad existen un sin número de aplicaciones que permiten facilitar la vida de las personas.

La tecnología RFID es un sistema de identificación remota e inalámbrica mediante radio frecuencia (RF) que existe desde los años 40, y en las últimas décadas ha venido ganando terreno gracias a su creciente difusión en un varios tipos de aplicaciones como son control de acceso, seguridad, casetas de peaje, tarjetas electrónicas de transporte, controles de inventario, entre otros.

El sistema para automatización de facturas ha sido realizado pensando en recortar el tiempo que los clientes utilizan en los supermercados en facturar los productos adquiridos, el sistema está integrado por elementos RFID como etiquetas (Tags) y lector, un Smartphone con sistema operativo Android el cual contiene una aplicación para el usuario la misma que le permitirá acceder al sistema de facturación mediante el cual podrá generar su factura de manera personal sin la intervención de otra persona.

Palabras Claves:

RFID, Android, Tags, RF, Smartphone

ABSTRACT

This work is done to find a better shopping experience for customers in supermarkets and the same will support the supermarkets with more accurate inventory control system.

The constant advance of the technology forces to be permanently updated and at present there are a number of applications that make life easier for people.

RFID technology is a remote radio wireless (RF) identification system that has existed since the 1940s, and in the last decades it has been gaining ground thanks to its growing diffusion in several types of applications such as access control, security, toll booths, electronic transport cards, inventory controls, among others.

The system for invoice automation has been made thinking about cutting the time that customers use in supermarkets in billing the products purchased, the system is composed of RFID elements such as tags (tags) and reader, a smartphone with Android operating system which It contains an application for the user that will allow you to access the billing system through which you can generate your invoice personally without the intervention of another person.

Keywords:

RFID, Android, Tags, RF, Smartphone

INDICE GENERAL

AGRADECIMIENTO	III
DEDICATORIA	IV
RESUMEN	V
INDICE GENERAL	VII
INDICE DE FIGURAS	X
INDICE DE TABLAS	XII
INTRODUCCIÓN	1
1. Antecedentes de la situación objeto de estudio	1
2. Problema de Investigación: Planteamiento y Justificación	1
Planteamiento.....	1
Justificación.....	2
3. Objetivos del Trabajo de Titulación	3
Objetivo general:.....	3
Objetivos específicos:.....	3
CAPÍTULO 1	4
FUNDAMENTACIÓN TEÓRICA	4
1.1 HISTORIA DE LA TECNOLOGÍA RFID	4
1.2 EL ESTANDAR EPC	6
1.3 IDENTIFICACIÓN POR RADIO FRECUENCIA	7
1.3.1 Características de la Tecnología RFID	8
1.4 COMPONENTES BÁSICOS DE UN SISTEMA RFID	9
1.4.1 Etiqueta RFID o TAG (Transponder)	9
1.4.2 Tipos de Etiqueta	10
1.4.3 Lector RFID	12
1.4.4 Sistema de procesamiento de datos	13
1.5 VENTAJAS DEL SISTEMA RFID SOBRE EL CODIGO DE BARRAS	14
1.6 FACTURAS	14
1.6.1 Importancia de la facturación	15
1.7 OTRAS TECNOLOGÍAS INALAMBRICAS	16
1.7.1 WIFI	16
1.7.2 Bluetooth (BT)	17
1.8 ESPECTRO RADIOELÉCTRICO	18

1.9	MICROCONTROLADORES	19
1.9.1	Partes del microcontrolador.....	20
1.9.2	Microcontroladores Atmel AVR.....	21
1.9.3	Familias AVR	21
1.10	SISTEMAS OPERATIVOS	22
1.10.1	Tipos de sistemas operativos.....	23
1.10.2	Sistema Android.....	24
1.10.3	Características del S.O. Android	24
1.11	BASE DE DATOS	24
1.12	HERRAMIENTAS DE DESARROLLO	25
1.12.1	PHP	25
1.12.2	NetBeans IDE 8.2.....	26
1.12.3	Apache WebService	26
CAPÍTULO 2.		28
MARCO METODOLÓGICO		28
2.1	TIPO DE INVESTIGACIÓN.	28
2.2	TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS.	30
2.3	METODOLOGÍA SELECCIONADA	31
2.3.1	Fase I. Definición de las especificaciones	31
2.3.2	Fase II. Esquema general del hardware	31
2.3.3	Fase III. Organigrama general	32
2.3.4	Fase IV. Adaptación entre hardware y software	32
2.3.5	Fase V. Organigramas modulares y codificación del programa	32
2.3.6	Fase VI. Implementación del hardware	32
2.3.7	Fase VII. Depuración del software	32
2.3.8	Fase VIII. Integración del hardware con el software.....	33
2.3.9	Fase IX. Construcción del modelo definitivo y pruebas finales.....	33
2.4	Cuadro y cronograma de actividades y recursos.	33
CAPÍTULO 3.		35
PROPUESTA		35
3.1	INTRODUCCIÓN	35
3.2	HARDWARE	36
3.2.1	Diagrama de bloques	36
3.2.2	Módulo Lector RFID RC522	38
3.2.3	Características del módulo lector RFID RC522 RF.....	39
3.2.4	MIFARE1 S50	39

3.2.5	Características de MIFARE1 S50	40
3.2.6	Microcontrolador ATmega 328.....	40
3.2.7	Características del ATmega 328.....	41
3.2.8	Bluetooth HC-06.....	42
CAPÍTULO 4.	47
IMPLEMENTACIÓN	47
4.1	DIAGRAMA DE BLOQUES	47
4.2	DIAGRAMA DE FLUJO	48
4.3	DISEÑO DE LA PLACA	49
4.4	PROGRAMACIÓN DEL AVR	55
4.5	PROGRAMACIÓN DE LA BASE DE DATOS	58
4.5.1	Creación de la base de datos escalable	60
4.6	PROGRAMACIÓN DE LOS SISTEMAS.....	63
4.6.1	Programación en Android.....	63
4.6.2	Programación de la aplicación Distribuida	65
4.7	PRUEBAS DE FUNCIONAMIENTO.....	72

INDICE DE FIGURAS

Figura 1.1 Diagrama de un sistema RFID	4
Figura 1.2. Evolución de la tecnología	6
Figura 1.3 Formato de Código Electrónico de Producto	7
Figura 1.4 Esquema de funcionamiento de un sistema RFID.....	8
Figura 1.5 Sistema RFID	9
Figura 1.6 Componentes de una etiqueta RFID	10
Figura 1.7 Tipos de Etiquetas RFID	10
Figura 1.8 Etiquetas RFID activas.....	11
Figura 1.9 Etiquetas RFID pasivas	11
Figura 1.10 Lectores RFID.....	13
Figura 1.11 Factura estándar.....	15
Figura 1.12 Conexión WiFi.....	16
Figura 1.13 Red Bluetooth	17
Figura 1.14 Diagrama de bloques de un microcontrolador	20
Figura 1.15 Microcontrolador ATMEL.....	21
Figura 1.16 Interacción del sistema operativo.....	22
Figura 1.17 Sistemas operativos.....	23
Figura 1.18 Sistemas operativos móviles	23
Figura 1.19 Estructura del sistema.	25
Figura 3.1 Diagrama de bloques	37
Figura 3.2 Modulo de lectura y transmisión de datos	37
Figura 3.3 Módulo de recepción de datos	38
Figura 3.4 Lector RFID RC522	39
Figura 3.5 MIFARE1 S50.....	40
Figura 3.6 ATmega 328	41
Figura 3.7 Distribución de pines Atmega328.....	42
Figura 3.8 Bluetooth HC-06	43
Figura 3.9 Sistema de base de datos.....	44
Figura 3.10 Pantalla de hphMyAdmin.	46
Figura 4.1 Diagrama en bloques	47
Figura 4.2 Diagrama de flujo del proyecto	49
Figura 4.3 Circuito esquemático de interconexión	50
Figura 4.4 Diagrama impreso	51
Figura 4.5 Impresión en papel transfer	51
Figura 4.6 Transferencia del circuito	52
Figura 4.7 Transferencia a la placa	52
Figura 4.8 Bandeja con cloruro férrico	53
Figura 4.9 Proceso con el taladro.....	53

Figura 4.10 Soldadura de elementos	53
Figura 4.11 Top placa	54
Figura 4.12 Bottom placa	54
Figura 4.13 Vista superior e inferior del contenedor	55
Figura 4.14 Compilación en lenguaje C	56
Figura 4.15 Bloque de conexión	57
Figura 4.16 Pines de conexión módulo Bluetooth	57
Figura 4.17 Diagrama de interacción	58
Figura 4.18 Base de datos	59
Figura 4.19 Pagina phpMyAdmin	61
Figura 4.20 Página donde se ingresan los productos en la base de datos	62
Figura 4.21 Base de datos creada en phpMyAdmin	62
Figura 4.22 Programación en Android Studio	63
Figura 4.23 Pantallas de acceso al sistema Android	65
Figura 4.24 Página de descarga NetBeans	66
Figura 4.25 Página de descara JDK	67
Figura 4.26 Instalación JDK	68
Figura 4.27 Pantalla de instalación JDK	68
Figura 4.28 Pantalla de instalación NetBeans	69
Figura 4.29 Progreso de instalación NetBeans	69
Figura 4.30 Pantalla de inicio de NetBeans	70
Figura 4.31 App web para venta	72
Figura 4.32 Dispositivo RFID	73
Figura 4.33 Pantalla de facturación	76

INDICE DE TABLAS

Tabla 1.1	5
Tabla 1.2	12
Tabla 1.3	14
Tabla 1.4	18
Tabla 1.5	19
Tabla 2.1	34
Tabla 4.1	74
Tabla 4.2	75

INTRODUCCIÓN

1. Antecedentes de la situación objeto de estudio

Antes de la introducción de la tecnología informática en el país; los establecimientos comerciales como librerías, supermercados, tiendas de ropa, tiendas de calzado, almacenes de electrodomésticos, farmacias, entre otros; controlaban sus inventarios y facturación de forma manual. Con el avance de la tecnología las empresas vieron la necesidad de automatizar la captura de la información vinculada con los procesos de control de inventarios y facturación.

El control de facturación por compras y ventas en supermercados se realiza de forma manual y el archivo de las facturas también se las realiza de la misma forma, las cuales son usadas al final de cada período contable para la realización de los balances financieros. Al efectuarse esta manera la el proceso de facturación en supermercados se comete algunos errores en la facturación de los productos. Los errores que se presentan de manera frecuente están los relacionados con artículos facturados y no entregados al cliente, códigos de artículos incorrectos o artículos sin código.

Esto provoca la demora al momento de facturar, ya que de existir un artículo sin código el cajero tendrá que llamar a otro empleado para que pueda ir a la percha a buscar el código correspondiente al producto, esto genera además de largas filas en los supermercados para cancelar los productos adquiridos.

2. Problema de Investigación: Planteamiento y Justificación

Planteamiento

Actualmente el proceso de facturación tradicional presenta muchos inconvenientes tanto para la empresa como para el usuario, ya que realizar este proceso de forma manual, toma

una cantidad de tiempo importante que representa ganancias para los supermercados y una experiencia agradable para los usuarios.

Hoy en día se han encontrado diversos problemas al instante de realizar la facturación de compras hechas en los supermercados de forma tradicional, los problemas más comunes en los que se incurre son los que se detallan a continuación:

- Retardo en el proceso de facturación.
- Pérdida por robos
- Fácil violación y alteración de la información procesada en los sistemas actuales
- Alteración y pérdida de facturas
- Códigos erróneos en los artículos.

Justificación

La tecnología RFID es una tecnología de identificación remota y automática similar a la tecnología de lectura de código de barras pero con una interfaz más amigable, y con algunas ventajas como que no se necesita línea de vista directa, durabilidad de las etiquetas tags, la seguridad, además que es una tecnología en constante avance y que permite la escalabilidad.

El empleo de etiquetas de identificación (TAGs) incluye aplicaciones en varios campos favoreciendo el intercambio de datos. Dentro las aplicaciones más utilizadas se mencionan las siguientes: acceso a instalaciones, seguimiento a contenedores, acceso al uso de telefonía pública, monederos electrónicos, transacciones bancarias, control de inventarios, control de animales, control de peaje e identificación de personal entre otras.

El sistema a implementar pretende solucionar los problemas que en la actualidad se vienen dando en cuanto a una mejor optimización del tiempo de facturación en los supermercados, a controlar de mejor manera los hurtos que se producen dentro de estos lugares.

3. Objetivos del Trabajo de Titulación

Objetivo general:

Desarrollar un sistema automático de facturación con tecnología RFID y una aplicación Android para agilizar el proceso de pago en las cajas de un supermercado.

Objetivos específicos:

- Definir las etiquetas (TAG's) idóneas a utilizar de acuerdo a la frecuencia y forma.
- Construir la placa electrónica del dispositivo de lectura de códigos del prototipo.
- Diseñar una aplicación Android que facilite la comunicación con el usuario.
- Crear una base de datos escalable a todos los productos de un supermercado.
- Desarrollar la comunicación del dispositivo de lectura de códigos con la base de datos y con el teléfono mediante bluetooth para la generación de la factura.
- Validar el sistema mediante pruebas de funcionamiento.

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA

1.1 HISTORIA DE LA TECNOLOGÍA RFID

“Si tuviera que elegir una tecnología que realmente marcara la diferencia en los negocios, esa tendría que ser RFID” Glover Ferguson, Ingeniero jefe en Accenture.

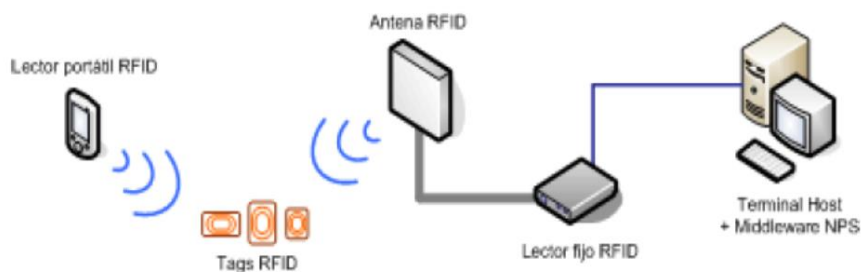


Figura 1.1 Diagrama de un sistema RFID

Fuente: www.nexpoint.com, 2006.

La tecnología RFID (Radio Frequency Identification) es un sistema de identificación remota e inalámbrica que existe desde 1939, el propósito primordial de esta tecnología es identificar mediante un lector, sin contacto y a la distancia el contenido de un objeto (Tag) mediante ondas de radio.

En los últimos años esta tecnología ha venido ganando terreno gracias a su creciente difusión en varias aplicaciones tales como control de acceso, casetas de peaje, tarjetas electrónicas de transporte, controles de inventario, apertura de vehículos, identificación de animales, tele-peaje, etc.

No fue sino gracias al descubrimiento de Faraday quien descubrió el concepto de inducción mutua, en el cual se fundamenta la alimentación de etiquetas pasivas que operan en un campo cercano. Mientras se desarrollaba la primera mitad del siglo veinte los avances tecnológicos desarrollados posibilitaron desarrollar etiquetas de largo alcance. (Ilyas Ahson, 2008)

La tecnología RFID no ha tenido un descubridor destacado, ha sido gracias a las numerosas aportaciones y colaboraciones de diferentes personas que se ha ido desarrollando, uno de los investigadores más relevantes Harry Stockman en un artículo afirma que “el trabajo considerable de investigación y de desarrollo tiene que ser realizado antes de que los problemas básicos restantes en la comunicación de la energía reflejada se solucionen, y antes de que el campo de aplicaciones útiles se explore”.

Usado por primera vez en la segunda guerra mundial (1939) por la armada británica con el fin de reconocer sus aviones para que no fueran objeto del fuego amigo. (ORSI, 2007)

Tabla 1.1

Evolución Histórica de la tecnología RFID

DECADAS	EVENTOS
1940 – 1950	<ul style="list-style-type: none"> Se define y emplea el radar. Esfuerzos de desarrollo en la Segunda Guerra Mundial. Se inventa RFID sobre 1948.
1950 – 1960	<ul style="list-style-type: none"> Primeras investigaciones sobre la tecnología RFID. Experimentos en laboratorios.
1960 – 1970	<ul style="list-style-type: none"> Desarrollo de la teoría RFID. Primeras pruebas de campo.
1970 – 1980	<ul style="list-style-type: none"> Explosión del desarrollo de RFID. Se aceleran las pruebas con RFID. Primeras implementaciones adaptadas con RFID.
1980 – 1990	<ul style="list-style-type: none"> Aplicaciones comerciales de RFID cobran importancia.
1990 – 2000	<ul style="list-style-type: none"> Surgen los estándares. RFID se despliega ampliamente.
2000 – 2010	<ul style="list-style-type: none"> Aparecen aplicaciones innovadoras, Combinación de RFID con servicios móviles personales. RFID subcutáneo para animales y humanos. RFID llega a formar parte de la vida diaria. En el año 2006 la Unión Europea realiza una consulta con objeto de conocer la opinión de la sociedad y los gobiernos europeos acerca de RFID.

Nota: Fuente: <https://www.google.com.ec/search?q=evolucion+de+la+tecnologia+RFID&source/>

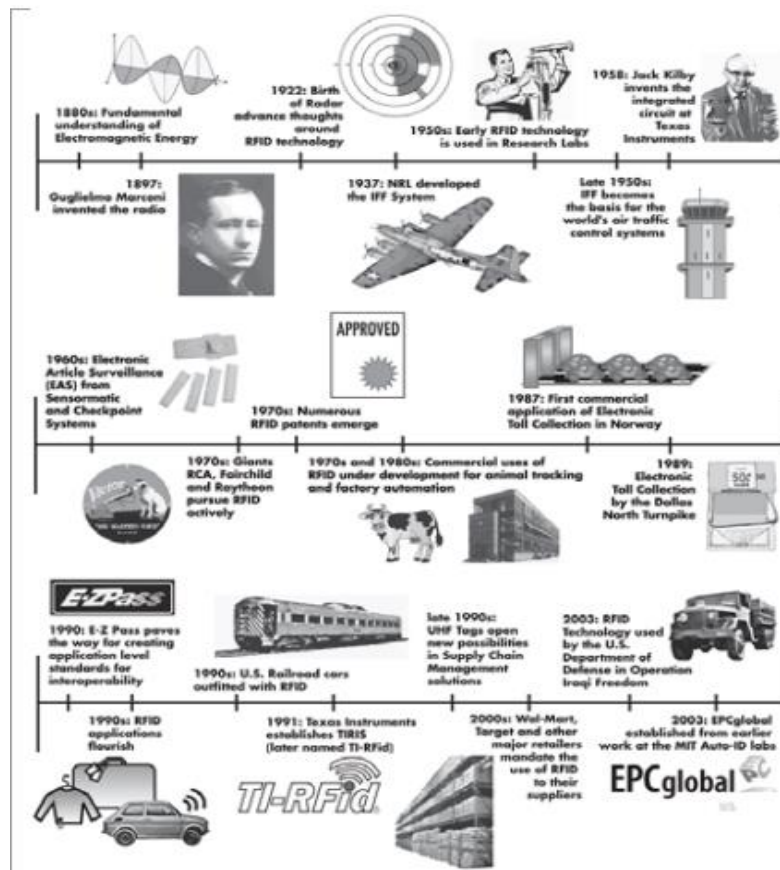


Figura 1.2. Evolución de la tecnología

Fuente: RFID Field Guide, Prentice Hall (2005)

1.2 EL ESTANDAR EPC

La falta de un estándar fue un freno para la expansión de la tecnología RFID, hasta el año 2003, el nuevo estándar (EPC) es independiente de la tecnología, define la estructura de los datos a codificar y las diferentes funcionalidades del sistema sin determinar la frecuencia de operación.

La tecnología RFID utiliza el estándar EPC global (2003) que en la actualidad ha sido aceptada a nivel mundial por los tres comités y patrocinadores que coordinan sus actividades estas son: ID CENTER, MIT, EPC CENTER

EPC Global Inc, es una organización independiente, y con estándares globales recomendados por la industria para el control de la adopción e implementación de la red

Global y la Tecnología EPC, esta organización mundial asigna códigos RFID; el objetivo de EPC Global es lograr una etiqueta básica a un coste bajo. (Telectrónica, 2006)

El código EPC por sus siglas en inglés Electronic Product Code, en español Código Electrónico de Producto es un estándar de codificación construido por GS1 y usado mundialmente para administrar información que tiene como una de funciones primordiales el permitir un control de inventario más preciso.

El código EPC es un número de serie único diseñado para identificar de forma exclusiva cualquier objeto a nivel mundial; el número EPC está incorporado a una base de datos, en la fig. 1.3 se muestra como está compuesto el código EPC.

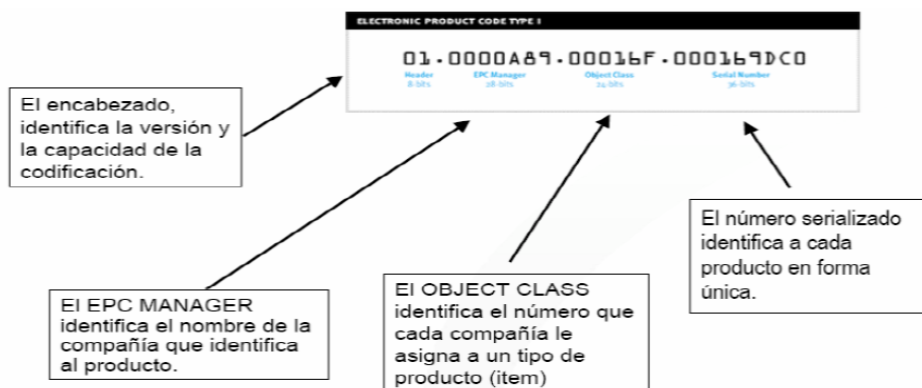


Figura 1.3 Formato de Código Electrónico de Producto

Fuente: Tomado del Observatorio Regional de la Sociedad de la Información (2007).

1.3 IDENTIFICACIÓN POR RADIO FRECUENCIA

La RFID es una tecnología de identificación y captura automática de información incluida dentro de etiquetas electrónicas (Tags), cuando dichas etiquetas entran en el rango de cobertura de un dispositivo de lectura y grabación RFID este emite una señal para que la etiqueta le transmita la información contenida en su memoria para transferirla hacia los sistemas de gestión, en la fig. 1,4 se puede muestra principio de funcionamiento de la tecnología RFID.

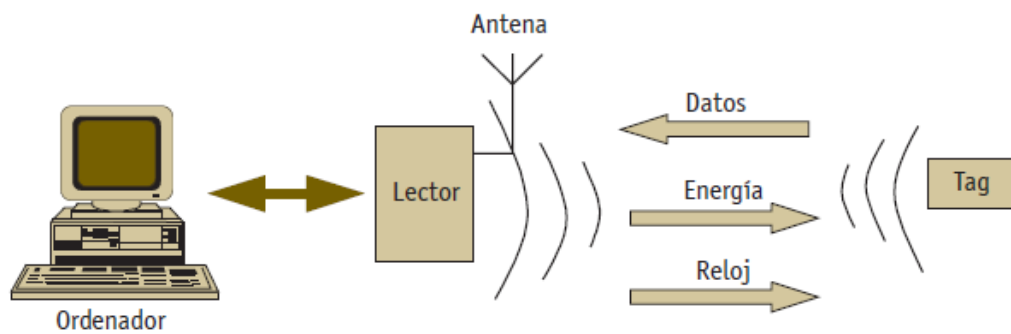


Figura 1.4 Esquema de funcionamiento de un sistema RFID.

Fuente: Informe de vigilancia médica Madrid, 2008.

La identificación por radio frecuencia es un sistema que nos brinda la facilidad de comunicarse sin cables entre dos o más objetos (comunicación inalámbrica), mediante la emisión y recepción de ondas de radio.

1.3.1 Características de la Tecnología RFID

El éxito de esta tecnología está en que el intercambio de datos se realiza sin la injerencia de ningún operador para iniciar el proceso de lectura de la RFID de modo que la identificación se realiza bajo las siguientes características:

- Sin contacto a distancia, no necesita de un operador que digite un código para activar su funcionamiento.
- Sin visualización directa, no requiere línea de vista directa entre el dispositivo lector y la etiqueta.
- Permite una identificación múltiple y simultánea, es capaz de identificar varias etiquetas al mismo tiempo.
- Identifica el movimiento y sin orientación determinada, la señal puede ser captada en cualquier dirección.
- Permite su escritura y lectura reiterada en tiempo real, se puede modificar el contenido de las etiquetas.

1.4 COMPONENTES BÁSICOS DE UN SISTEMA RFID

En la figura 1.5 se observa de qué está compuesto un sistema de identificación por Radiofrecuencia (RFID).

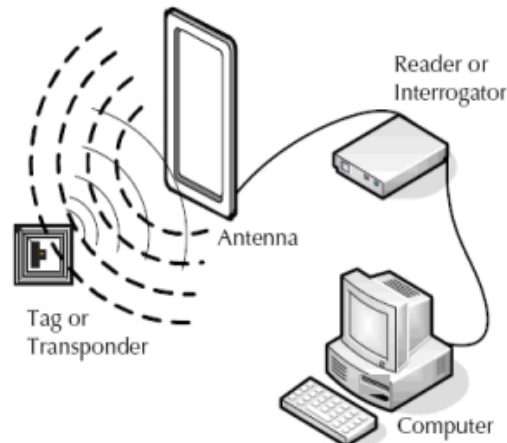


Figura 1.5 Sistema RFID

Fuente: <http://www.pymescentral.com/tarjetas-pvc/rfid>

El sistema RFID está conformado por 3 elementos básicos los cuales son:

- La Etiqueta RFID o TAG
- Lector RFID
- Sistema de procesamiento de datos

1.4.1 Etiqueta RFID o TAG (Transponder)

Consiste en un chip pequeño, conformado por una antena pequeña capaz de transmitir información hacia un dispositivo de lectura, se encargada de mantener los datos a procesarse, el TAG RFID está compuesto por tres elementos como se visualiza en la figura 1.6.

1.4.1.1 El chip.-Dispositivo donde electrónicamente se graba y lee la información.

1.4.1.2 La antena.-se encarga de recoger la señal de radio frecuencia (RF) activando el proceso de lectura y grabación en el chip, se clasifican de acuerdo a su intensidad de frecuencia en 3 tipos: baja frecuencia (LF), alta frecuencia (HF) y ultra alta frecuencia (UHF).

1.4.1.3 El encapsulado.-para el que se utilizan muy diferentes tipos adaptados al ambiente donde el Tag realizará su función; incluidos el encapsulado especial para ambientes hostiles donde son determinantes las temperaturas los líquidos y grasas, etc.



Figura 1.6 Componentes de una etiqueta RFID

Fuente: www.advantec.com

1.4.2 Tipos de Etiqueta

Existe una amplia gama de tipos de etiquetas RFID, su clasificación va a depender de las características que estas que se requiera que estas cumplan, en la fig. 1.7 se muestran los distintos tipos de TAGs RFID que existen en el mercado.



Figura 1.7 Tipos de etiquetas RFID

Fuente: <https://www.smartshippingmadeeasy.com>

1.4.2.1 **Tipo activo:** en estas etiquetas la alimentación es necesaria, por eso poseen su propia fuente independiente de energía, en la figura 1.8 se observa un ejemplo de etiqueta Activa.



Figura 1.8 Etiquetas RFID activas

Fuente: <http://www.tantorsystems.com/rfid.htm>

1.4.2.2 **Tipo pasivo:** son aquellas que no requieren batería, no poseen señal eléctrica, son alimentados únicamente por el campo magnético RFID, en la figura 1.9 se visualizan ejemplos de Tags pasivos.

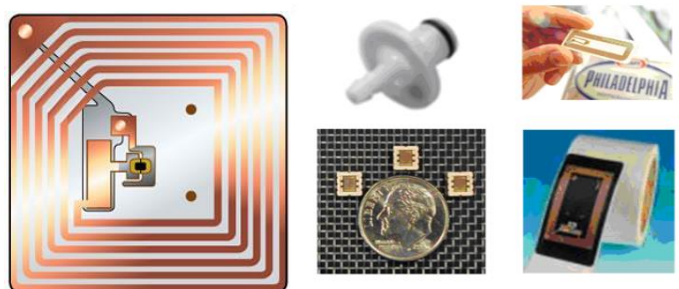


Figura 1.9 Etiquetas RFID pasivas

Fuente: <https://mundonfc.wordpress.com/2012/02/08/-rfid>

Tabla 1.2**Diferencia entre etiquetas pasivas y etiquetas activas.**

	Etiqueta Pasiva	Etiqueta Activa
Alcance	0.1 – 10m	10 – 100m
Alimentación	Campo RF	Batería o mixta
Sensores (Temperatura etc.)	Típicamente no	Típicamente si
Tiempo de vida	Ilimitado	Limitado por la batería
Coste	0,04 € ~ 0.8 €	0,08 € ~ 17 €
Dimensiones	Pequeño	Grande
Tipo de comunicación	Señalización pasiva	Señalización activa o pasiva
Ejemplos típicos	Etiqueta EPC, rastreo de animales, tarjetas inteligentes	Etiquetas para contenedores de transportes

Nota: Fuente: <http://itcmin2006.li2.uchile.cl/presentaciones/ITC>

1.4.3 Lector RFID

Es el encargado de transmitir energía suficiente para activar la etiqueta y que de esta manera los datos proporcionados puedan ser leídos, se encarga de percibir la señal que brinda la etiqueta.

El lector RFID cumple varias funciones entre ellas la principal tarea es activar las etiquetas TAGs y transferir la información proporcionada por ellas.

1.4.3.1 Tipos de lector/grabador e interrogador

Existen varios tipos de lector grabador interrogador RFID que se clasifican de la siguiente forma:

- ❖ **De mano:** es el más sencillo y económico.
- ❖ **Carretilla:** normalmente alimentado desde su propia batería.

- ❖ **Antena:** fundamentalmente utilizada para lecturas automáticas al atravesar arcos.
- ❖ **Fijo:** se utilizan en la lectura de etiquetas RFID ubicadas en la zona de entrada y salida de productos.



Figura 1.10 Lectores RFID

Fuente: <https://www.zebra.com/la/es/products/rfid/rfid-handhelds.html>

En la figura 1.11 se observa una variedad de tipos de lector grabador RFID.

1.4.4 Sistema de procesamiento de datos

También llamado RFID Middleware es el encargado de interpretar los datos obtenidos por el lector, maneja la comunicación entre los sistemas de aplicación y el hardware RFID.

El Middleware mueve los datos desde y hacia los puntos de transacción, por ejemplo, durante el proceso de lectura de una etiqueta, el middleware moverá los datos que están contenidos en una etiqueta desde el lector hacia un sistema de la empresa, al contrario, en un proceso de escritura de la etiqueta, el middleware moverá todos los datos desde el sistema de la empresa hacia el lector apropiado y en la última instancia hacia la etiqueta. El middleware posee 4 funciones fundamentales: la recolección de la información, el enrutamiento de información, administración de procesos y la administración de dispositivos. (Hunt et al., 2007)

1.5 VENTAJAS DEL SISTEMA RFID SOBRE EL CODIGO DE BARRAS

La tecnología RFID tiene bastante similitud con la tecnología antecesora en este caso el código de barras, son tecnologías de identificación automática por lo tanto es inevitable hacer una comparación entre ambas.

La tecnología RFID supera varias de las restricciones que presenta el código de barras, que es el sistema de identificación de objetos más utilizado hasta el momento en los locales comerciales.

Tabla 1.3

Diferencia entre el código de barras Vs. RFID

Códigos de barras	RFID
Requiere visión directa entre el lector y el código	No se requiere visión directa entre el lector y la etiqueta
Sensible a errores de impresión y a que su fijación sea correcta	Lectura independiente de dobleces o relieves en la adhesión de la etiqueta
Identifica un tipo de producto, pero no una unidad de dicho producto	Se identifica cada producto unitario, no sólo su modelo
Muy sensibles al paso del tiempo.	Resistencia a condiciones adversas de suciedad, temperatura, humedad, etc.,
Toda la información es visual por lo que cualquier persona puede acceder a ella	Mayor seguridad gracias a la protección de los datos ,mediante contraseñas o transmisión de datos cifrada
La información de las etiquetas es inalterable	Posibilidad de modificar la información contenida en las etiquetas
Solo se permite la lectura simultanea de un etiqueta	Identificación simultanea de varias etiquetas

Nota: Fuente: <https://www.youtube.com/watch?v=POCVUL98Aok>

1.6 FACTURAS

Una factura es un documento mercantil que revela toda la información de una operación de compraventa de un producto, bien o prestación de algún servicio, como se

puede observar en la figura 1.11. La emisión de una factura es de carácter obligatorio ya que es una prueba física de la realización de una operación económica.

El formulario de factura estándar incluye los siguientes campos:

- Tu Logo**: Espacio para el logo del proveedor.
- FACTURA N° 0001-**: Número de factura.
- DIA, MES, AÑO**: Campos para la fecha de emisión.
- C.U.I.T.**: Código Único de Identificación Tributaria.
- Ing. Brutos** y **Inicio de Actividades**: Información fiscal y operativa.
- I.V.A. Responsable Monotributo**: Sección para datos de IVA.
- Señor(es):** y **Domicilio:** Datos del destinatario.
- IVA**: Sección para clasificar el tipo de IVA (Responsable Inscripto, Responsable no Inscripto, IVA Responsable Monotributo, No Responsable, Exento, Consumidor Final).
- Condiciones de Venta:** Campos para Contado, Cta. Corriente, y Remito N°.
- Tabla de Detalles:** Tabla con columnas: Cant, Detalle, P. Unit, TOTAL.
- TOTAL \$:** Campo para el monto total de la factura.
- Nota al pie:** Información de contacto y datos técnicos de la imprenta.

Figura 1.11 Factura estándar

Fuente: Facturación Tradicional y Facturación Electrónica, 2004.

1.6.1 Importancia de la facturación

Desde la antigüedad, los seres humanos se han encargado de realizar un sin número de actividades que tienen como principal actividad el canje de productos con la objetivo de obtener una mejor calidad de vida y bienestar económico.

La facturación tiene una gran utilidad no solo a la hora de especificar las características y precio de los productos o bienes adquiridos, ya que al solicitar la factura el vendedor garantiza la que el bien o producto se encuentre sin desperfectos o falla de algún material y por este medio se ofrece garantía del producto; también a la hora del pago de impuestos puesto que en las facturas se refleja el pago del impuesto al valor agregado.

1.6.1.1 Facturación electrónica.- Una factura electrónica es exactamente una factura en sí mismo, en el Ecuador desde el año 2013 la emisión de facturas electrónicas es obligatoria de acuerdo a lo publicado en el registro oficial N°956.

1.7 OTRAS TECNOLOGÍAS INALÁMBRICAS

En la actualidad, debido a la necesidad del ser humano de comunicarse unos con otros, y también al desarrollo y avance de nuevas tecnologías inalámbricas que se expanden en un rango muy amplio de aplicaciones se ve en la necesidad de conocer más acerca de estas tecnologías.

El principio fundamental de las tecnologías inalámbricas es la comunicación que se lleva a cabo sin cables, mediante ondas de radio frecuencia.

A continuación se exploraran las tecnologías que actualmente se están usando en el mercado.

1.7.1 WIFI

Es una tecnología que permite la conexión inalámbrica de varios dispositivos electrónicos. Hoy en día los dispositivos habilitados con Wifi han proliferado de manera sustancial (tales como celulares, computadoras, televisores, videoconsolas, reproductores de música, etc.) estos dispositivos pueden interconectarse entre sí o a través de un punto de ingreso a una red inalámbrica, en la figura 1.12 se muestra las formas de conexión de los dispositivos Wifi.



Figura 1.12 Conexión WiFi

Fuente: <http://cruzlanda.blogspot.com/2010/04/tecnologias-inalambricas-la-tecnologia.html>

1.7.2 Bluetooth (BT)

Se trata de una tecnología inalámbrica de corto alcance que fue implantada para reemplazar cables conectando dispositivos móviles. (Játiva, 2016)



Figura 1.13 Red Bluetooth

Fuente: (Abinayaa & Jayan, 2104)

En la figura 1.13 se muestra la forma de conexión de una red bluetooth.

Esta tecnología nace ante la necesidad de comunicarse de forma ágil, fácil y segura; y con un costo bajo y un consumo de energía reducido. Fue la empresa Ericsson en 1994 quien impulsó esta tecnología, junto con otras grandes compañías del sector tecnológico como Nokia, Intel, IBM, Toshiba, Motorola quienes apoyaron el estándar.

Está dirigida a aplicaciones de voz y datos, Bluetooth opera a una distancia mínima de 1 metro hasta 100 metros dependiendo de la clase del dispositivo Bluetooth. Funciona en la banda ISM de los 2,4 GHz. (Bluetooth SIG, Inc., 2015)

Tabla 1.4
Características de bluetooth

Clave para Bluetooth		
Características	Especificaciones	Notas
Frecuencia portadora	2400-2483.5 (banda ISM)	Excepciones Francia:2446.5-2483.5 Japón: 2471-2497 España: 2445-2475
Tipo de modulación	FSK binario con filtrado Gaussiano a una velocidad de IMb/s	Índice de modulación nominal es de 0.32 o puede estar en el rango de 0.28 a 0.35. Pico de desviación permitido es de 175 Khz.
Escaneo	Velocidad d escaneo es de 1600 saltos por segundo para operación normal cuatro secuencias espaciales de escaneo estan reservadas para el voltaje de conexión	Periodicidad de la secuencia es de 23 horas y 18 minutos
Potencia de transmision en mW	Clase1: 1-100 Clase 2: 0.25-2.5 Clase 3: 1	El control de potencia es requerido para la clase 1 pero es opcional para las otras clases
Rango de operación en metros	0.1 a 10 metros	Hasta 100 metrlos para clase 1
Maxima capacidad de procesamiento de datos	Para un canal asincronico puede ser: Un enlace asincronico puede ser: Un enlace asimetrico hasta 721Kb/s en una dirección y 56.6 Kb/s en la otra dirección. Para un enlace simétrico con 432.6 Kb/s en ambas direcciones	La capacidad de procesamiento de datos es un poco menor a 1 Mb/s debido al encabezamiento de el protocolo.

Nota: Fuente:https://ldc.usb.ve/~poc/RedesII/Grupos/G1/como_funciona.html

1.8 ESPECTRO RADIOELÉCTRICO

“Es el medio a través del cual se transmiten las frecuencias de las ondas de radio electromagnéticas que posibilitan los servicios de telecomunicación como son la radio, televisión, internet, la televisión digital terrestre, entre otros, el cual es administrado por el gobierno de cada país.” (Gallardo, 2015, p. 93)

El organismo encargado de su gestión es la Unión Internacional de Telecomunicaciones (UIT) que divide el espectro radioeléctrico en bandas de frecuencia, como se visualiza en la tabla 1.5.

Tabla 1.5

Designación de bandas del espectro radioeléctrico según la UIT

Nombre	Abreviatura	Banda UIT	Frecuencias	Longitud de onda
			< 3 Hz	> 100.000 km
Extra baja frecuencia	ELF	1	3-30 Hz	100.000–10.000 km
Súper baja frecuencia	SLF	2	30-300 Hz	10.000–1000 km
Ultra baja frecuencia	ULF	3	300–3000 Hz	1000–100 km
Muy baja frecuencia	VLf	4	3–30 KHz	100–10 km
Baja frecuencia	LF	5	30–300 KHz	10–1 km
Media frecuencia	MF	6	300–3000 KHz	1 km – 100 m
Alta frecuencia	HF	7	3–30 Mhz	100–10 m
Muy alta frecuencia	VHF	8	30–300 Mhz	10–1 m
Ultra alta frecuencia	UHF	9	300–3000 Mhz	1 m – 100 mm
Súper alta frecuencia	SHF	10	3-30 Ghz	100-10 mm
Extra alta frecuencia	EHF	11	30-300 Ghz	10–1 mm
			> 300 Ghz	< 1 mm

Nota: Fuente: Elementos de sistemas de telecomunicaciones, 2015

1.9 MICROCONTROLADORES

Un microcontrolador es un circuito integrado programable que en su interior contiene básicamente un microprocesador, unidades de memoria (RAM y ROM), puertos de entrada-salida y periféricos, todo esto dentro de un pequeño chip, en el cual podemos programar con total flexibilidad y relativa facilidad.

“Un microcontrolador es un computador completo, aunque de limitadas prestaciones, que está contenido en el chip de un circuito integrado y se designa a gobernar una sola tarea” (Angulo & Angulo, 2003, p. 1)

En un solo chip esta contenido la CPU, la memoria RAM, ROM, los puertos de entradas y salida y otros periféricos como se muestra en la figura 1.14.

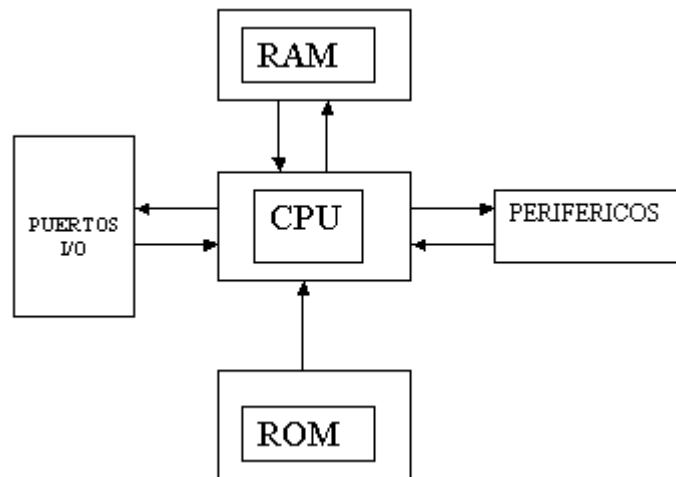


Figura 1.14 Diagrama de bloques de un microcontrolador

Fuente: <http://www.electronicaestudio.com/microcontrolador.htm>

1.9.1 Partes del microcontrolador

1.9.1.1 **CPU.-** Es la unidad central de procesamiento que es la parte más importante del microcontrolador ya que es el que realiza todos los procesos dentro del microcontrolador.

1.9.1.2 **La unidad de memoria.-** Es aquella que almacena los datos del microprocesador, existen varios tipos de memoria en el microcontrolador.

- **ROM:** (Read Only Memory) en español memoria solo de lectura., utilizada en el almacenamiento de datos de forma permanente el programa que se ejecuta.
- **RAM:** (Random Access Memory) en español significa memoria de acceso aleatorio, esta memoria no guarda datos de manera permanente ya que al apagar la fuente de alimentación se pierde el contenido que este almacenada en ella.

1.9.1.3 **Puertos de entrada y salida.-** Estos puertos son los que permiten la comunicación del procesador con el exterior a través de interfaces o con otros dispositivos.

1.9.2 Microcontroladores Atmel AVR

Los AVR es una familia de microcontroladores RISC que pertenece al fabricante estadounidense ATMEL, la tecnología de los microcontroladores AVR es CMOS de 8 bits con bajo consumo de voltaje, esto implica que poseen un juego reducido de instrucciones.



Figura 1.15 Microcontrolador ATMEL

Fuente: Taller de microcontroladores, 2012.

La arquitectura AVR fue diseñada por dos estudiantes del Norwegian Institute of Technology, y después fue perfeccionada y desarrollada en Atmel Norway, una empresa adjunta de Atmel, fundada por los dos arquitectos diseñadores del chip.

Una característica que describe a estos microcontroladores es la memoria flash y EEPROM que han incorporado, el diseño de AVR se diferencia de los demás microcontroladores de 8 bits por poseer una mayor cantidad de registros (32), al tener un conjunto de registros disminuye la necesidad de recurrir a la memoria, por lo cual mejora la velocidad de procesamiento y disminuye la necesidad de almacenamiento de datos.

1.9.3 Familias AVR

Existen varias categorías en el mercado, las cuales se puede clasificar de la siguiente manera:

- TINY AVR: pequeños microcontroladores de propósito general con una memoria flash hasta 2 Kbytes y 128 bytes de memoria SRAM y EEPROM.

- AVR: son microcontroladores de propósito general con 8 Kbytes de memoria flash y 512 bytes de memoria SRAM y EEPROM.
- Mega AVR: memoria flash hasta 256 Kbytes, 4 Kbytes de memoria EEPROM y SRAM.
- AVR XMEGA: memoria de programa de 384 Kbytes, velocidad de hasta 33 MHz.

1.10 SISTEMAS OPERATIVOS

Un sistema operativo es un programa o conjunto de programas (software) encargado de gestionar todos los recursos del sistema informático, es el que facilita la comunicación entre el usuario y el ordenador (todos los programas del ordenador y los dispositivos de hardware) de forma confortable y eficiente, en la figura 1.16 se muestra la interacción del sistema operativo con su entorno.



Figura 1.16 Interacción del sistema operativo

Fuente: <http://www.areatecnologia.com/sistemas-operativos>

Un sistema operativo debe cumplir algunos objetivos. Martínez (2001) afirma que: “El objetivo primario de un sistema operativo es optimizar todos los recursos del sistema para que este pueda soportar los requerimientos”. (p.3)

Un sistema operativo cumple varias funciones pero las más destacadas son administrar los recursos del ordenador, coordinar el hardware y la organización de archivos. “Una de las principales funciones del S.O. es ocultar toda esa complejidad y brindar al programador un conjunto más conveniente de instrucciones para trabajar” (Martínez, 2001, p.4).

1.10.1 Tipos de sistemas operativos

Existen distintos tipos de sistemas operativos, a continuación se observaremos los sistemas operativos dependiendo del dispositivo que se esté usando.

1.10.1.1 *Sistemas operativos para PC.* - o computadores de escritorio existen en gran cantidad, los más utilizados son: Windows, Mac, Linux, en la figura 1.18 se muestran los logotipos de los sistemas operativos más usados.



Figura 1.17 Sistemas operativos

Fuente: Fundamentos de Sistemas Operativos, 2015

1.10.1.2 *Sistemas Operativos Móviles.*- son los sistemas operativos utilizados en dispositivos móviles (Smartphones) o tablets, son muchos como por ejemplo: Android, IOS, Symbian, Blackberry OS, Windows Phone, entre otros, siendo los más utilizados Android e IOS,; en la figura 1.19 se muestran los logotipos de los sistemas operativos para dispositivos móviles.



Figura 1.18 Sistemas operativos móviles

Fuente: <http://dexpierta.com/sistemas-operativos-moviles>

1.10.2 Sistema Android

Es un sistema operativo diseñado por la compañía Google, que fue desarrollado principalmente para dispositivos móviles con pantalla táctil, este sistema basa la mayor parte de su operación en el software libre (un núcleo de Linux).

Android ha tenido gran acogida ya que Google a diferencia de Apple no fabrica sus propios dispositivos, sino que ha otorgado licencias para el uso de su sistema operativo prácticamente a todos los fabricantes de hardware, por lo que la gran mayoría de los modelos de teléfonos inteligentes y tabletas corren sobre Android. (Wolf, Ruiz, Bergero, meza, 2015)

1.10.3 Características del S.O. Android

- Sistema de código abierto, basado en Linux
- Sistema multitarea real de aplicaciones
- Ofrece diferente formas de mensajería
- Adaptable a diferentes resoluciones de pantalla
- Soporta java y varios formatos multimedia.
- Soporte de HTML, HTML5, Adobe Flash Player, entre otros.
- Gran catálogo de aplicaciones gratuitas y de pago para descargar e instalar.

1.11 BASE DE DATOS

Camps et al. (2005) afirma que “estos Conjuntos de ficheros interrelacionados, con estructuras complejas y compartidos por varios procesos de forma simultanea (unos on-line y otros por lotes), recibieron al principio del nombre de *Data Banks*, y después, a inicios de los años setenta, el de *Data Bases*” (p.7).

Una base de datos es un sistema de información organizada de modo que sea de fácil acceso, que gestione y actualice datos, una base de datos es un repositorio donde se guarda información que se pueda almacenar y recuperar.

En la Figura 1.20 se muestra la estructura de un sistema de Base de Datos.

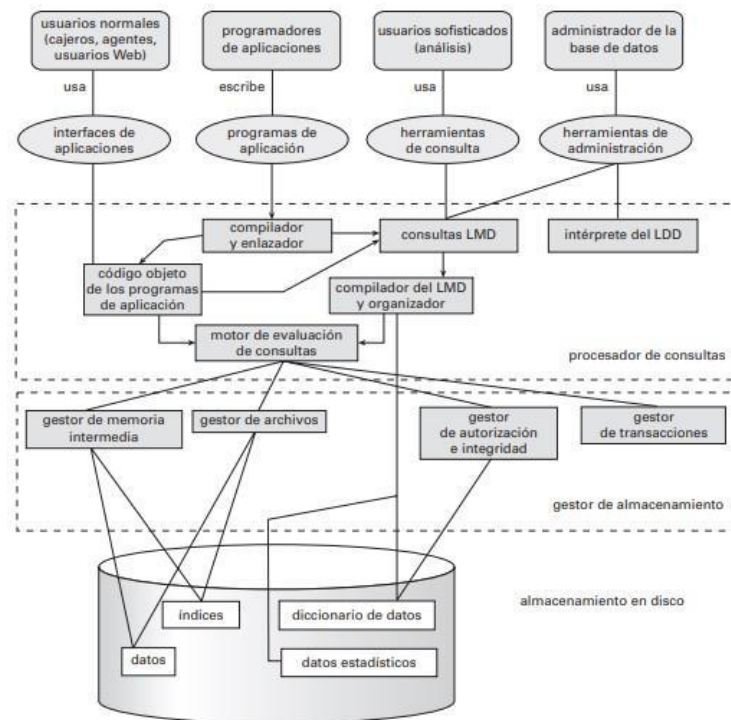


Figura 1.19 Estructura del sistema.

Fuente: tomado de Fundamentos de base de datos, 2005.

1.12 HERRAMIENTAS DE DESARROLLO

1.12.1 PHP

Fue creado originalmente por Rasmus Lerdorf en 1994, su implementación principal es producida por The PHP Group. (Heurtel, 2016).

El lenguaje PHP (históricamente Personal Home Page, oficialmente acrónimo recursivo de Hypertext Preprocessor) es un lenguaje de programación de código abierto que se caracteriza por su versatilidad, robustez, potencia y modularidad, muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Es recomendable utilizar PHP es su extrema simplicidad para el principiante, ya que a su vez ofrece muchas características avanzadas para los programadores profesionales tales como:

- Es un lenguaje libre.
- Está disponible para varios sistemas operativos (Windows, GNU/Linux, UNIX, entre otros).
- Tiene una amplia documentación en su sitio web.
- Es considerado un lenguaje de fácil aprendizaje
- Su capacidad de conexión con la mayoría de los motores de base de datos que se utiliza en la actualidad
- Se considera un lenguaje altamente eficiente.

1.12.2 NetBeans IDE 8.2

NetBeans es un entorno de desarrollo integrado libre y de código abierto, para aplicaciones programadas a través de Java, es un producto de uso libre y gratuito sin restricciones.

NetBeans es una herramienta diseñada para escribir, compilar, depurar y ejecutar programas, aunque este escrito en Java puede servir también para cualquier otro lenguaje de programación, la programación a través de NetBeans se realiza mediante componentes de software modulares. Llamados módulos.

Con NetBeans IDE es posible crear potentes aplicaciones para el escritorio, también para la web, así como para dispositivos portátiles, como móviles o Pocket PC.

1.12.3 Apache WebService

Es un servidor web de distribución libre y de código abierto, es muy popular, con una inserción actual del 70% del total de servidores web del mundo; su arquitectura es modular.

Esta aplicación puede ejecutarse en múltiples sistemas operativos tales como: Windows, Mac OS X, y los sistemas basados en Unix.

El principal oponente de Apache es Microsoft Internet Information Services (IIS), Apache es desarrollado por una comunidad abierta de desarrolladores auspiciados por la Apache Software Foundation.

1.12.4 MY SQL

Es la base de datos de código abierto más popular en el mundo, gracias a su versatilidad, confiabilidad, rendimiento, facilidad de uso y escalabilidad.

A su vez es un sistema de gestión de base de datos relacional, multi-hilo y multi-usuario; es una base de datos integrada y segura para transacciones, compatible con múltiples plataformas informáticas y ofrece varias aplicaciones.

CAPÍTULO 2.

MARCO METODOLÓGICO

En este capítulo se mencionan los aspectos metodológicos y prácticos que se utilizaron para desarrollar un Prototipo de un sistema automático de facturación con tecnología rfid y una aplicación android para agilizar el proceso de pago en las cajas de un supermercado. Éste describió el tipo de investigación que se realiza, las técnicas e instrumentos de recolección de datos, la metodología seleccionada (fases del proyecto), cuadro y cronograma de recursos, herramientas y materiales utilizados.

2.1 TIPO DE INVESTIGACIÓN.

De acuerdo con los criterios expuestos por Hurtado, J (2006), la presente investigación fue proyectiva por cuanto intenta “planear soluciones a una situación determinada a partir de un proceso previo de indagación. Implica explorar, describir, explicar y proponer alternativas de cambio, mas no necesariamente ejecutar la propuesta”. (p.117).

Por otra parte. Palella y Martins (2006) definen dicha investigación como aquella que “conlleva a diseño o creaciones dirigidas a cubrir una necesidad y basadas en conocimientos anteriores. De allí que el término proyectivo refiere a un proyecto en cuanto aproximación o modelo teórico. El investigador puede llegar a este mediante vías diferentes las cuales involucran procesos, enfoques, métodos y técnicas propias”. (p.103).

Por lo expuesto se puede decir que la investigación es de tipo Proyectiva puesto que, a través de estudios previos, propone el desarrollo de un sistema automático de facturación con tecnología rfid y una aplicación android para agilizar el proceso de pago en las cajas de un supermercado, brindando un aporte de gran importancia a la vanguardia tecnológica. El cual involucra procesos como enfoque, métodos y técnicas propias.

Por otro lado, según Chávez (1994). Las investigaciones **DESCRIPTIVAS** “son todas aquellas que se orientan en recolectar informaciones relacionadas con el estado real de las personas, objetos, situaciones o fenómenos, tal como se presentaron en el momento de su relación, describe lo que se mide, sin realizar inferencias ni verificar hipótesis”.(p.176)

A su vez Tamayo y Tamayo, señala que “este tipo de estudio busca únicamente describir situaciones o acontecimientos; básicamente no está interesado en comprobar explicaciones, ni en probar determinadas hipótesis. Con mucha frecuencia se usan las descripciones por encuestas (estudios por encuestas), aunque éstas también pueden servir para probar hipótesis específicas y poner a prueba explicaciones.”.

Basándose en lo antes descrito la investigación se considera descriptiva ya que pretende observar, dar detalles y recolectar informaciones relacionadas con el estado real del comportamiento del prototipo de un sistema automático de facturación con tecnología rfid y una aplicación android para agilizar el proceso de pago en las cajas de un supermercado.

Según Palella y Martins (2006), el diseño de **Campo** “consiste en la recolección de datos directamente de la realidad donde ocurre los hechos sin manipular o controlar variables. Estudia los fenómenos sociales en su ambiente natural”. (p. 97). También afirman que el diseño **no experimental**, es aquel que se realiza sin manipular de forma deliberada ninguna variable. El investigador no sustituye intencionalmente las variables independientes. Se observan los hechos tales como ocurren en su contexto real en un lapso determinado o no, para luego analizarlo.

Por su parte Hernández y otros (2006) describen que, “en el diseño no experimental, las variables independientes ocurren y no es posible manipularlas, no se tiene control directo sobre dicha variables ni si pueden influir sobre ellas, porque ya sucedieron, al igual que sus efectos”.(p.205). Por su parte, Heinemann, K (2003), menciona que “la investigación donde se recopilan los datos una sola vez en un momento determinado, posee un diseño de tipo Transversal.”. (p. 176).

Bavaresco (2001), plantea que “La investigación también puede ser de campo;

cuando se realiza en el propio sitio donde se encuentra el objeto de estudio. Ello permite el conocimiento más a fondo del problema por parte del investigador y puede manejar los datos con más seguridad. Así podrá soportarse en diseños exploratorios, descriptivos, experimentales y predictivos”. (p.28)

Considerando las definiciones anteriores y basándose en las características del presente trabajo, resulta aplicable el diseño No Experimental, debido a que no se pretende modificar o manipular la realidad actual de las variables, es decir, se observan las situaciones ya existentes, sin ser provocadas intencionalmente por el investigador. El objeto de estudio es Transversal, puesto a que se recopilan los datos una sola vez en un momento determinado.

Por último según la forma de obtener los datos es de Campo, ya que se realiza en el propio sitio donde se encuentra. Cuando se conoce el tipo y diseño de investigación, se tiene de manera precisa y clara a donde pretende llegar ésta y con qué finalidad se realiza este importante trabajo.

2.2 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS.

Se refiere al uso de diversas técnicas e instrumentos que son utilizadas en buscar información con la finalidad de manifestar el problema planteado. Toda investigación se sustenta en los datos y la información recolectada durante la elaboración de la misma. Esta documentación permite al investigador el dominio de cada uno de los procesos relacionados con el proyecto, lo cual es vital para la elaboración objetiva y respaldo de propuestas, conclusiones, recomendaciones, entre otras.

Ahora bien, para la realización de la investigación se utilizó como técnica e instrumento para recolectar datos, la observación de manuales técnicos, herramientas electrónicas, software, entre otras. En esta investigación se utilizó la técnica de la observación documental, de presentación resumida, descrita por Balestrini (1998, p. 154), como “una lectura general de los textos que contienen las fuentes de información que son de mucho interés, extrayendo los datos identificados de utilidad para la investigación”.

La misma fue ejecutada en esta investigación a partir de las búsquedas especializadas de documentos, patentes, informes o publicaciones relativas a atendencias tecnológicas asociadas con las tecnologías inalámbricas. En cuanto al instrumento aplicado, Arias (2006, p. 69) lo define como “cualquier recurso, dispositivo o formato (en papel o digital) que se utiliza para obtener, registrar o almacenar información”.

2.3 METODOLOGÍA SELECCIONADA

Las actividades planificadas para el desarrollo de la investigación están basadas en la metodología planteada por Angulo, J. (1986), debido a la concordancia de los objetivos con las fases propuestas por el autor. De este modo, la metodología utilizada es **directa** al acudir a las nueve (9) fases que el mismo propone.

En la primera fase se definirán las especificaciones; en la segunda se establecerá el esquema general del hardware; en la tercera, se conformará el organigrama general; en la cuarta, se realizará la adaptación entre el hardware y el software; en la quinta, se constituirán los ordinogramas modulares y se realizará la codificación del programa; en la sexta, se establecerá la implementación del hardware; en la séptima, se ejecutará la depuración del software; en octavo lugar se llevará a cabo la implementación del hardware con el software, y en noveno lugar se construirá el modelo definitivo y se realizarán las pruebas finales, quedando constituidas según se expone a continuación:

2.3.1 Fase I. Definición de las especificaciones

En esta primera fase se define con la mayor precisión el funcionamiento del sistema a desarrollar, también se deben establecer los estímulos de entrada y salida sin explicar las razones.

2.3.2 Fase II. Esquema general del hardware

En esta fase, se desarrolla en forma general los bloques funcionales que los componen y las interconexiones entre sí de manera lógica. Se asume como bloques funcionales, cada

una de las partes elementales del sistema encargada de hacer un único trabajo en particular.

2.3.3 Fase III. Organigrama general

Siguiendo la secuencia, se establece un diagrama de flujo que se estima indicará el funcionamiento en forma general del sistema. Se denomina general porque su objetivo es de servir como base para el desarrollo del software, el cual está sujeto a modificaciones. Este se realiza en forma general pero que ilustre el funcionamiento del sistema

2.3.4 Fase IV. Adaptación entre hardware y software

Luego de formar la estructura del hardware y el software se acuerdan los mecanismos para la comunicación entre ambos, garantizado que la información entra y sale de forma correcta.

2.3.5 Fase V. Organigramas modulares y codificación del programa

En esta fase cada uno de los siguientes bloques del diagrama de flujo se codifican individualmente, asegurándose que cada parte realice el trabajo en forma eficiente y segura, esto se logra codificando en forma independiente con el software seleccionado para obtener un óptimo resultado.

2.3.6 Fase VI. Implementación del hardware

En esta etapa se elabora el circuito, tomando en cuenta las características de todos los materiales a utilizar así como también las hojas de especificaciones de los fabricantes de los componentes.

2.3.7 Fase VII. Depuración del software

A este nivel se prueba y se depura el programa hasta que el funcionamiento sea adecuado. Se caracteriza porque en ella se prueba el programa incluso bajo condiciones

extremas.

2.3.8 Fase VIII. Integración del hardware con el software.

Para continuar se prueban las conexiones entre el hardware y el software para confirmar que exista una correcta correspondencia entre ambos y que el sistema interactúa de manera eficiente.

2.3.9 Fase IX. Construcción del modelo definitivo y pruebas finales.

En esta última fase, se materializa un dispositivo final en circuitos impresos y los programas se compilan para llegar a llevar a cabo el funcionamiento general del sistema.

2.4 Cuadro y cronograma de actividades y recursos.

Una vez planteada la metodología utilizada, se muestra un cuadro con la planificación de las actividades realizadas para dar respuesta a los objetivos específicos presentados en el trabajo de investigación. Mostrando la concordancia entre éstos últimos con las fases de Angulo, J. (1986), así como los recursos a empleados para lograrlas. Además, se presenta un cronograma con las actividades ejecutadas durante el periodo académico Abril 2018- Agosto 2018.

Tabla 2.1 Análisis de costos del proyecto.

CUADRO DE PRECIOS DE LOS ELEMENTOS QUE INTERVIENEN EN EL SISTEMA				
	Tipo	Cantidad	Valor unitario (USD)	Valor Total (USD)
Dispositivos eléctricos	Batería de 9V recargable	1	14	14
	Resistencias	3	0,15	0,45
	Capacitores cerámicos	2	0,15	0,30
	Cristal	1	3	3
	Pulsador	1	0,25	0,25
	Conector de batería	1	1	1
	Conector de carga para la batería	1	1,25	1,25
	Dispositivos electrónicos	Módulo RFID RC-522	1	15
Bluetooth HC-06 genérico		1	10	10
Microcontrolador ATmega 328		1	15	15
Mifare1 S50		10	3,50	35
Regulador 7805		1	3	3
Cable de interfaz		1	3	3
Dispositivo mecánico	Placa	1	10	10
	Carcasa	1	75	75
	Base para circuito integrado	1	1	1
Desarrollo del software	Horas de trabajo para el desarrollo			450
Total				637.25

Nota: Elaborado por el autor

CAPÍTULO 3.

PROPUESTA

3.1 INTRODUCCIÓN

Este proyecto está orientado a mejorar la experiencia de compras de los usuarios de los supermercados mediante la utilización de las nuevas tecnologías que están en auge como los lectores de tarjetas RFID, esto ayuda a clasificar etiquetas en un rango determinado, con lo que el usuario final tiene la posibilidad de adquirir productos directamente y pagar por ellos con eficiencia y sin demora.

La tecnología RFID que es una comunicación mediante ondas de radiofrecuencia, que a diferencia de la tecnología antecesora de esta (código de barras) no necesita una línea de vista directa, no necesita una fuente de voltaje ya que esta posee su propia fuente de alimentación, las ondas son cortadas por el tag, con esto se consigue una identificación del tag que a su vez está asociada a un determinado producto.

El presente proyecto plantea el optimizar el tiempo de los compradores al evitar que realicen largas filas para facturar sus productos en los supermercados, ya que en horas pico las filas para facturar los productos adquiridos son muy largas y toma una gran cantidad de tiempo el realizar este proceso, ya que primero se pasa cada producto por el lector de código de barras, luego se factura y se empaca. Para el proceso antiguo cada cliente toma los productos de las perchas, los coloca en un coche y para finalizar su compra debe realizar una larga columna, en la cual el tiempo de espera depende de varios factores como la cantidad de productos de cada cliente, la agilidad del asistente de caja etc. Esto se convierte en un proceso poco eficiente tanto para la parte del cliente como para la del supermercado.

Este proyecto presenta una nueva alternativa a los usuarios al momento que se realizan compras y al facturarlas ya que el dispositivo permite mediante la tecnología RFID pasar el producto deseado por el lector RFID y mediante una aplicación Android instalada en un Smartphone, realizar la facturación de los productos de manera individual sin necesidad de un cajero y así evitar la demora de tiempo al momento de facturar compras realizadas, este proceso establece que cada cliente toma un producto previamente etiquetado con un tag, de aquí cada coche estará equipado con un módulo lector RFID y este a su vez enlaza con una aplicación desarrollada para Android y alojada en su Smartphone, esta aplicación indica al sistema facturador desarrollado como una aplicación distribuida que un producto ha sido seleccionado por el cliente, al final el cliente solo debe pasar por una caja especial de cobro simplemente a pagar su factura sin pérdidas de tiempo y optimizando el proceso.

En la realización de este proyecto se han estudiado varios dispositivos electrónicos buscando las características que se adapten a los objetivos planteados, por lo cual se han seleccionado los siguientes elementos:

3.2 HARDWARE

A continuación, se describen los elementos de hardware usados para el desarrollo del proyecto, sus características, especificaciones y la interrelación entre cada uno de ellos, así como los diagramas de bloques.

3.2.1 Diagrama de bloques

En la figura 3.1 se muestra el diagrama de bloques del prototipo a diseñar, el cual va desde la interfaz de usuario y sus capas posteriores hasta llegar a la aplicación web, que es la aplicación en la cual se realiza la factura, para su posterior pago.

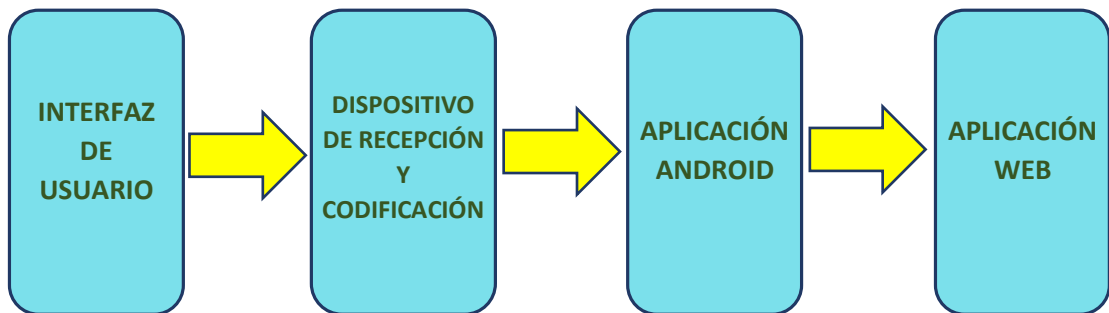


Figura 3.1 Diagrama de bloques

Fuente: Elaborado por el autor

En la figura 3.2 se muestran los módulos de lectura y transmisión de datos los cuales son los que registran la información contenida en el tag y la envían mediante un dispositivo Bluetooth hacia la aplicación en el celular.

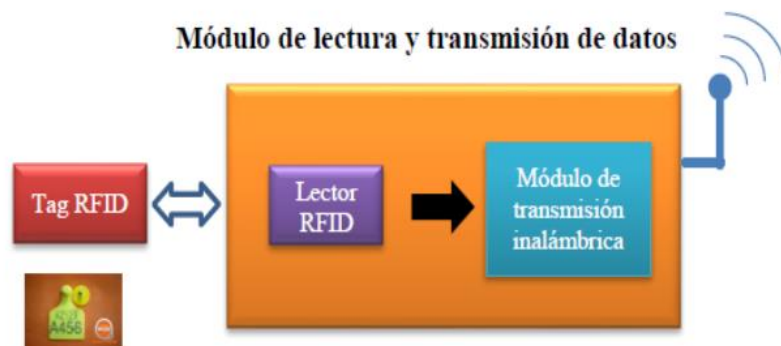


Figura 3.2 Modulo de lectura y transmisión de datos

Fuente: Elaborado por el autor

En la figura 3.3 se muestra el módulo de recepción de datos el cual está compuesta por la aplicación Android la cual una vez que el dispositivo lee y envía la información contenida por el tag, la procesa y va generando la factura de compras y posteriormente la enviara a la página web del cajero para cancelar los valores adeudados por el cliente.

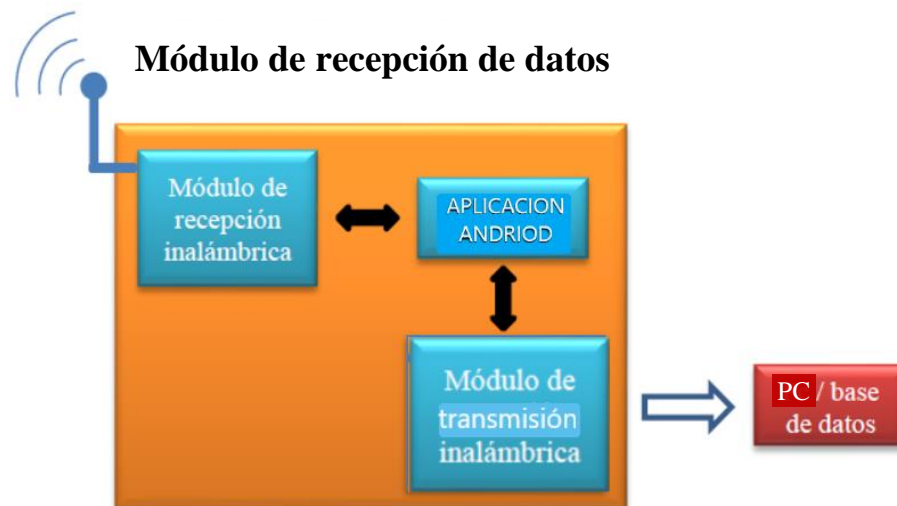


Figura 3.3 Módulo de recepción de datos

Fuente: Elaborado por el autor

El módulo de lectura y transmisión de datos, al igual que el módulo de recepción de datos está compuesto por los siguientes elementos:

3.2.2 Módulo Lector RFID RC522

Este es un módulo lector y grabador de Tags RFID, que utiliza 3,3 V como voltaje de alimentación y que se controla a través del protocolo SPI (Serial Peripheral Interface), por lo que se puede implementar con cualquier microcontrolador con interfaz SPI, lo cual ayuda para convertir la comunicación en full dúplex además de ser sumamente sencilla para su implementación. Este es el dispositivo en primera línea con el cliente, es este módulo el encargado de leer los códigos de los tags, para después ser procesados en las siguientes etapas. En la figura 3.4 se observa el módulo, como se observa la antena receptora consta de un bobinado impreso en la PBC, es aquí en donde se genera el campo en el que en el momento que se acerca un tag este es leído.

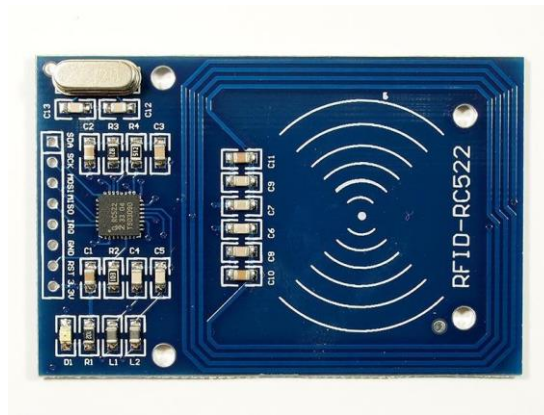


Figura 3.4 Lector RFID RC522

Fuente: <https://www.lelong.com>

3.2.3 Características del módulo lector RFID RC522 RF

- Corriente de operación: 13-26mA a 3,3V.
- Isb de stand by: 10-13mA a 3,3V.
- Frecuencia de operación: 13.56 MHz.
- Distancia de lectura: 0 a 60 mm.
- Protocolo de comunicación: SPI.
- Velocidad de datos máxima: 10 Mbit/s.
- Dimensiones: 40 x 60 mm.
- Temperatura de operación: -20 a 80°C.
- Humedad de operación: 5% - 95%.
- Máxima velocidad de SPI: 10Mbit/s
- Incluye llavero y TAG.

3.2.4 MIFARE1 S50

MIFARE es una tecnología que ha tenido un gran éxito en el mundo entero, se las conoce también por el nombre “tarjetas inteligentes” o “Tags”, son tarjetas de PVC que incluyen en su interior una antena que permite la comunicación de dicha tarjeta con un lector o receptor de radiofrecuencia (RFID) a una distancia que varía entre 2 cm y 10 cm, estas Tags trabajan en un rango de frecuencia desde 125 Khz a 13,56 Mhz.



Figura 3.5 MIFARE1 S50

Fuente: merkatronix.com/index.php?id_product=271&controller=product

En la figura 3.5 se muestra la MIFARE1 S50 que resuena a una frecuencia de 13,56 MHz.

3.2.5 Características de MIFARE1 S50

Características

- Tamaño pequeño, robustez y durabilidad, no se descolora
- A prueba de agua, a prueba de golpes, anticorrosión
- Disponible en diferentes colores
- Disponible en diferentes diseños
- Disponible en LF 125Khz y HF 13.56MHz
- Disponible en diferentes chips (TK4100, ATA5567, MIFARE 1 S50, MIFARE 1 S70, Ultraligero, Desfire, I-CODE SLI)
- Color personalizado y marca de inyección de tinta para el número de identificación o logotipo disponible a petición

3.2.6 Microcontrolador ATmega 328

Se trata de un circuito integrado, el cual tiene la capacidad de ejecutar instrucciones que se encuentran almacenadas en su memoria a través de un lenguaje de programación de alto nivel, sus principales bloques se dividen en unidad central de procesamiento, memoria y periféricos de E/S. Con esto se verifica que el microcontrolador es análogo a un computador.

Entre la amplia gama de microcontroladores AVR se ha seleccionado el modelo Atmega 328, el cual se puede observar en la figura 3.6 que es un microcontrolador de alto rendimiento que pertenece a la serie megaAVR, y posee arquitectura RISC. Su voltaje de funcionamiento se encuentra entre el rango de 1.8 - 5.5 voltios, con pines de entrada/salida programables y ciclos de reloj, esto ayuda a realizar la integración entre los dispositivos como el lector RFID y el módulo Bluetooth.

Usado normalmente en aplicaciones en las que se necesita control de sistemas de baja potencia y bajo costo, una de sus aplicaciones más comunes es como controlador de las placas Arduino, especialmente en los modelos Arduino Uno y Arduino Nano. Su programación se basa en la tabla de lenguaje en C, a través de un escritorio de programación amigable.

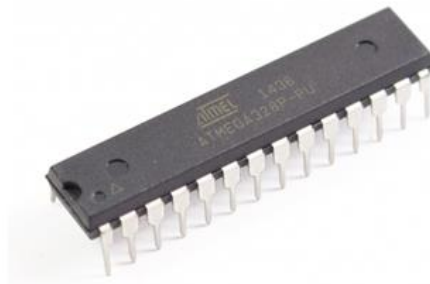


Figura 3.6 ATmega 328

Fuente: <http://www.naylampmechatronics.com/atmel/111-atmega328.html>

3.2.7 Características del ATmega 328

- Chip: ATMEGA328P-PU
- Formato DIP
- Pines: 28
- Memoria FLASH: 32KB
- Memoria RAM: 2KB
- EEPROM: 1KB
- Máxima frecuencia de funcionamiento: 20Mhz
- CPU: 8-bit AVR
- Pines de entrada/salida: 23
- Entradas Analógicas (ADC): 6

En la figura 3.7 se muestra la distribución de los pines para el microcontrolador, esta distribución es muy importante al momento de realizar la programación, ya que aquí se define la lógica de la programación, definiendo que pines se establecen como entrada y salida, alimentación VCC y GND.

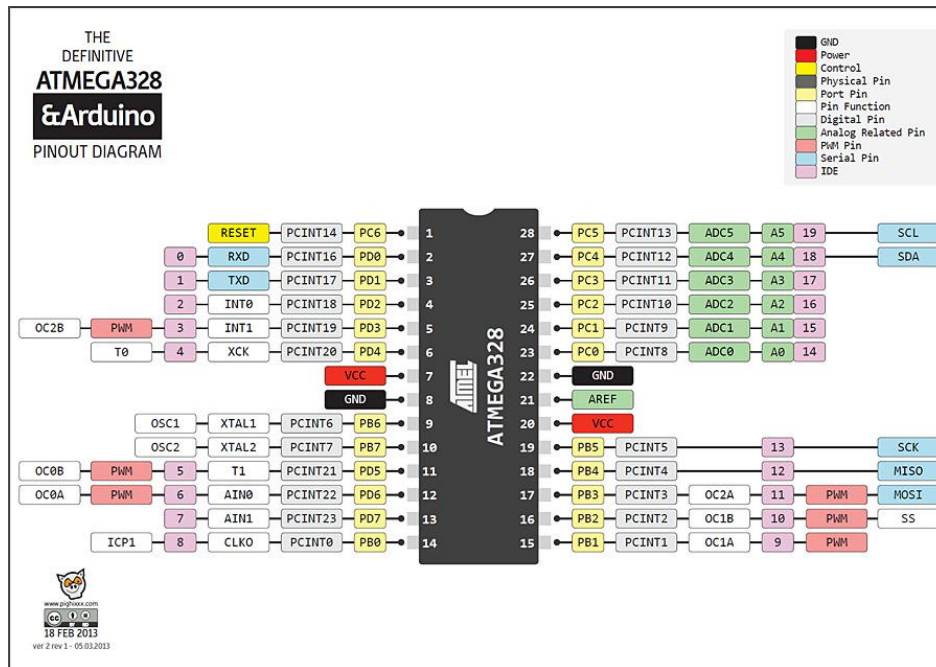


Figura 3.7 Distribución de pines Atmega328

Fuente: <http://saber.patagoniatec.com/atmega-328-micro-microcontrolador-chip-uno-nano-pro-mini-tutorial-arduino-arduino-argentina-ptec/>

3.2.8 Bluetooth HC-06

Es uno de los dispositivos más económicos del mercado, el modelo HC-06 solo puede actuar como esclavo, a diferencia del modelo HC-05 que puede actuar como maestro-esclavo. Esto quiere decir que el módulo será controlado por el dispositivo Smartphone, pero el dispositivo bluetooth no podrá controlar ningún dispositivo.

La comunicación Bluetooth es un estándar de comunicación en redes inalámbricas de área personal o WPAN (Wireless Personal Area Network), utiliza comunicación de radio en una frecuencia en la banda ISM de 2,4Ghz, su nombre y logo provienen de la parte mítica de tribus Noruegas, su principal objetivo es la de facilitar la comunicación entre dispositivos cercanos que manejen el estándar, estos dispositivos se deben encontrar en un rango de

máximo cuatro metros para un óptimo control, ya que el dispositivo RFID se encuentra en el carro de compras, no hay problema para la comunicación con el Smartphone así se genera una comunicación constante hasta que se genera a factura.

Utiliza un protocolo UART RS 232 serial, el cual es ideal para aplicaciones inalámbricas, además es fácil de implementar con PC, Microcontroladores o Arduino, se trata de códigos generados en 1L y 0L en un rango de 3,5v para el 1L.

La tarjeta posee un adaptador de 4 pines de fácil acceso para uso en protoboard, como se muestra en la figura 3.8 su antena se encuentra en la parte superior y se trata de una pista impresa en la PBC de cobre.

Este modelo de placa solo puede actuar como esclavo un modelo superior que puede actuar como master/slave es el HC-05, su tabla de programación incluye líneas básicas de comandos AT, configurado por default para trabajar en 9600 bps, cuando se realiza la configuración para conexión por default el nombre del dispositivo por defecto que se divide es HC-06, e código para vincular el dispositivo y el Smartphone es 1234 y el rate o velocidad 9600 bps, una vez que se realiza la vinculación el LED color azul inicia un parpadeo constante con lo que se verifica que existe la comunicación entre los dispositivos.

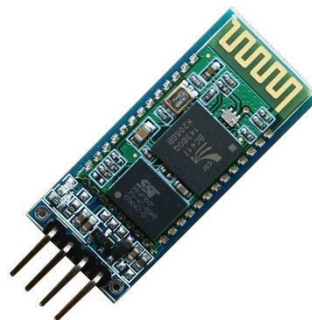


Figura 3.8 Bluetooth HC-06

Fuente: <https://www.prometec.net/bt-hc06>

Los pines del Bluetooth HC-06 correspondientes son:

- VCC
- GND

- RX
- TX

3.3 PHPMYADMIN

Un sistema gestor de base de datos, es un sistema de almacenamiento, consulta, y modificación de información estructurada y relacionada entre sí, consta de los siguientes elementos, gestor de base, sistemas de almacenamiento o servidores, su lenguaje y los usuarios. Su principal ventaja es que los scripts almacenados dentro de la plataforma pueden ser exportados y llevados para ser ejecutados en otras plataformas, esto gracias a que el lenguaje en común es SQL.

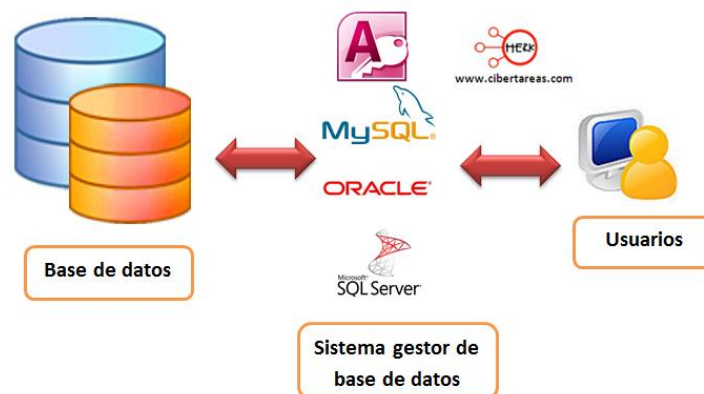


Figura 3.9 Sistema de base de datos

Fuente: <https://homeworkdatabase.wordpress.com/2015/06/27/sgbd-o-smbd-sistema-manejador-de-base-de-datos/>

Este sistema funciona de la siguiente manera; la información debe ser almacenada de manera relacionada en sistemas de almacenamiento en forma de tablas o tuplas, para que un usuario pueda realizar consultas a través de un gestor o de una interfaz, se realiza la programación de las tuplas con un lenguaje SQL es cual parametriza la información y con esto puede ser almacenada en dispositivos de almacenamiento sólidos, es importante el concepto de relación ya que para que el gestor funcione la información de las tuplas debe ser relacionada, por ejemplo en una tabla de cliente los datos internos pueden ser: nombres, cédula, teléfono, están relacionados ya que todos los datos hacen referencia al mismo ente,

existen diferentes gestores que manejan lenguaje SQL como MySQL, Oracle, SQLServer, Access.

SQL Structure Query Lenguaje o lenguaje estructurado de programación, es un lenguaje usado para para la gestión de bases de datos estructuradas dentro de sus principales características se puede observar que maneja:

- -lenguaje de definición de datos
- -lenguaje de manipulación de datos
- -Definición de vistas
- -Integridad
- -Control de vistas
- Dentro de los datos que maneja se pueden encontrar los siguientes:
- -Varchar
- -Date
- -Time
- -Datetime

En este proyecto se utiliza como gestor de la base de datos el programa phpMyadmin, es un gestor de acceso web en el cual el almacenamiento de la información se realiza en los servidores del comercio previamente configurados en la red de almacenamiento y acceso. Es importante que el administrador de la red sea la persona encargada de la seguridad de los servidores ya que la información que se almacena es de alto riesgo como datos de clientes, o tarjetas de pago.

Es una herramienta de software libre escrita en lenguaje PHP, destinada a manejar la administración de MySQL en la Web, puede crear, alterar y eliminar datos de la base de datos, en la figura 3.9 se muestra la pantalla del Software hphMyAdmin. La pantalla esta dividida en dos partes, en la parte izquierda se observa las diferentes bases de datos que se van creando y las que el programa tiene por defecto, en la parte central y derecha se observan los diferentes menús de configuración en los cuales se crean las bases de datos.

Como se observa en la figura 3.10 su principal ventaja es la administración vía web, con una interfaz gráfica amigable e intuitiva, permite importar diferentes scripts creados en otros gestores y manejarlos directamente.

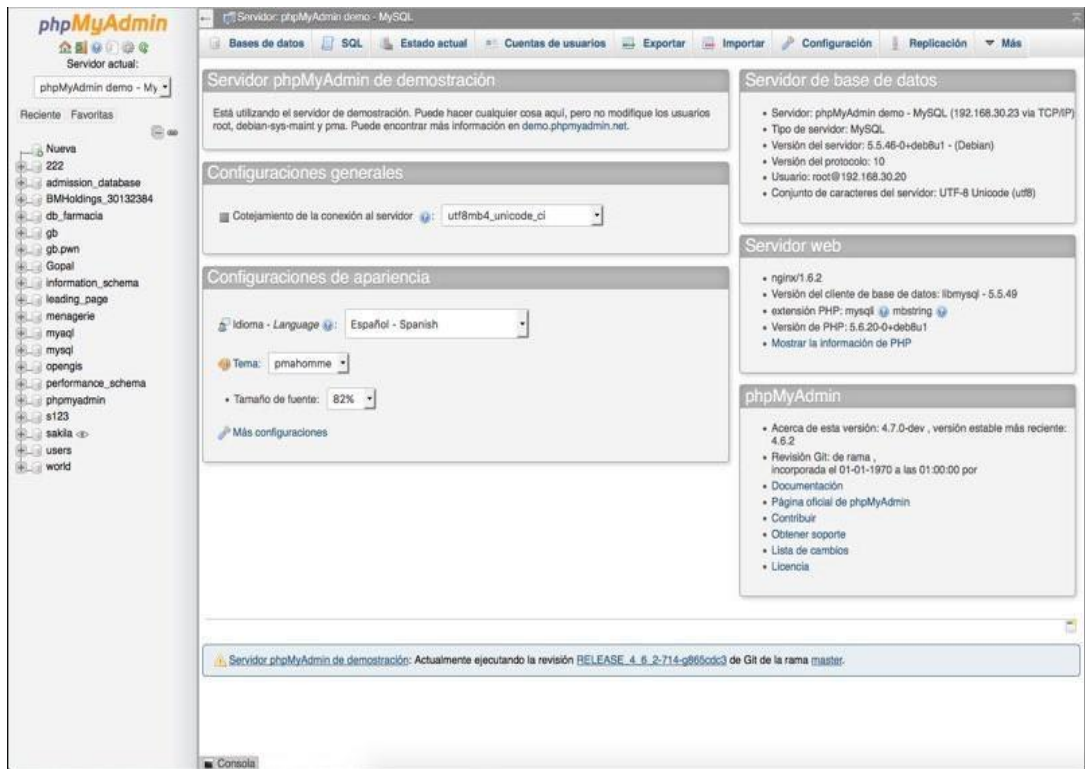


Figura 3.10 Pantalla de hphMyAdmin.

Fuente: www.hostinet.com/formacion/wp-content/uploads/2009/01/phpmyadmin

CAPÍTULO 4.

IMPLEMENTACIÓN

4.1 DIAGRAMA DE BLOQUES

Se inicia con el diseño de los principales bloques y la comunicación entre los mismos de esta manera se observa en la figura 4.1 sus principales componentes

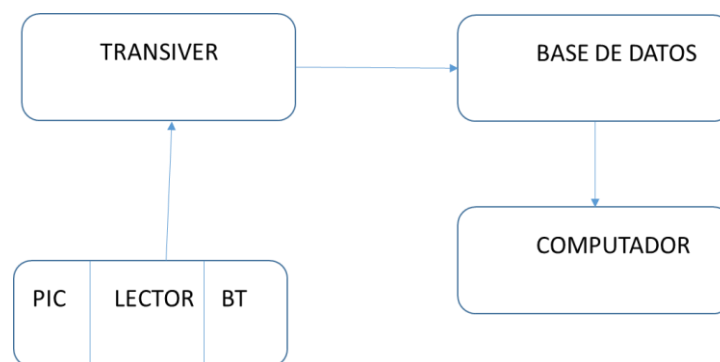


Figura 4.1 Diagrama en bloques

Fuente: Elaborado por el autor

El objetivo principal es facilitar al usuario de centros de comercio, la experiencia al momento de realizar sus compras reduciendo el tiempo y la molestia de hacer filas en las cajas para realizar los pagos.

De esta manera se describe el primer módulo el cual consta del lector RFID, el mismo que lee los tags que se encuentran pegados en los diferentes productos y previamente identificados en la base de datos, cuando el cliente inicia el proceso de compras, vincula su Smartphone cargado con la aplicación desarrollada en Android para este proyecto, y se registra con su código y su nombre de esta manera con lo que el cliente está listo para tomar un producto y que pueda realizar consultas de costo, stock y añadir el producto seleccionado

al sistema de compras para realizar su compra, cada vez que el cliente quiera un nuevo producto, debe realizar este proceso por cada producto que desee adquirir.

La base de datos es la responsable de verificar los productos y además es la encargada de generar la venta y la factura final del cliente, de esta manera se convierte en el módulo que interactúa con el cliente, el producto.

Los diferentes sistemas desarrollados realizan consultas y modificaciones en el servidor web, en un comercio este servidor puede estar alojado dentro del cuarto de servidores, en este proyecto como ya se explicó el servidor web que se está utilizando es el servidor apache con la dirección: <http://192.168.1.6:8080> .

Finalmente, con los datos que se almacenan en la base de datos, es decir cliente, productos, con todos sus atributos, se genera una factura la cual el cliente cancela en una caja especial (express) para realizar pagos de manera ágil y eficiente.

4.2 DIAGRAMA DE FLUJO

Un diagrama de flujo describe un proceso de cualquier tipo, es usado para observar las diferentes etapas de una tarea, involucrando todas las posibilidades de falla, con esto se mejora el análisis de fallas e inconvenientes, aún estando en etapa de desarrollo, facilitando el proceso de implementación

Se describen los principales procesos involucrados en el proyecto, de esta manera se define el flujograma de los componentes, así como del proceso que sigue el cliente desde la elección de producto, hasta que se registra para el pago de la factura

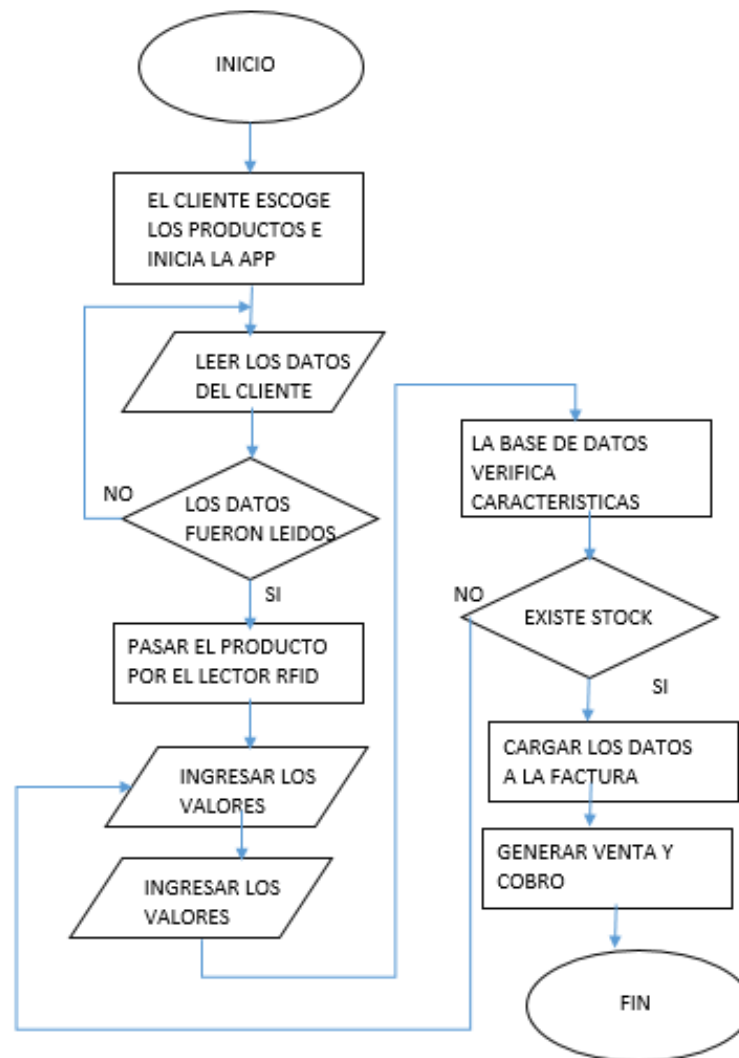


Figura 4.2 Diagrama de flujo del proyecto

Fuente: Elaborado por el autor

El proceso detallado, es el proceso de venta de productos, con sus diferentes variantes de acuerdo a las posibilidades con las que se encuentre el cliente, de esta manera se obtiene el flujograma que detalla el proceso de compra a través del proyecto lector con RFID.

4.3 DISEÑO DE LA PLACA

Para el diseño de la placa se toma como referencia el diseño esquemático del circuito de interconexión entre los dispositivos y el microcontrolador. Para esto se realiza la simulación con el programa Proteus en su versión 8.6, que se puede apreciar en la figura 4.3.

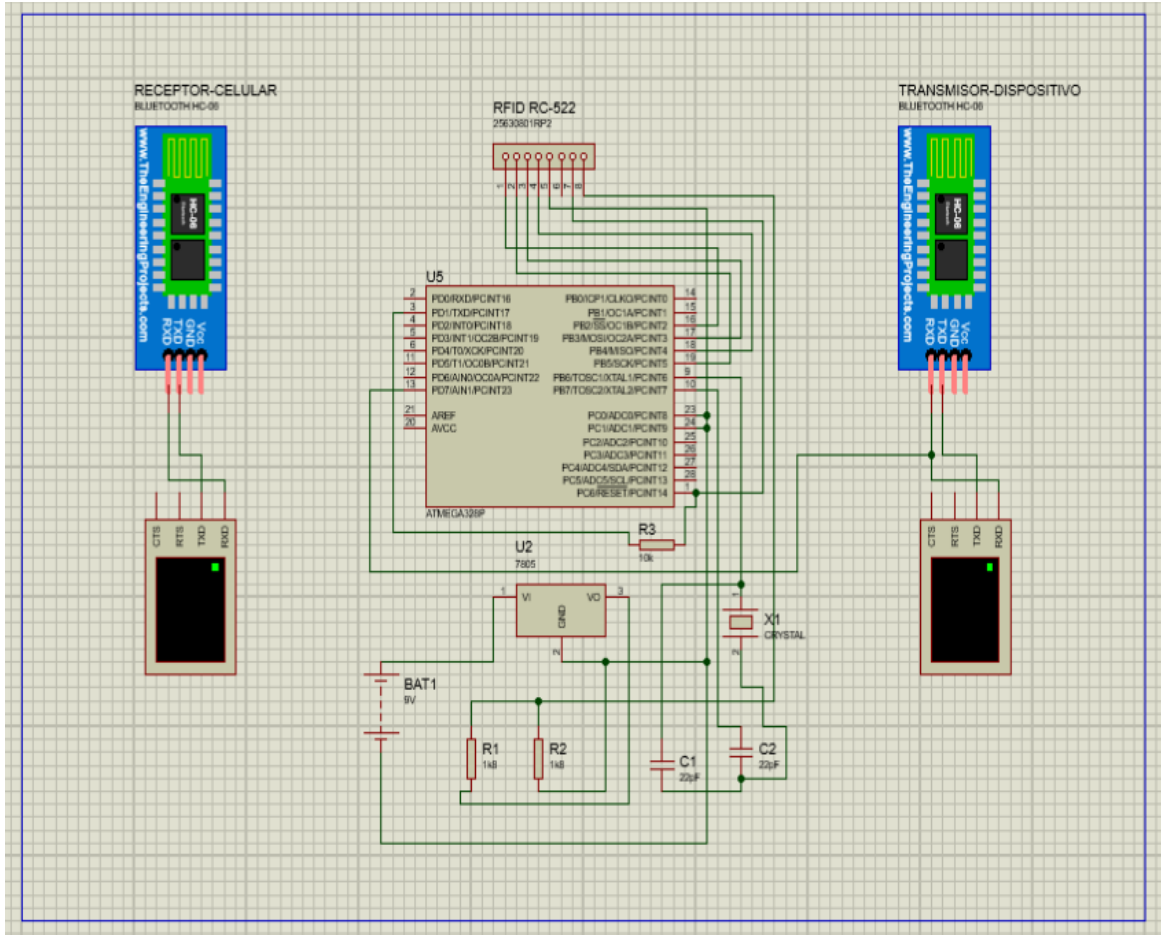


Figura 4.3 Circuito esquemático de interconexión

Fuente: Elaborado por el autor

El siguiente paso es convertir este circuito esquemático en un circuito para la creación de la PBC, esto se lo realiza a través del programa Ares que está incluido en las opciones del Proteus, de esta manera es fácil pasar del circuito esquemático al circuito impreso. Para luego realizar el proceso de fabricación del circuito en la placa, como se muestra en la figura 4.4.

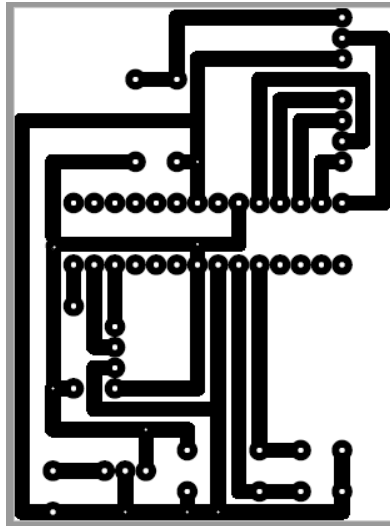


Figura 4.4 Diagrama impreso

Fuente: Elaborado por el autor

Una vez que el circuito impreso está listo se debe imprimir el archivo con una impresora láser en acetatos o en papel transfer, de esta manera el circuito se debe transferir a la baquelita por medio de calor, en la figura 4.5 se observa el diagrama PCB impreso en papel termotransferible.

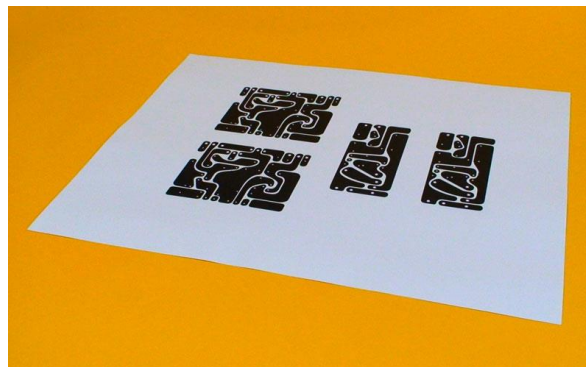


Figura 4.5 Impresión en papel transfer

Fuente: Elaborado por el autor

La baquelita debe ser preparada para ser sometida a calor, así que se debe dejar la superficie de la baquelita completamente limpia sin residuos de ninguna clase y de la medida adecuada, como se observa en la figura 4.6.

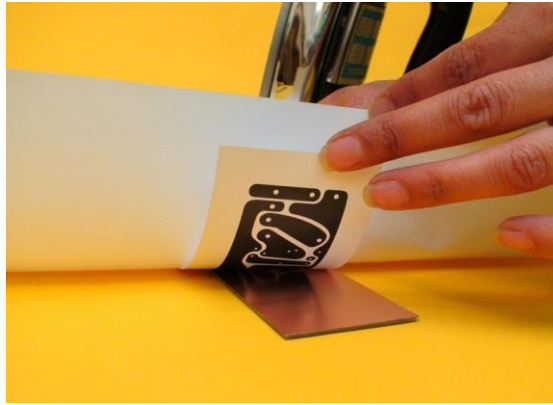


Figura 4.6 Transferencia del circuito

Fuente: Elaborado por el autor

En la figura 4,7 se muestra como se debe aplicar calor al papel transfer que se encuentra pegado a la baquelita, de esta manera la tinta se transfiere a la placa y se obtiene el diseño de circuito.



Figura 4.7 Transferencia a la placa

Fuente: Elaborado por el autor

Para quitar la parte de cobre no cubierta por la tinta transferida, se necesita quemar esta parte con ácido, esto se lo puede realizar con cloruro férrico, como se muestra en la figura 4.8.



Figura 4.8 Bandeja con cloruro férrico

Fuente: Elaborado por el autor

Para obtener la placa final existen dos procesos más, el primero consiste en realizar todos los agujeros para que los pines puedan ser soldados a la placa, y por último se deben soldar todos los elementos, esto se lo realiza con estaño y un cautín de 20-30 vatios



Figura 4.9 Proceso con el taladro

Fuente: Elaborado por el autor

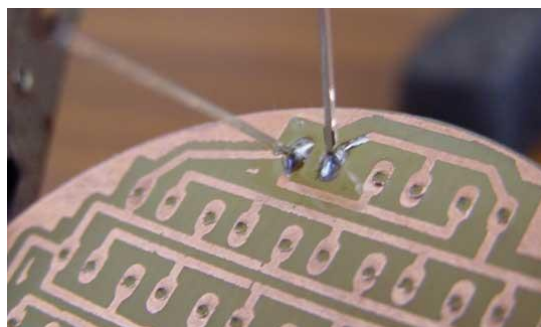


Figura 4.10 Soldadura de elementos

Fuente: Elaborado por el autor

El resultado final se observa en la figura 4.11 y 4.12, se muestran los elementos ya soldados a la placa tanto en su vista superior como en su vista inferior

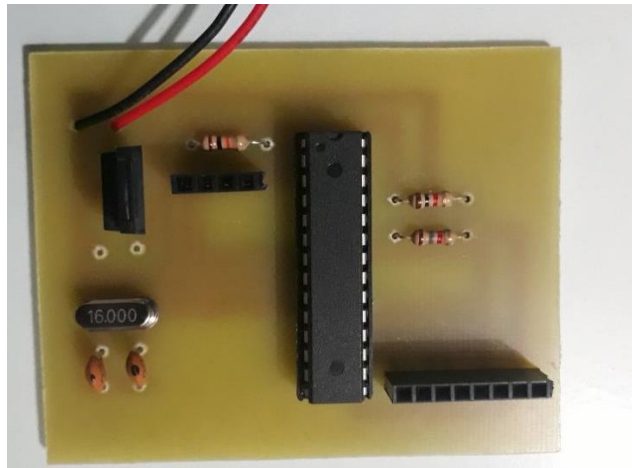


Figura 4.11 Top placa

Fuente: Elaborado por el autor

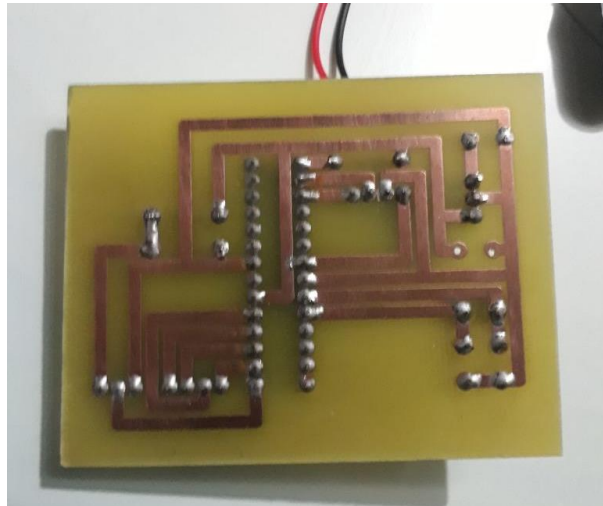


Figura 4.12 Bottom placa

Fuente: Elaborado por el autor

Ya que se trata de un proyecto electrónico es necesario el diseño y construcción de un contenedor el cual está construido en plástico que fue creado mediante una impresora 3D como se muestra en la figura. 4.13.

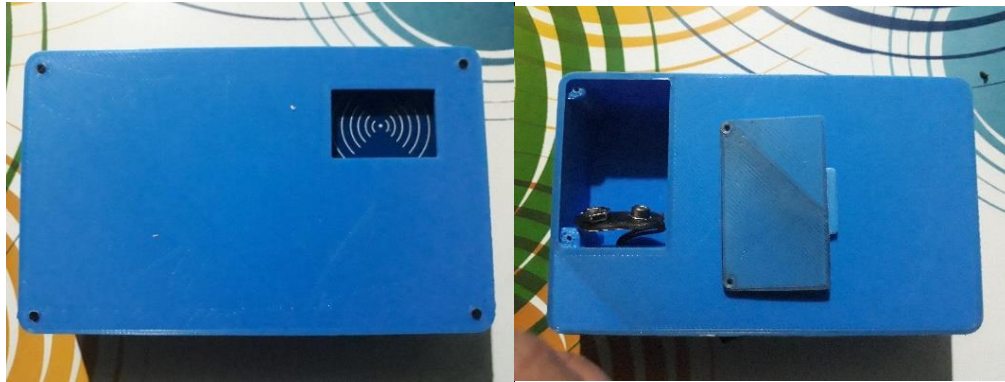


Figura 4.13 Vista superior e inferior del contenedor

Fuente: Elaborado por el autor

El contenedor está diseñado para evitar recalentamiento de las partes y para dar fácil acceso a los elementos para mantenimiento cuando se lo necesite, tiene dos compartimentos que separan la placa electrónica y la batería, este es el módulo es el que se añade al carro de compras y en el cual el cliente pasa los productos de los que va a realizar la compra.

4.4 PROGRAMACIÓN DEL AVR

El microcontrolador ejecuta un programa previamente almacenado en su memoria flash, este programa ejecutable está compuesto de una cadena de ceros y unos, para el AVR que se usa tiene una longitud de palabra de 8 bits, esto puede cambiar dependiendo del modelo y la estructura del CPU, su lenguaje de programación es de alto nivel esto permite mayor eficiencia al momento de escribir su código en comparación con lenguajes de bajo nivel que se convierten en una tarea más complicada, además que en lenguajes de bajo nivel, el programador debe conocer a fondo la estructura del microcontrolador.

En el proyecto se usa un lenguaje de alto nivel, el lenguaje C tiene la ventaja de ofrecer posibilidad de trabajo tanto a nivel de bytes como de bits

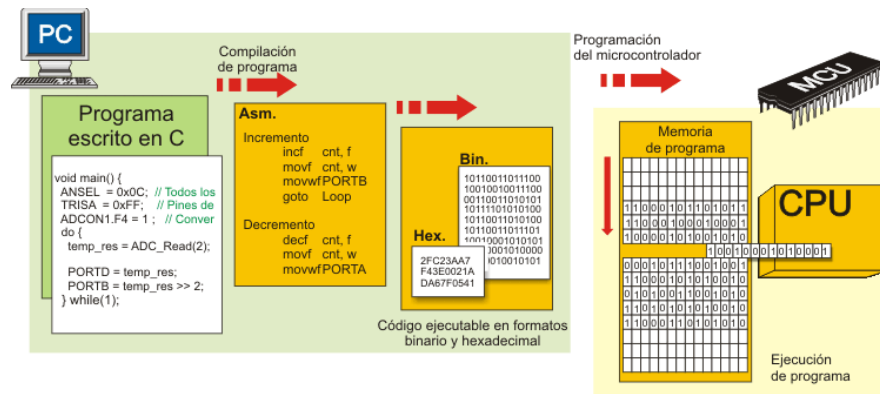


Figura 4.14 *Compilación en lenguaje C*

Fuente de: <https://www.mikroe.com/ebooks/microcontroladores-pic-programacion-en-c-con-ejemplos/lenguajes-de-programacion>

Para la programación del PIC se deben tomar en cuenta los elementos que interactúan en este módulo como son el microcontrolador, la tarjeta Bluetooth y el lector RFID, la programación de la tarjeta lectora RFID debe incluir las bibliotecas de códigos SPI y a biblioteca MFRC522

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

Se definen los pines 9 y 10 para el reset de la tarjeta y para el SS del RC 522 respectivamente, y se crea el objeto con el cual se inicia la lectura de las tarjetas.

```
constexpr byte RST_PIN = 9;
constexpr byte SS_PIN = 10;
MFRC522 mfrc522(SS_PIN, RST_PIN);
```

Dentro del código se inicia los procesos de lectura de las tarjetas iniciando la lectura en el puerto serial con una velocidad de 9600 baudios y con los protocolos SPI de esta manera.

```

void setup() {
  Serial.begin(9600);
  SPI.begin()
  MFRC522::StatusCode status;
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial()) {

```

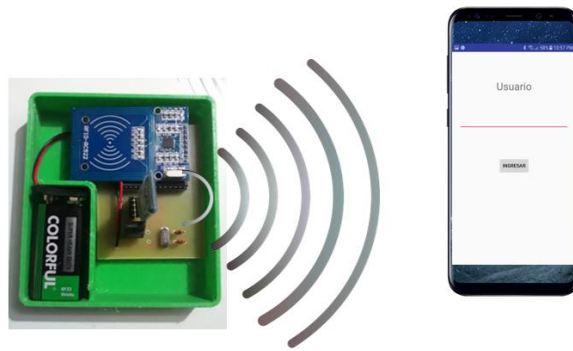


Figura 4.15 Bloque de conexión

Fuente: Elaborado por el autor

En tarjeta Bluetooth HC-06 se encuentran cuatro pines para su conexión los pines de alimentación Vcc y GND, además de dos pines señalados para la transmisión Tx y recepción Rx como se muestra en la figura 4.16.

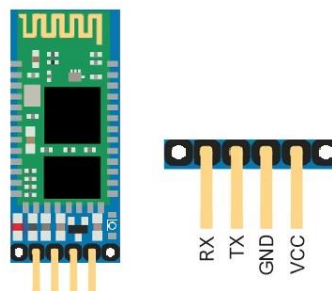


Figura 4.16 Pines de conexión módulo Bluetooth

Fuente: <http://www.martyncurrey.com/connecting-2-arduinios-by-bluetooth-using-a-hc-05-and-a-hc-06-easy-method-using-cmode/>

Los pines de Tx y Rx se conectan en forma cruzada con los pines de Tx y Rx del microcontrolador, en este caso se debe conectar el pin Rx de Bluetooth al pin número tres del microcontrolador y el pin Tx del módulo Bluetooth al pin número dos del microcontrolador.

De esta manera los datos obtenidos por el lector son interpretados por el PIC y enviados por el módulo Bluetooth hacia el celular, aquí es donde el cliente realiza las acciones que afectan la información hacia la base de datos.

4.5 PROGRAMACIÓN DE LA BASE DE DATOS

Dentro de la base de datos se definió el uso del gestor phpmyadmin para lo cual el primer paso es descargar el aplicativo de la página <https://www.phpmyadmin.net/downloads/> la versión que se está usando es la 4.7.8 que es compatible con PHP 5.5 al 7.2 y MySQL 5.5 y con las versiones las nuevas.

Ya que phpmyadmin es un gestor que trabaja con motor web se debe instalar un servidor web local para acceder al gestor, en este proyecto se usa el servidor Web Apache, se trata de un servidor HTTP de código abierto y multiplataforma el mismo que sirve como nexo para las diferentes consultas de los diferentes usuarios que están siendo servidos por la base de datos, en la figura 4.17 se observa el diagrama entre los gestores y como estos interactúan entre sí.

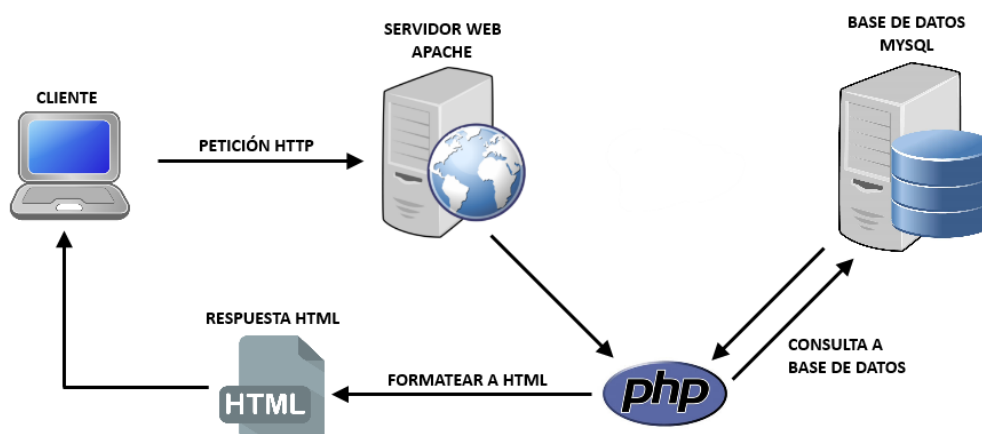


Figura 4.17 Diagrama de interacción

Fuente: El autor

Para la programación de la base de datos se usa el lenguaje SQL con su paquete de instrucciones correspondiente, mediante el gestor web phpMyAdmin el cual está escrito en PHP, este gestor es el encargado de administrar la base de stock y reconocimiento de los diferentes productos a medida que son registrados con el AVR. De esta manera las principales tablas creadas son: producto, cantidad, costo, id, cliente.

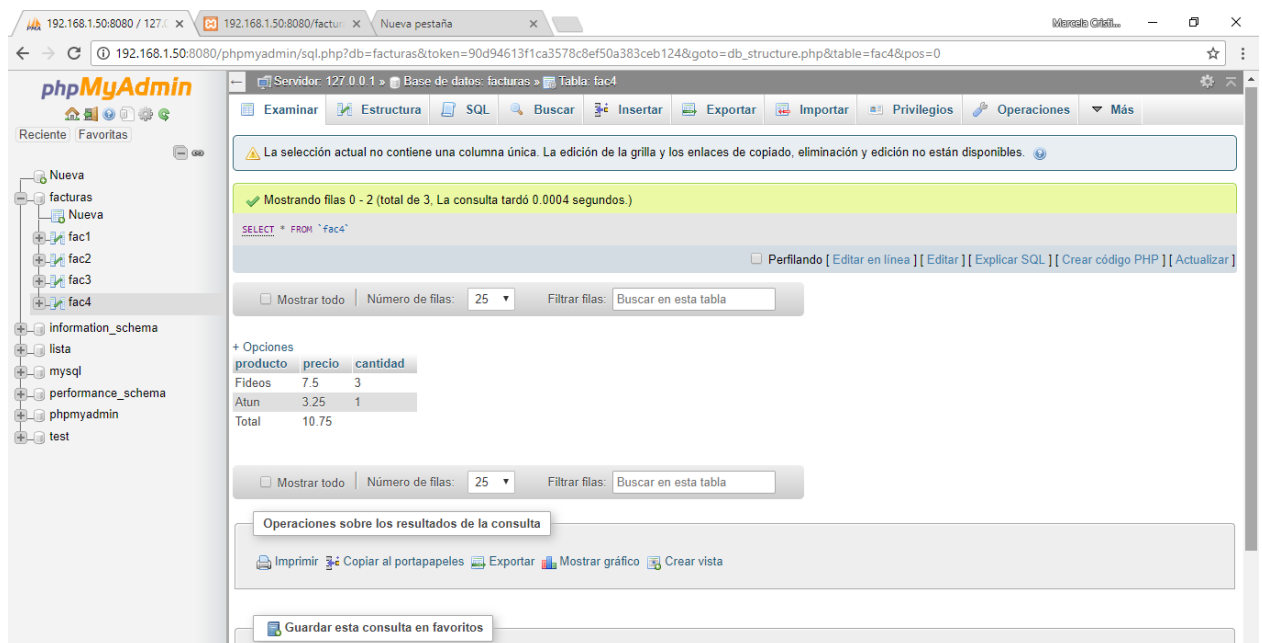


Figura 4.18 Base de datos

Fuente: Elaborado por el autor

El script de la base de datos está escrito en lenguaje SQL se debe tomar en cuenta los procesos que se pueden realizar sobre una base de datos, como consultar, modificar etc.

Dentro de las líneas de código una de las más importantes es la conexión con el servidor para lo cual se usa a siguiente línea de código. Se puede ver la conexión a la base local como usuario root dentro del repositorio facturas lo cual genera el trabajo dentro de la base seleccionada.

```
$scone=mysql_connect('localhost','root','');
$select=mysql_select_db('facturas',$scone);
```

```
$num="select count(*) as total from information_schema.tables where  
table_schema=".". "facturas".";
```

El lenguaje SQL incluye sentencias para realizar las alteraciones y consultas dentro de la base así como lenguaje de definición.

Para crear una tabla dentro del script se pueden utilizar las siguientes líneas de código en el cual se observa la definición de la tabla, el tipo de dato que se va a almacenar y la longitud máxima que puede tener, dependiendo de la información que se va a almacenar se usará diferentes tipos de datos.

```
$tabla="create table ". "fac". "$sal." (producto VARCHAR(30), precio  
VARCHAR(30), cantidad VARCHAR(30));
```

4.5.1 Creación de la base de datos escalable

Para la creación de la base de datos con la cual trabajara el dispositivo se debe seguir el siguiente proceso:

Primero ingresamos a la siguiente dirección: <http://192.168.1.6:8080/phpmyadmin/>, una vez que ingresamos a esta dirección en nuestro navegador se nos mostrará la vista del gestor phpMyAdmin., posteriormente se nos desplegara una ventana de donde se escoge el ITEM lista y a continuación seleccionamos nueva (fig. 4.19) .

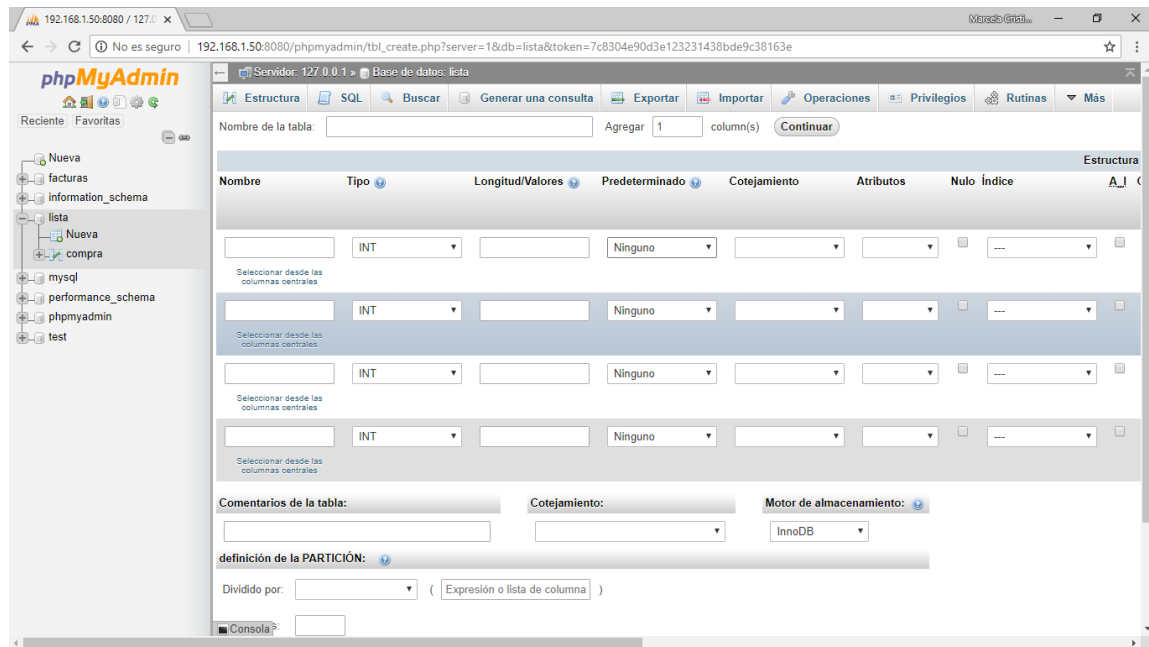


Figura 4.19 Pagina *phpMyAdmin*

Fuente: Elaborado por el autor

Una vez que aparezca la ventana (fig. 4.19) debemos asignarle un nombre a la tabla, el nombre y valores a las columnas que se quiere que tenga la tabla.

Cuando ya esté creada la tabla procedemos a agregar los productos, en la parte superior procedemos a buscar una pestaña con el nombre insertar y hacemos clic en esta, se desplegará una ventana donde ingresaremos los productos que irán en la base de datos con las siguientes características:

Producto.- aquí se escribe el nombre del producto.

Precio.- el valor del producto.

Nomimg.- aquí se agrega una imagen con extensión .jpg que es la que se mostrara en la aplicación.

Una vez realizado este proceso hacemos clic en continuar, y repetimos el proceso las veces que se quiera ingresar un producto nuevo a la base de datos como se muestra en la figura 4.20.

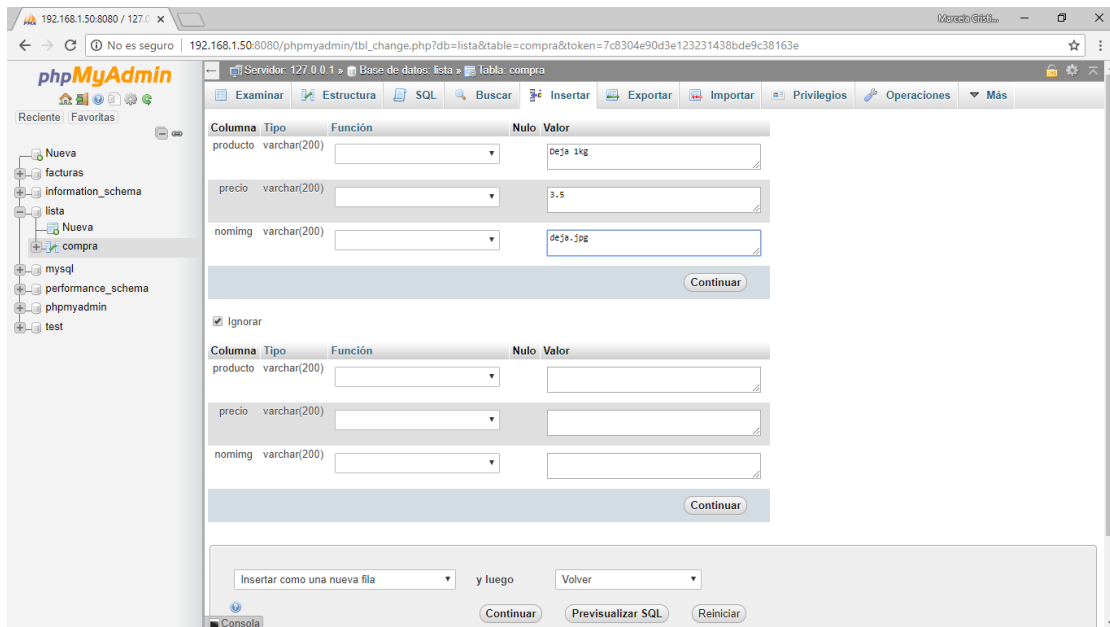


Figura 4.20 Página donde se ingresan los productos en la base de datos

Fuente: Elaborado por el autor

Una vez ingresados todos los productos requeridos ya está lista la base de datos para el dispositivo, en la figura 4.21 se muestra la base de datos con los productos creados.

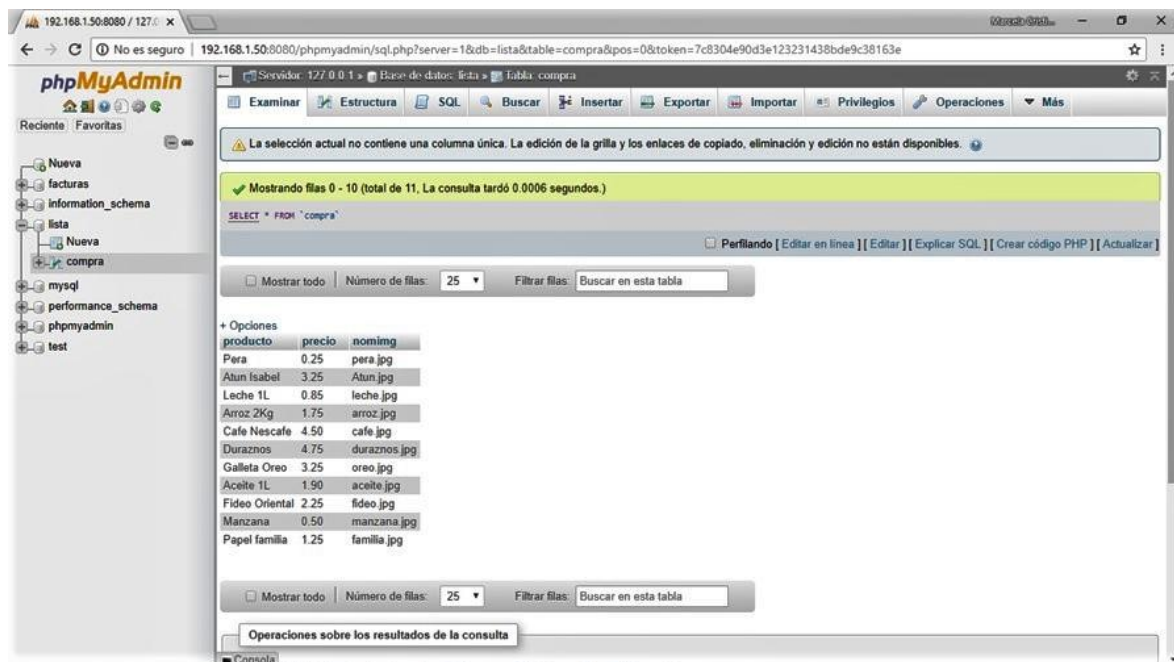


Figura 4.21 Base de datos creada en phpMyAdmin.

Fuente: Elaborado por el autor

4.6 PROGRAMACIÓN DE LOS SISTEMAS

Se deben tomar en cuenta dos sistemas principales, el primero de registro a través de la app de Android y la segunda que registra directamente la venta en el local comercial, la programación en Android se la realiza mediante el gestor Visual Studio el cual tiene como ventaja principal una comunicación en directo para comprobar el proceso de programación paso a paso en los dispositivos Android.

4.6.1 Programación en Android

La programación en Android se la realiza a través del aplicativo Visual Studio que es un entorno de desarrollo o IDE, el cual usa herramientas de desarrollo SDK (Software Development Kit), Android SDK comprende herramientas de depuración para varios núcleos del sistema Android, esto es de gran ayuda ya que facilita la distribución para diferentes modelos de teléfonos que corren bajos el sistema Android, en la figura 4.22se observa la pantalla de diseño del Android studio.

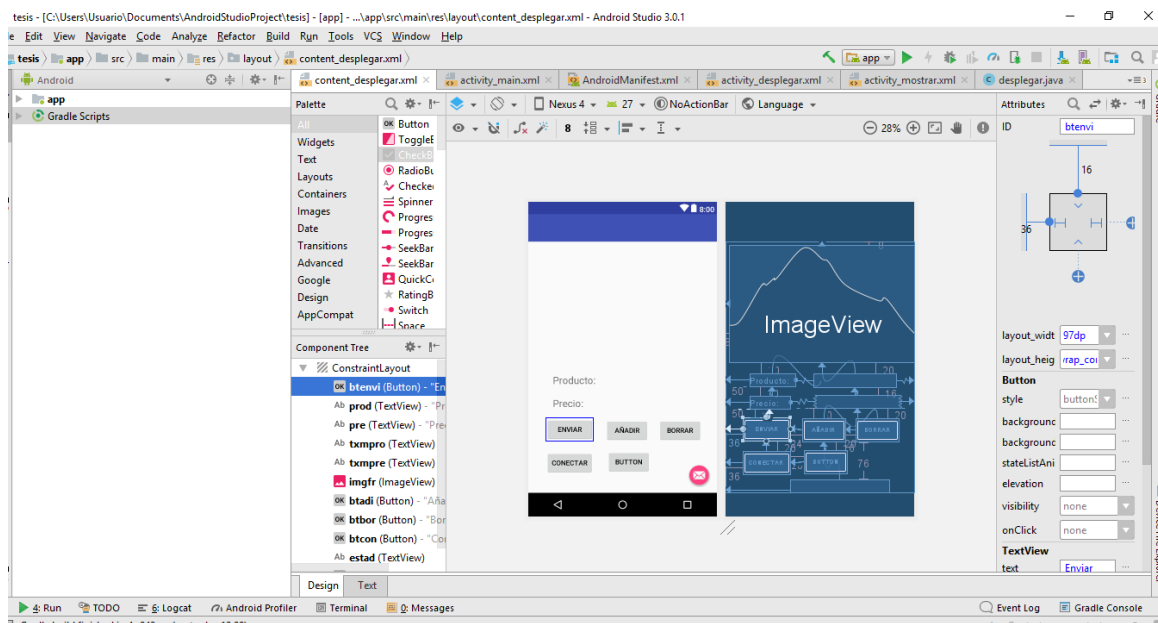


Figura 4.22 Programación en Android Studio

Fuente: Elaborado por el autor

Las programaciones de las aplicaciones en Android derivan de la misma lógica de la programación orientada a objetos es por eso que dentro de su programación se puede definir las siguientes funciones, la conexión con la base de datos, la integración con el dispositivo bluetooth y el envío de los productos seleccionados para realizar la factura.

El primer paso es importar las bibliotecas necesarias para realizar la programación dentro de paquete en el cual se haya definido el área de trabajo

```
package com.example.usuario.tesis;

import android.app.Activity;
import android.content.Intent;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
```

Se detallan los ejemplos de las bibliotecas principales, en este caso se necesitan estas bibliotecas para el uso de imágenes, textos, compatibilidad y para el uso de la comunicación por medio del dispositivo bluetooth.

En la siguiente línea se muestra la conexión con a base de datos general

```
public class desplegar extends AppCompatActivity {  
private String dir = "http://192.168.1.50:8080/";  
private String link = "http://192.168.1.50:8080/dbcom.php";
```

A continuación, se detallan las pantallas de acceso programadas en Android Studio y cargadas en el teléfono del cliente (ver figura 4.23). Una vez que el usuario se registra, accede al menú en el que verifica el estado de cada uno de los productos de los que se vaya a realizar la compra y que serán usados para realizar la factura.

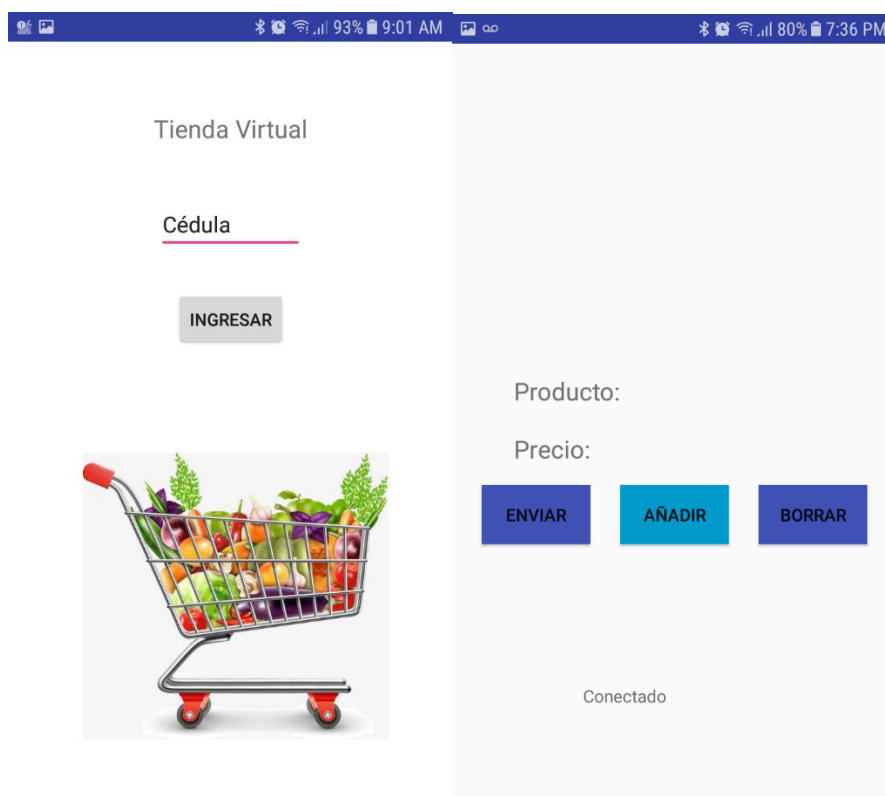


Figura 4.23 Pantallas de acceso al sistema Android

Fuente: Elaborado por el autor

4.6.2 Programación de la aplicación Distribuida

La aplicación web distribuida permite realizar la facturación, se la diseña en el gestor NetBeans el cual es multiplataforma y multilenguaje, de esta manera se programa para una aplicación web la cual esta almacenada en un servidor en el local comercial, y es esta la

interfaz desde la cual se registran las compras del cliente, como servidor web phpMyAdmin usa por defecto el servidor apache el cual también es usado por la aplicación de ventas.

Esta aplicación está escrita en lenguaje HTML que es un lenguaje basado en etiquetas, que se usa para el diseño de páginas en internet.

Cada etiqueta define diferentes áreas dentro de la programación por ejemplo la etiqueta <html></html> define el inicio y fin de la página, existe gran variedad de etiquetas, entre las principales se tienen:

- <head>
- <title>
- <style>
- <body>
- <h1>

Para instalar el gestor NetBeans se debe realizar de la página oficial de NetBeans <https://netbeans.org/downloads/> en la que se pueden descargar la aplicación para varias plataformas y dependiendo del trabajo que se realice hay varias opciones.

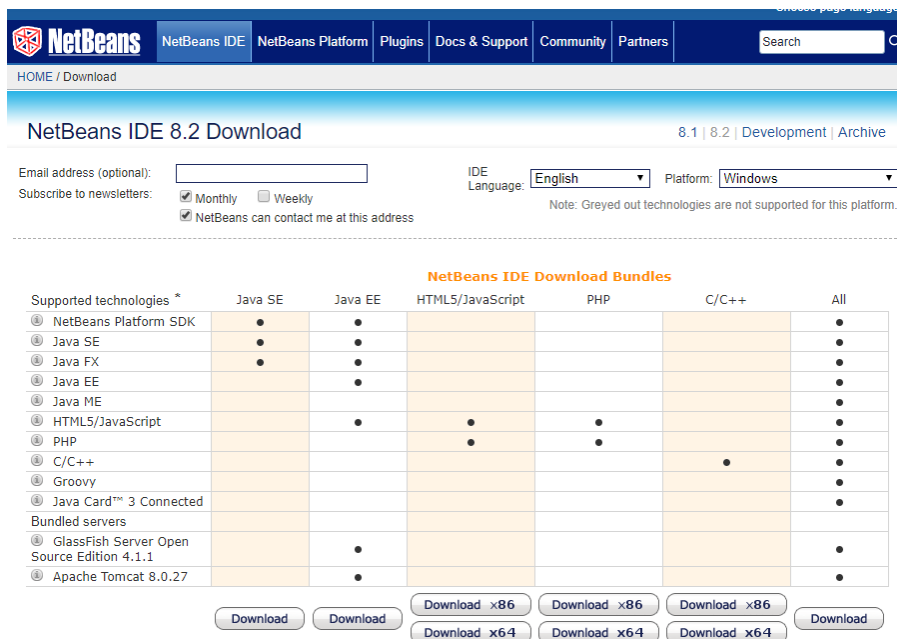


Figura 4.24 Página de descarga NetBeans

Fuente: Elaborado por el autor

Para el proyecto se descarga el aplicativo soportado por Windows y que soporta todas las tecnologías de desarrollo, con esto se garantiza que en el entorno de trabajo se podrán desarrollar diferentes tipos de aplicaciones en diferentes tecnologías. La versión de NetBeans con la que se trabaja es la 8.2, antes de instalar NetBeans, se debe instalar el Java Development Kit JDK este paquete de herramientas se puede descargar de la página principal de Oracle <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.92 MB	jdk-8u161-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.88 MB	jdk-8u161-linux-arm64-vfp-hflt.tar.gz
Linux x86	168.96 MB	jdk-8u161-linux-i586.rpm
Linux x86	183.76 MB	jdk-8u161-linux-i586.tar.gz
Linux x64	166.09 MB	jdk-8u161-linux-x64.rpm
Linux x64	180.97 MB	jdk-8u161-linux-x64.tar.gz
macOS	247.12 MB	jdk-8u161-macosx-x64.dmg
Solaris SPARC 64 bit (SUN4 microproc)	139.08 MB	jdk-8u161-solaris-sparcv9.tar.gz

Figura 4.25 Página de descarga JDK

Fuente:<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Este paso es importante ya que si el instalador de NetBeans no encuentra el JDK no instalará el entorno de desarrollo, se debe descargar las herramientas para el sistema operativo en que se vaya a trabajar, para este proyecto se debe descargar para un sistema operativo Windows de 64 bits y se inicia con la instalación. Se instala como cualquier programa.

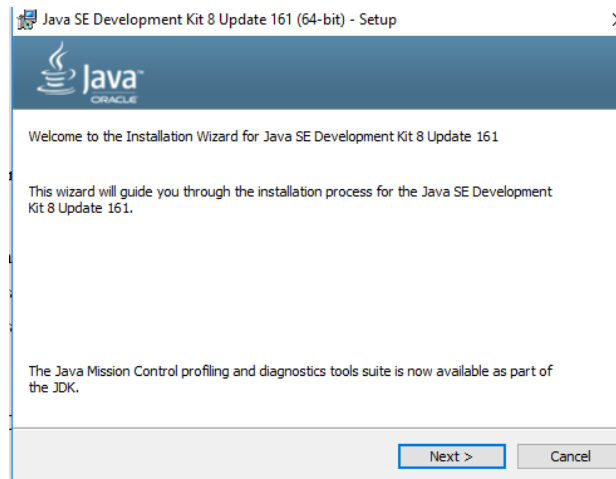


Figura 4.26 Instalación JDK

Fuente: Elaborado por el autor

Finalmente, cuando esté instalado aparecerá la pantalla indicando que las herramientas de desarrollo se instalaron correctamente. Con este paquete de herramientas instalado ya se puede continuar con la instalación de NetBeans.

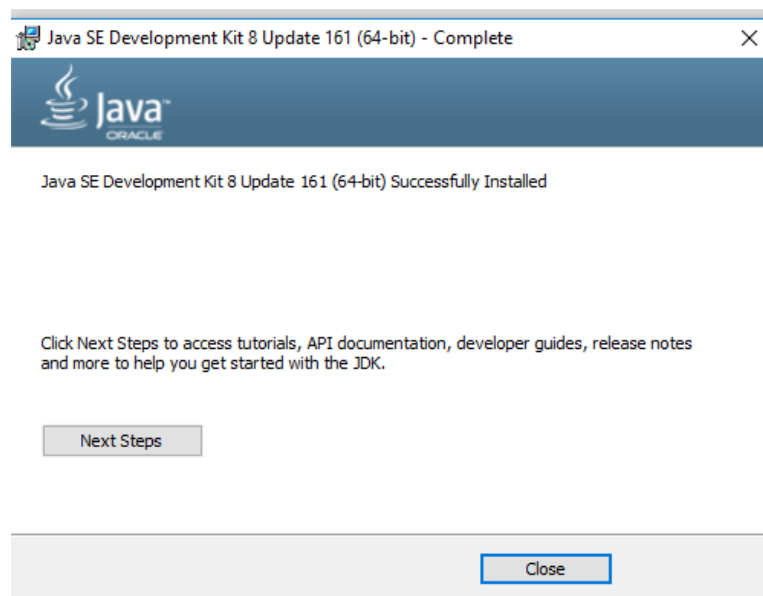


Figura 4.27 Pantalla de instalación JDK

Fuente: Elaborado por el autor

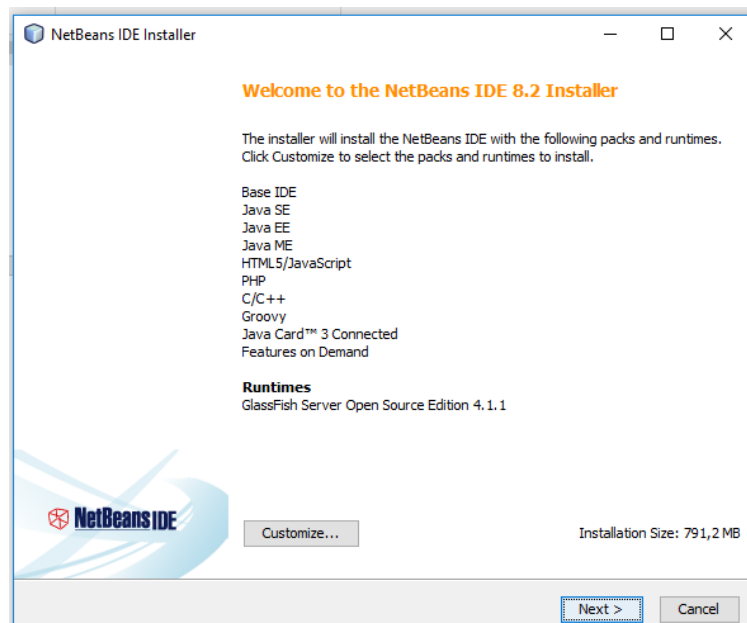


Figura 4.28 Pantalla de instalación NetBeans

Fuente: Elaborado por el autor

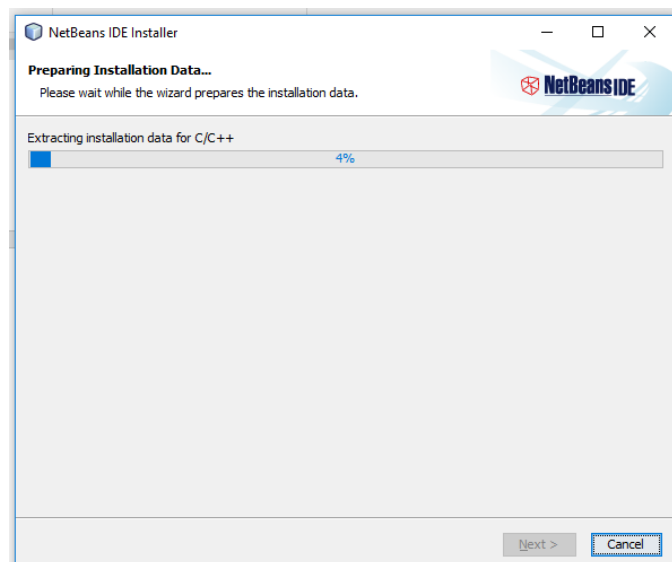


Figura 4.29 Progreso de instalación NetBeans

Fuente: Elaborado por el autor

Para crear un nuevo proyecto, se lo realiza en forma de empaquetado para un nuevo proyecto web.

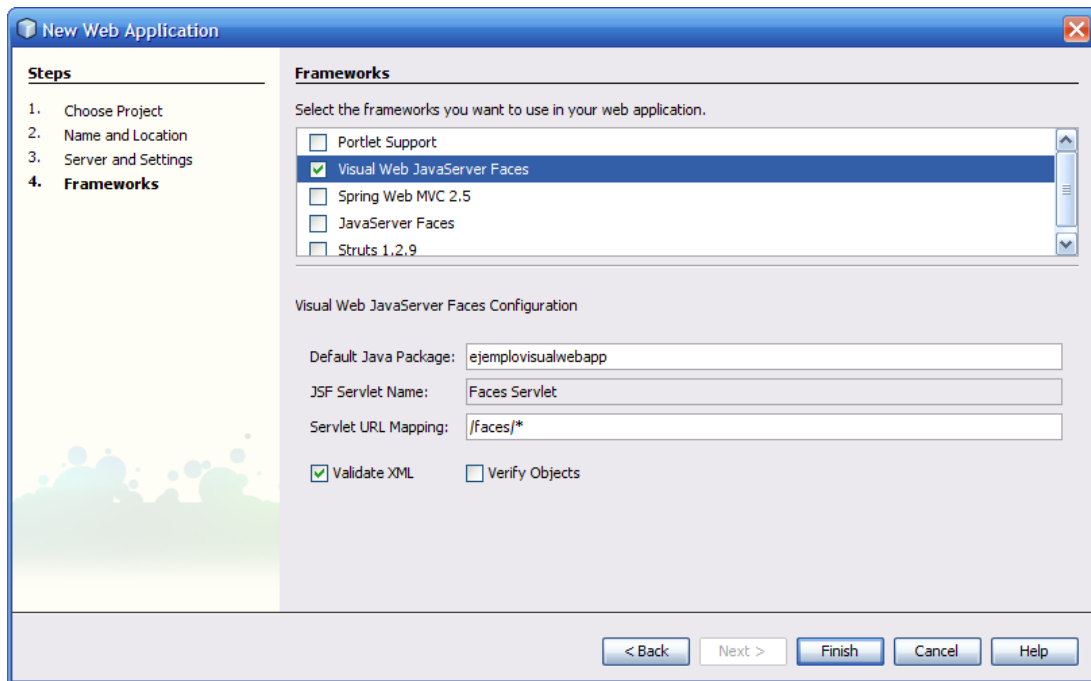


Figura 4.30 Pantalla de inicio de NetBeans

Fuente: Elaborado por el autor

Todo el proyecto para el esquema de facturación está escrito en lenguaje HTML por lo que se deben tomar en cuenta ciertos criterios, por ejemplo uno de los primeros pasos es la conexión entre la aplicación y la base de datos la cual se lo realiza con las siguientes etiquetas.

```

$con=mysql_connect('localhost','root','');
$select=mysql_select_db('facturas',$con);
$result=mysql_query("select* from fac".$nfac,$con);
echo "<table>";

```

Como se observa en la figura, en los proyectos en HTML se puede agregar formatos de alineación, color, y texto para presenta una mejor apariencia, por ejemplo, para agregar botones y colocarlos con margen se puede realizar con las siguientes líneas de código.

```

<body topmargin="0" leftmargin="0" marginwidth="0" marginheight="0">
    <div class="div1">

        <button class="boton bt1">
        <a href=prueba.php>Inicio</a>
        </button>
        <button class="boton bt2">
        <a href=factura.php>Factura</a>
        </button>
        </div>
    <div class="div2">

```

Cada una de las etiquetas que muestran el producto, precio y cantidad tienen un formato de letra color banca y esto se lo puede realizar con las siguientes líneas de código, como se observa si se cambia el estilo de letra de color blanco o negro se debe cambiar el Font color a black, y con esto cambia el color de la letra que se está usando en la aplicación.

```

color=white>CANTIDAD</font></th></tr>";
while($rs=mysql_fetch_array($result))
{
    echo "<tr>"
    ."<td><font color=white>".$rs['producto']."</font></td>"
    ."<td><font color=white>".$rs['precio']."</font></td>"
    ."<td><font color=white>".$rs['cantidad']."</font></td>"

```

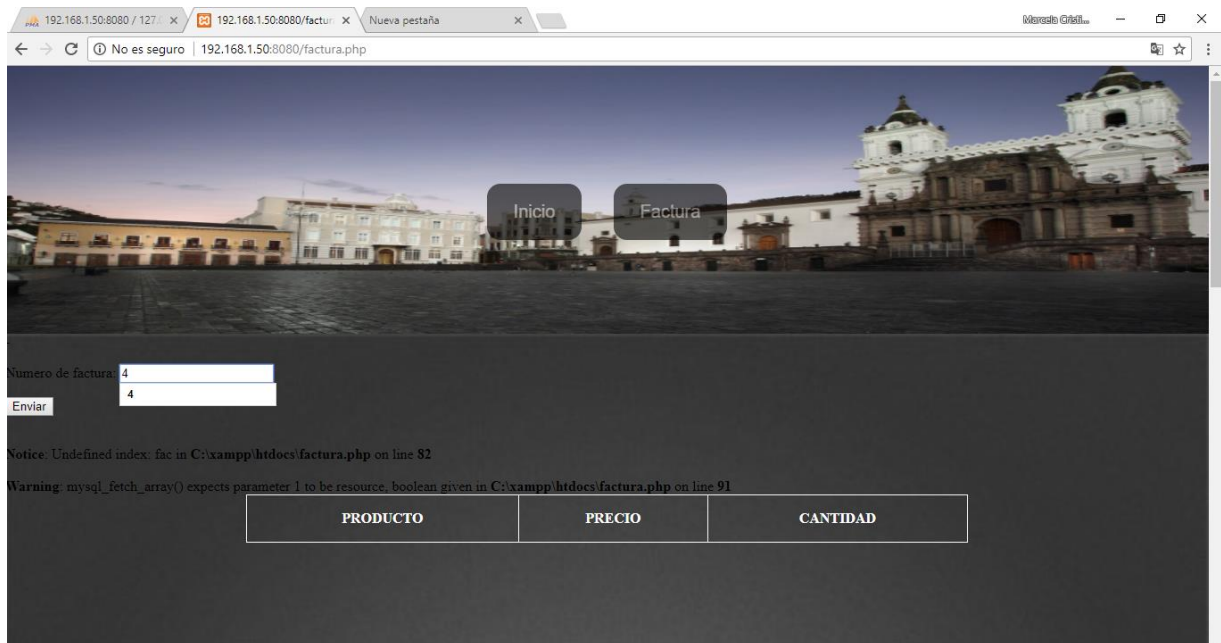


Figura 4.31 App web para venta

Fuente: Elaborado por el autor

Finalmente, la aplicación Web ayuda al proceso de facturación con los datos previamente almacenados en la base de datos y que son tomados desde la lectura de cada uno de los productos con el lector RFID y que pasan por la aplicación Android.

4.7 PRUEBAS DE FUNCIONAMIENTO

En un escenario de compras cada uno de los productos ya tiene un tag de identificación, cada cliente puede acceder a la aplicación Android en la cual registra su nombre e identificación para posteriormente cargar los productos a su cuenta, previamente el cliente puede verificar el costo del producto, con el botón añadir el producto seleccionado se registra para ser cargado en la factura mediante la aplicación web de esta manera cada cliente solo tiene que acercarse a la caja para realizar su pago ya que su factura está cargada en el sistema con todos sus productos y con el valor que debe cancelar



Figura 4.32 Dispositivo RFID

Fuente: Elaborado por el autor

4.7.1 Pruebas con las tags.

Se realizaron distintas pruebas con varios modelos de Tags. Para el desarrollo del sistema automático de facturación mostrado en el presente trabajo de titulación, se ha escogido las Tags tipo llavero estándar MIFARE1 S50 el cual trabaja en un rango de frecuencia desde 125Khz a 13,56Mhz, es un llavero Tag de proximidad que tiene una distancia máxima de detección de 10 cm.

Las pruebas realizadas en el proceso han sido satisfactorias ya que las lecturas de las tags colocadas en los distintos productos de prueba se las realizo sin problemas ya que ningún producto de estos presenta inconvenientes para que el lector pueda detectarlos.

Tabla 4.1
Pruebas de funcionamiento del dispositivo.

PRUEBAS DE FUNCIONAMIENTO				
Prueba Modo	Distancia sin Obstáculos (Metros)	Observaciones	Distancia con Obstáculos (Metros)	Observaciones
Conexión del prototipo con el celular	1	Sí	1	Si
	2	Sí	2	Si
	3	Sí	3	Si
	4	Sí	4	Si
	5	Sí	5	Si
	6	Sí	6	Si, con 2 seg. De retardo
	7	Si, con 2 seg. De retardo	7	Si, con 2 seg. De retardo
	8	Si, con 2 seg. De retardo	8	Si, con 2 seg. De retardo
	9	Si, con 2 seg. De retardo y se desvinculó el enlace por una sola vez.	9	Si, con 2 seg. De retardo
	10	Si, con 2 seg. De retardo	10	Si, con 2 seg. De retardo
Lectura de las Tags en el prototipo	Distancia sin Obstáculos (Cm)	Observaciones	Distancia con Obstáculos (cm)	Observaciones
	1	Funciona	1	Funciona
	2	Funciona	2	Funciona
	3	Funciona	3	No funciona
	4	No funciona	4	No funciona
	5	No funciona	5	No funciona
	6	No funciona	6	No funciona
	7	No funciona	7	No funciona
	8	No funciona	8	No funciona
	9	No funciona	9	No funciona
	10	No funciona	10	No funciona

Nota: Fuente: Elaborado por el autor

Tabla 4.2**Lista de chequeo del dispositivo.**

LISTA DE CHEQUEO			
DISPOSITIVO	ACTIVIDAD		OBSERVACIÓN
Batería de 9V	Cargada totalmente	✓	Mantener la batería cargada.
RFID RC522	LED indicador de actividad iniciada.	✓	Cuando la batería esta baja la intensidad del LED indicador disminuye.
Bluetooth HC-06	LED indicador de actividad iniciada	✓	Cuando la batería esta baja la intensidad del LED indicador disminuye.
	Parpadeo de LED indicador lento, vinculación con otro Bluetooth	✓	Cuando el parpadeo termina significa que el dispositivo está vinculado.
Samsung J5 PRIME	Aplicación instalada	✓	La aplicación APK es de fácil instalación en cualquier tipo de celular con sistema Android
Aplicación móvil	Inicio de sesión y envío de comandos primarios al dispositivo	✓	Cuando el dispositivo no está vinculado con el móvil la aplicación se cierra y solicita vincularse con el dispositivo.
Interruptor de encendido	Prueba de encendido	✓	El interruptor funciona correctamente.

Nota Elaborado por el autor

A continuación, se muestran las facturas obtenidas en cada una de las pruebas de compras, con el sistema funcionando el único proceso que realiza el comercio es el proceso de cobro en una caja especial en la que el cliente solo debe acercarse a realizar su pago, ya que a factura está cargada en el sistema

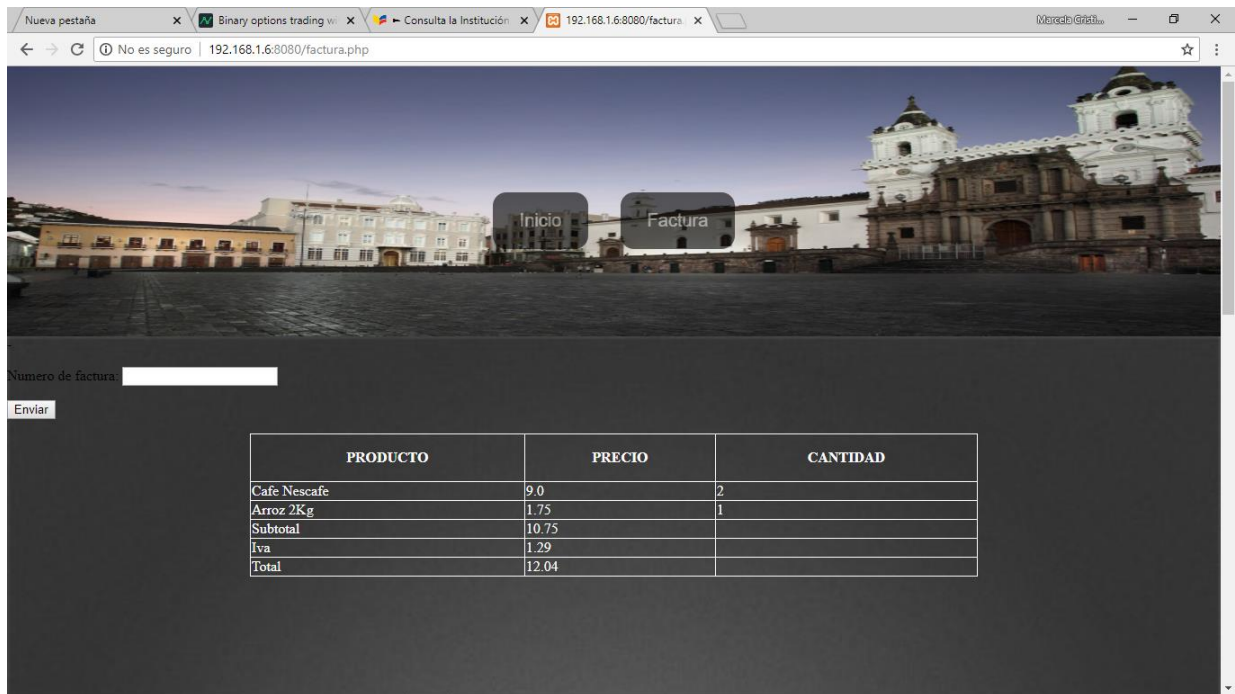


Figura 4.33 Pantalla de facturación

Fuente: Elaborado por el autor

CONCLUSIONES

Luego del análisis y discusión de los resultados, a continuación se presentan las conclusiones del presente estudio, las cuales permiten visualizar los hallazgos más importantes referidos al prototipo de un sistema automático de facturación con tecnología RFID y una aplicación Android para agilizar el proceso de pago en las cajas de un supermercado. Las mismas que se detallan con base en los objetivos específicos que se formularon para la investigación.

- Con respecto a definir las etiquetas (TAG´S) idóneas a utilizar de acuerdo a la frecuencia y forma, se encontró que los resultados fueron satisfactorios ya que las TAGS utilizadas en el sistema automático de facturación son a prueba de agua, resistentes a golpes, corrosión y tiene una durabilidad de 10 años aproximadamente.
- En relación a construir la placa electrónica del dispositivo de lectura de códigos del prototipo, los resultados revelan que se realizó un diseño compacto y funcional acordes a las especificaciones técnicas.
- Ahora bien, con relación a diseñar una aplicación Android que facilite la comunicación con el usuario, la aplicación Android facilita al usuario la interacción de este con el sistema de compras y reduce el tiempo que el usuario permanece dentro del supermercado.
- En cuanto a crear una base de datos escalable a todos los productos de un supermercado, la base de datos es el repositorio más importante, ya que dependiendo del estado de estos datos se generará la factura que finalmente cancelará el cliente.
- Con respecto a desarrollar la comunicación del dispositivo de lectura de códigos con la base de datos y con el teléfono mediante Bluetooth para la generación de la factura, los resultados revelan que todo el sistema está diseñado para ser convergente, entre sistemas

de software y hardware aplicando diferentes técnicas de programación e integrando todos los subsistemas para que la experiencia hacia el usuario final sea la más fácil e intuitiva

- Y para finalizar en relación a validar el sistema mediante pruebas de funcionamiento, las pruebas realizadas al prototipo han permitido validar el sistema automático de facturación mostrando una alta eficiencia al momento de trabajar con esta.

RECOMENDACIONES

En función de las conclusiones, a continuación se esbozan una serie de recomendaciones pertinentes al desarrollo del prototipo de un sistema automático de facturación con tecnología RFID y una aplicación Android para agilizar el proceso de pago en las cajas de un supermercado.

En tal sentido, se recomienda en primer lugar, que en próximos proyectos se debe adecuar el módulo lector directamente con un carrito de compras inteligente de manera que el sistema se integre de manera más eficiente.

Además se sugiere que la fuente de alimentación del dispositivo sea lo más exacta en la salida de voltaje, que no varíe mucho con la carga utilizada, porque de este voltaje dependerá el rango de identificación y alcance del dispositivo.

Así mismo, se sugiere que el ambiente de compras debe estar lo más limpio de señales parásitas posibles, esto ya que pequeñas interferencias en el sistema pueden interferir en la lectura de las tarjetas que se encuentran en cada producto.

De la misma forma ya que el sistema está diseñado para servicios de ventas de productos se recomienda adquirir diferentes tipos de tags que se adapten a las diferentes formas de los productos.

También se recomienda no sumergir el dispositivo en agua o cualquier otro líquido o fluido que afecte a los módulos que contiene en su interior.

Finalmente se sugiere analizar la posibilidad de una producción a gran escala del sistema automático de facturación, para proporcionar una disminución en los gastos operativos y en el tiempo utilizado por los usuarios de los supermercados.

BIBLIOGRAFIA

- Diego, G. J. (2012). *El Gran Libro de HTML5, CCC3 y JAVASCRIPT*. Barcelona: MARcombo.
- Jose Angulo Usategui, I. A. (2003). *Mocrocontroladores*. Madrid: MgGrae-Hill.
- Libera. (2010). *RFID: Tecnologia, Aplicaciones y Perspectivas*. Málaga, España: libreria Networks.
- Manish Bhuptani, Shahram Moradpour. (2005). *Guia de campo de RFID: Implementacion de sistemas de identificacion por radio frecuencia*. New Jersey: Sun Microsystems/Prentice Hall.
- Martinez, D. L. (2001). *Sistemas Operativos*. Buenos Aires.
- Observatorio Regionalo de la Sociedad de la Información. (2007). *Rfid: Tecnologia de identificacion por Radiofrecuencia y sus principales aplicaciones*. Madrid.
- Portilla Garcia Javier, B. N. (2008). *Tecnologia de identificación por radiofrecuencina (RFID): Aplicación en el ambito de la salud*. Madrid.
- Rafael Camps Paré, L. A. (2005). *Sotfware libre: Base de Datos*. Catalunya.
- Telectrónica. (2005). *Introducción a la identificación por radio frecuencia*. Buenos Aires: Telectrónica.
- Vasquez, S. G. (2015). *Elementos de sistemas de telecomunicaciones*. Madrid: Parainfo.

ANEXOS

CODIGOS DE PROGRAMACION

ANDROID STUDIO ACTIVITY DESPLEGAR.JAVA

```

package com.example.usuario.tesis;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.os.Handler;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.ProtocolException;
import java.net.URL;
import java.util.Set;
import java.util.UUID;
import org.json.JSONException;
import org.json.JSONObject;

public class desplegar extends AppCompatActivity {
    private String dir = "http://192.168.1.50:8080/";
    private String link = "http://192.168.1.50:8080/dbcom.php";
    private String msg;
    private String dimg;
    private String[][] mdat=new String[35][5];
    Bundle emen=new Bundle();
    private String mpro;
    private int cant;
    private float preci;
    private int maxv=30;
    private float tpreci=0;
    private String mfac;
    private String mpre;
    private String paspro;

```

```

private String paspre;
private String pascant;
private String pastot;
private String ultpro;
private TextView txtmpro;
private TextView txtmpre;
private TextView txttest;
private Button btnenvi;
private Button btnadi;
private Button btnbor;
//private Button btncon;
private Button btnsalida;
private ImageView imgv;
private Bitmap limg;
private boolean enco;
private int sdat;
    BluetoothDevice btdis;
    BluetoothAdapter btadap;
    BluetoothSocket btench;
    InputStream entcad;
    OutputStream salcad;
int bufpos;
byte[] lbuf;
volatile boolean detbt;
    Thread hilodt;
private String salida;
private String gupro=null;
private String gupre=null;
private String gucant=null;
    JSONObject datos= new JSONObject();
@Override
public void onSaveInstanceState(Bundle savedInstanceState) {
super.onSaveInstanceState(savedInstanceState);
    savedInstanceState.putString("gupro", paspro);
    savedInstanceState.putString("gupre", paspre);
    savedInstanceState.putString("gucant", pascant);
}
@Override
public void onRestoreInstanceState(Bundle savedInstanceState) {
super.onRestoreInstanceState(savedInstanceState);
gupro= savedInstanceState.getString("gupro");
gupre= savedInstanceState.getString("gupre");
gucant= savedInstanceState.getString("gucant");
}
@Override
public void onResume() {
super.onResume();
try
{
    encobt();
    abrbt();
}
catch (IOException ex) {
}
}
/* String ent="1";

    try {
        envdat(ent);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

```

    }
    try {
        entdat();
        txttest.setText(salida);
    }
    catch (IOException ex){

    }*/

}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_desplegar);
    txtmpro=(TextView)findViewById(R.id.txmpro);
    txtmpre=(TextView)findViewById(R.id.txmpre);
    imgv=(ImageView)findViewById(R.id.imgfr);
    btnenvi=(Button)findViewById(R.id.btenvi);
    btnadi=(Button)findViewById(R.id.btadi);
    btnbor=(Button)findViewById(R.id.btbtor);
    txttest=(TextView)findViewById(R.id.estad);
    //btncon=(Button)findViewById(R.id.btcon);
    btnsalida=(Button)findViewById(R.id.btsalida);
    btnenvi.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            try {
                datos.put("entr", "2");
                datos.put("cod", "0");
                datos.put("pro", gupro);
                datos.put("pre", gupre);
                datos.put("can", gucant);
            } catch (JSONException e) {
                e.printStackTrace();
            }
            sdat=1;
            conec();
        }
    });
    btnbor.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            agre(2);
        }
    });
    btnadi.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            agre(1);
        }
    });
    btnsalida.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            String ent="1";
            try

```

```

    {
        envdat(ent);
    } catch (IOException e) {
        e.printStackTrace();
    }
    entdat();

}
});
/* btnprub.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            datos.put("entr", "1");
            datos.put("cod", "1");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        sdat=0;
        conec();
    }
});*/
//Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
//setSupportActionBar(toolbar);

// FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
//fab.setOnClickListener(new View.OnClickListener() {
//    @Override
//    public void onClick(View view) {
//        Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
//            .setAction("Action", null).show();
//    }
//});
}

```

```

void agre(int a)
{
    if(gupro!=null)
    {
        String[] vecgupro=gupro.split(",");
        String[] vecgupre=gupre.split(",");
        String[] vecgucant=gucant.split(",");
        for(int i=0;i<(vecgupro.length)-1;i++)
        {
            mdat[i][0]=vecgupro[i];
            mdat[i][1]=vecgupre[i];
            mdat[i][2]=vecgucant[i];
            ultpro=vecgupro[(vecgupro.length)-2];
        }
    }
}

```

```

if(a==2){
    mpro=ultpro;
}
for(int i=0;i<maxv;i++) {
    if(mpro.equals(mdat[i][0]))
    {
        cant=Integer.parseInt(mdat[i][2]);
        preci=Float.parseFloat(mdat[i][1]);
    }
}

```



```

if(a==1) {
preci = (preci / cant) * (cant + 1);

cant++;
mdat[i][2] = String.valueOf(cant);
mdat[i][1] = String.valueOf(preci);
}

if(a==2) {
if(cant>1) {
preci = (preci / cant) * (cant - 1);
cant--;
mdat[i][2] = String.valueOf(cant);
mdat[i][1] = String.valueOf(preci);
} else{
mdat[i][0]=null;
mdat[i][1]=null;
mdat[i][2]=null;
}
}

//pnumi=i;
//enco=true;
i=maxv;
enco=true;
} else{
enco=false;
}
}
if(enco==false)
{
for(int i=0;i<maxv;i++)
{
if((mdat[i][0])==null)
{
mdat[i][0]=mpro;
ultpro=mpro;
mdat[i][1]=mpre;
mdat[i][2]="1";
i=maxv;
}
}

paspro="";
paspre="";
pascant="";
for(int i=0;i<maxv;i++)
{
if((mdat[i][0])!=null)
{
paspro=paspro+mdat[i][0]+",";
paspre=paspre+mdat[i][1]+",";
pascant=pascant+mdat[i][2]+",";
} else{
i=maxv;
}
}
String[] vectot=paspre.split(",");

```

```

tpreci=0;
for(int i=0;i<vectot.length;i++)
    {
tpreci=tpreci+Float.parseFloat(vectot[i]);
    }
pastot = String.valueOf(tpreci);
paspro=paspro+"Total";
paspre=paspre+pastot;
pascant=pascant+" ";
    Intent intent= new Intent(desplegar.this,mostrar.class);
emen.putString("PRO",paspro);
emen.putString("PRE",paspre);
emen.putString("CAN",pascant);
//emen.putString("TOT",pastot);
intent.putExtras(emen);
    startActivity(intent);
    }
private void conec()
    {
final Handler puente = new Handler();
final String salud;
        Thread hilo = new Thread(new Runnable() {
@Override
public void run() {
try
    {
        URL url = new URL(link);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setDoOutput(true);
        conn.setUseCaches(false);
        conn.setDoInput(true);
        conn.setRequestMethod("POST");
        conn.connect();
        DataOutputStream output = null;
//DataInputStream input = null;
        output = new DataOutputStream(conn.getOutputStream());
try {
            String envi = datos.toString();
            output.writeBytes(envi);
            output.flush();
            output.close();
        } catch (IOException ex) {
        }
        String codigoRespuesta = Integer.toString(conn.getResponseCode());
        System.out.println(codigoRespuesta);
if(codigoRespuesta.equals("200")){//Vemos si es 200 OK y leemos el cuerpo del mensaje.
msg = readStream(conn.getInputStream());
            System.out.println(msg);
final String[] msgm=msg.split(",");
ding=dir+msgm[2];
        puente.post(new Runnable() {
public void run() {
if(sd==0) {
mpro = msgm[0];
mpre = msgm[1];
                String[] vecmpro=mpro.split(">");
mpro=vecmpro[(vecmpro.length)-1];
txtmpro.setText(mpro);
txtmpre.setText(msgm[1]);
                    }else{

```

```

mfac=msgm[0];
        Intent intent= new Intent(desplegar.this,mostrar.class);
emen.putString("PRO",gupro);
emen.putString("PRE",gupre);
emen.putString("CAN",gucant);
emen.putString("FAC",mfac);
        intent.putExtras(emen);
        startActivity(intent);

    }
// nimg=msgm[2];
System.out.println(dimg);

    }
    });
    }
    conn.disconnect();
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (ProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
if(sdat==0) {
try {
        URL url = new URL(dimg);
        HttpURLConnection con = (HttpURLConnection) url.openConnection();
        con.setDoOutput(true);
        con.setUseCaches(false);
        con.setDoInput(true);
        con.connect();
        BitmapFactory.Options options = new BitmapFactory.Options();
        options.inSampleSize = 2;

limg = BitmapFactory.decodeStream(con.getInputStream());

        puente.post(new Runnable() {
public void run() {

imgv.setImageBitmap(limg);

                }
            });
        conn.disconnect();
dimg = "";

        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
});
hilo.start();
}

```

```

void encobt()
{
btadap = BluetoothAdapter.getDefaultAdapter();
if(btadap == null)
{
txttest.setText("bluetooth no disponible");
}

if(!btadap.isEnabled())
{
    Intent enableBluetooth = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(enableBluetooth, 0);
}

    Set<BluetoothDevice> pairedDevices = btadap.getBondedDevices();
if(pairedDevices.size() >0)
{
for(BluetoothDevice device : pairedDevices)
{
if(device.getName().equals("HC-06"))
{
btdis = device;
break;
}
}
}
txttest.setText("Bluetooth encontrado");
}
void abrbt() throws IOException
{
    UUID uuid = UUID.fromString("00001101-0000-1000-8000-00805f9b34fb"); //Standard
    SerialPortService ID
btench = btdis.createRfcommSocketToServiceRecord(uuid);
    //try {
btench.connect();
    //}
    // catch (IOException ex)
    //{
    //    txtmen.setText("Bluetooth no abierto");
    //}
salcad = btench.getOutputStream();
entcad = btench.getInputStream();
txttest.setText("Conectado");
}
void envdat(String msg) throws IOException
{
    //String msg = "1"; //myTextbox.getText().toString();
    //msg += "\n";
salcad.write(msg.getBytes());

    //txttest.setText("mensaje enviado");
}
void entdat()
{
final Handler puente = new Handler();
final byte delimiter = 10; //This is the ASCII code for a newline character
detbt = false;
bufpos = 0;
lbuf = new byte[1024];
hilodt = new Thread(new Runnable()

```

```

    {
public void run()
    {
while(!Thread.currentThread().isInterrupted() && !detbt)
    {
try
    {
int bytesAvailable = entcad.available();
if(bytesAvailable >0)
    {
byte[] packetBytes = new byte[bytesAvailable];
entcad.read(packetBytes);
for(int i=0;i<bytesAvailable;i++)
    {
byte b = packetBytes[i];
if(b == delimiter)
    {
byte[] encodedBytes = new byte[bufpos];
                System.arraycopy(lbuf,0, encodedBytes,0, encodedBytes.length);
final String dato = new String(encodedBytes, "US-ASCII");
salida=dato;
bufpos = 0;
puente.post(new Runnable() {
public void run() {

try {
datos.put("entr", "1");
datos.put("cod", salida);
datos.put("pro", "0");
datos.put("pre", "0");
datos.put("can", "0");
                } catch (JSONException e) {
                    e.printStackTrace();
                }
sdat=0;
                conec();

try {
btench.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }
else
    {
lbuf[bufpos++] = b;
    }
    }
}
catch (IOException ex)
{
detbt = true;
}
}
});
hilodt.start();
}

```

```

private String readStream(InputStream in) throws IOException{
//BufferedReader r = null;
BufferedReader r = new BufferedReader(new InputStreamReader(in));
    StringBuilder total = new StringBuilder();
    String line;
    while ((line = r.readLine()) != null) {
        total.append(line);
    }
    if(r != null){
        r.close();
    }
    in.close();
    return total.toString();
}
}

```

DESPLEGAR.XML

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context="com.example.usuario.tesis.desplegar"
tools:showIn="@layout/activity_desplegar">

<Button
android:id="@+id/btenvi"
android:layout_width="97dp"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:layout_marginStart="36dp"
android:layout_marginTop="16dp"
android:text="Enviar"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/pre"
tools:ignore="MissingConstraints" />

<TextView
android:id="@+id/prod"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_alignParentTop="true"
android:layout_marginStart="50dp"
android:layout_marginTop="20dp"
android:text="Producto: "
android:textSize="20dp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/imgfr"

```

```

tools:ignore="MissingConstraints" />

<TextView
    android:id="@+id/pre"
    android:layout_width="83dp"
    android:layout_height="wrap_content"

    android:layout_below="@+id/prod"
    android:layout_marginStart="50dp"
    android:layout_marginTop="20dp"
    android:text="Precio:"
    android:textSize="20dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/prod"
    tools:ignore="MissingConstraints" />

<TextView
    android:id="@+id/txmpro"
    android:layout_width="170dp"
    android:layout_height="29dp"
    android:layout_alignLeft="@+id/btenvi"
    android:layout_alignParentTop="true"
    android:layout_alignStart="@+id/btenvi"
    android:layout_marginEnd="50dp"
    android:layout_marginStart="50dp"
    android:layout_marginTop="20dp"
    android:textSize="20dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/prod"
    app:layout_constraintTop_toBottomOf="@+id/imgfr"
    tools:ignore="MissingConstraints" />

<TextView
    android:id="@+id/txmpre"
    android:layout_width="175dp"
    android:layout_height="0dp"

    android:layout_alignTop="@+id/pre"
    android:layout_marginEnd="50dp"
    android:layout_marginStart="50dp"
    android:layout_marginTop="16dp"
    android:textSize="20dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toEndOf="@+id/pre"
    app:layout_constraintTop_toBottomOf="@+id/txmpro"
    tools:ignore="MissingConstraints" />

<ImageView
    android:id="@+id/imgfr"
    android:layout_width="382dp"
    android:layout_height="240dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints" />

<Button
    android:id="@+id/btadi"

```

```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignTop="@+id/btbor"
android:layout_centerHorizontal="true"
android:layout_marginStart="24dp"
android:layout_marginTop="20dp"
android:text="Añadir"
app:layout_constraintStart_toEndOf="@+id/btenvi"
app:layout_constraintTop_toBottomOf="@+id/txmpre"
tools:ignore="MissingConstraints" />

```

```

<Button
android:id="@+id/btbor"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@+id/btenvi"
android:layout_marginStart="20dp"
android:layout_marginTop="20dp"
android:text="Borrar"
app:layout_constraintStart_toEndOf="@+id/btadi"
app:layout_constraintTop_toBottomOf="@+id/txmpre"
tools:ignore="MissingConstraints" />

```

```

<Button
android:id="@+id/btcon"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="36dp"
android:layout_marginTop="20dp"
android:text="Conectar"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btenvi" />

```

```

<TextView
android:id="@+id/estad"
android:layout_width="203dp"
android:layout_height="43dp"
android:layout_marginStart="133dp"
android:layout_marginTop="76dp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btadi" />

```

```

<Button
android:id="@+id/btsalida"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="28dp"
android:layout_marginTop="16dp"
android:text="Button"
app:layout_constraintStart_toEndOf="@+id/btcon"
app:layout_constraintTop_toBottomOf="@+id/btadi" />

```

```

</android.support.constraint.ConstraintLayout>

```

ACTIVITY MOSTRAR.JAVA

```

package com.example.usuario.tesis;

```



```

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.ListView;
import android.widget.TextView;

import java.util.ArrayList;

public class mostrar extends AppCompatActivity {
private Bundle llave=new Bundle();
private ListView lvp;
private TextView txtfac;
private String entpro;
private String entpre;
private String entcant;
private String entfac;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_mostrar);
llave = this.getIntent().getExtras();
lvp= (ListView)findViewById(R.id.listaprod);
txtfac= (TextView)findViewById(R.id.txtfac);
entpro=llave.getString("PRO");
entpre=llave.getString("PRE");
entcant=llave.getString("CAN");
entfac=llave.getString("FAC");
txtfac.setText(entfac);
// enttot=llave.getString("TOT");
String[] vecpro=entpro.split(",");
String[] vecpre=entpre.split(",");
String[] veccant=entcant.split(",");
ArrayList<items> lista = new ArrayList<items>();
items comp;
for (int i=0; i<vecpro.length;i++)
{
comp = new items(veccant[i],vecpro[i],vecpre[i]);
lista.add(comp);
}

itemadap adap=new itemadap(this,lista );
lvp.setAdapter(adap);

}
}

```

ARCHIVO PHP BASE DE DATOS SERVIDOR WEB

DBCOM.PHP

```

<?php
$json = file_get_contents('php://input');
$obj = json_decode($json);

```

```

$llav = $obj->{'entr'};
$cod = $obj->{'cod'};
$pro = $obj->{'pro'};
$pre = $obj->{'pre'};
$can = $obj->{'can'};
//$llav = data_get($obj,'entr');
//$cod = data_get($obj,'cod');
//$pro = data_get($obj,'pro');
//
//$pre = data_get($obj,'pre');
//$can = data_get($obj,'can');
//$llav=1;
//$cod=1;
if($llav==1)
{
    $con=mysql_connect('localhost','root','');
    $selec=mysql_select_db('lista',$con);
    $result=mysql_query('select * from compra');
    //$vec=mysql_num_rows($result);
    mysql_data_seek($result,$cod) ;
    $ext=mysql_fetch_array($result);
    echo ">".$ext['producto'].",".$ext['precio'].",".$ext['nomimg'].",". "fin";
    mysql_close($con);
}
if($llav==2)
{
    //$pizza = "porciÃ³n1,porciÃ³n2,porciÃ³n3,porciÃ³n4,porciÃ³n5,porciÃ³n6";
    $vpro = explode(",",$pro);
    $vpre=explode(",",$pre);
    $vcan=explode(",",$can);
    $lon = count($vpro);
    $cne=mysql_connect('localhost','root','');
    $select=mysql_select_db('facturas',$cne);
    $num="select count(*) as total from information_schema.tables where
table_schema=".""."facturas"."";
    $result=mysql_query($num);
    $data=mysql_fetch_assoc($result);
    $sal=$data['total'];
    $sal=$sal+1;
$tabla="create table "."fac".$sal." (producto VARCHAR(30), precio VARCHAR(30), cantidad
VARCHAR(30))";
    //$tabla="create table "."fac".$sal." (producto VARCHAR(30))";
    $restab=mysql_query($tabla,$cne);
    for ($i = 0; $i < $lon; $i++)
    {
        $stab="".$vpro[$i].",".$vpre[$i].",".$vcan[$i]."";
        $resu=mysql_query("insert into "."fac".$sal." (producto,precio,cantidad) values
(",".$stab.")", $cne);
    }
    echo $sal.".".""."."."."."."."";
    /*if($restab==0)sql
    {
        printf("No se ha podido crear la tabla<P>\n");
    }else{
        printf("La tabla se ha creado correctamente<P>\n");
    }*/
}

```

```
}
?>
```

PAGINA WEB PARA CONSULTA BASE DE DATOS FACTURA.PHP

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th {
border: 1px solid white;
margin-left: 20%;
}

table {
border-collapse: collapse;
width: 60%;
}

th {
height: 50px;
}

.div1,div {
margin-top: 0;
margin-left: 0;
width: 100%;
height: 300px;
background-image: url("mic.jpg");
background-size: 100% 100%;

}

.div2,div {
margin-top: 0;
margin-left: 0;
width: 100%;
height: 1500px;
background-image: url("fon.jpg");
background-size: 100% 100%;
background-repeat: no-repeat;

}

.boton,a:link,a:visited {
background-color: #000000;
border: none;
color: white;
border-radius: 15px;
padding: 10px 15px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 20px;
cursor: pointer;
```

```

opacity: 0.6;
}

.bt1 {
position: absolute;
top: 20%;
left: 40%;
}
.bt2 {
position: absolute;
top: 20%;
right: 40%;
}

</style>
</head>

<body topmargin="0" leftmargin="0" marginwidth="0" marginheight="0">
  <div class="div1">
    <button class="boton bt1">
      <a href=prueba.php>Inicio</a>
    </button>
    <button class="boton bt2">
      <a href=factura.php>Factura</a>
    </button>
  </div>
  <div class="div2">
    <form action="factura.php" method="post">
      <p>Numero de factura: <input type="text" name="fac" /></p>
      <p><input type="submit" /></p>
    </form>
    <?php
$fac = $_POST["fac"];
//echo $nom;
$con=mysql_connect('localhost','root','');
$select=mysql_select_db('facturas',$con);
$result=mysql_query("select* from fac".$fac,$con);
echo "<table>";
echo "<tr><th><font color=white>PRODUCTO</font></th>";
echo "<th><font color=white>PRECIO</font></th>";
echo "<th><font color=white>CANTIDAD</font></th></tr>";
while($rs=mysql_fetch_array($result))
{
echo "<tr>"
      . "<td><font color=white>".$rs['producto'].</font></td>"
      . "<td><font color=white>".$rs['precio'].</font></td>"
      . "<td><font color=white>".$rs['cantidad'].</font></td>"
      . "</tr>";
}
echo "</table>";
?>
  </div>
</body>
</html>

```

PROGRAMA ARDUINO LECTURA DE TAGS RFID

```

#include <SPI.h>

#include <MFRC522.h>

constexpr byte RST_PIN = 9;
constexpr byte SS_PIN = 10;
MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
}

void loop() {
  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
  byte block=1;
  byte len;
  MFRC522::StatusCode status;
  if ( ! mfrc522.PICC_IsNewCardPresent() ) {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial() ) {
    return;
  }
  byte buffer2[18];

  status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, &key,
  &(mfrc522.uid)); //line 834

  status = mfrc522.MIFARE_Read(block, buffer2, &len);

  byte num=buffer2[0];
  Serial.write(num);
  Serial.println();
}

```

```
delay(1000);  
mfr522.PICC_HaltA();  
mfr522.PCD_StopCrypto1();  
}
```



PRODUCT / PROCESS CHANGE NOTIFICATION (PCN)

PCN Number:	SC164305
Notification Subject:	CCB 2733.01 Final Notice: Implement Catalog Part Number (CPN) Changes to Atmel products
PCN Issue Date:	December 13, 2016
Change Category:	Logistics
PCN Status:	Final notification
Microchip/Atmel Parts Affected:	This change applies to virtually all Atmel products.
Description of Change:	Implement Catalog Part Number (CPN) convention changes to Atmel products. Please make the necessary revisions to ensure flawless order/entry and receipt of your products.
Pre and Post Change Information:	<p>Please see the attached affected catalog part number (CPN) list to identify the products affected and changes between the current and new catalog part numbers.</p> <p>Note: Many of the current parts required an additional code after the catalog part number, referred to as the SL code, which some customers may have been unaware of when ordering. To ensure accuracy after the implementation date please query your local sales office with questions if the part number you order is the same as the catalog part number identified in the first column of the attached files.</p>
Impacts to Data Sheet:	None
Change Impact:	None
Reason for Change:	To improve productivity as part of the integration of Atmel and Microchip.
Change Implementation Status:	In progress

ATMEL San Jose • 1800 Technology Dr, San Jose • CA 95110 USA



Implementation Date: January 1, 2017

Time Table Summary:

	September 2016					October 2016				→	January 2017
Workweek	36	37	38	39	40	41	42	43	44		1
Final PCN Issue Date									X		
Anticipated Implementation Date											X

Method to Identify Change: Catalog Part Number (CPN)

Qualification Report: Not Applicable

Data Sheet: To view the applicable data sheet please go to the [document library](#) found on [atmel.com](#)

Revision History:
October 27, 2016: Issued final notification.
December 13, 2016: Revised the final notification to include additional line items and revise existing line items in the affected parts list. The additions and revision fields are highlighted and font is bold. Please review the attached affected parts list.

Please contact your local [Microchip](#) or [Atmel](#) Sales Office or Distributor with questions or concerns regarding this notification.

ATMEL San Jose • 1600 Technology Dr, San Jose • CA 95110 USA

Description

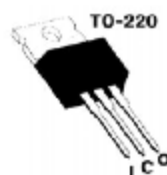
This series of fixed-voltage monolithic integrated-circuit voltage regulators is designed for a wide range of applications. These applications include on-card regulation for elimination of noise and distribution associated with single-point regulation. In addition, they can be used with power-pass elements to make high-current voltage regulators. Each The internal of these regulators can deliver up to 1.5 A of output current.

Features

- Output Current in Excess of 1.5A (PD²15W)
- No External Components Required
- Internal Short Circuit Current Limiting
- Internal Thermal Overload Protection
- Output Transistor Safe-Area Compensation
- Output Voltage Offered in 4% Tolerance

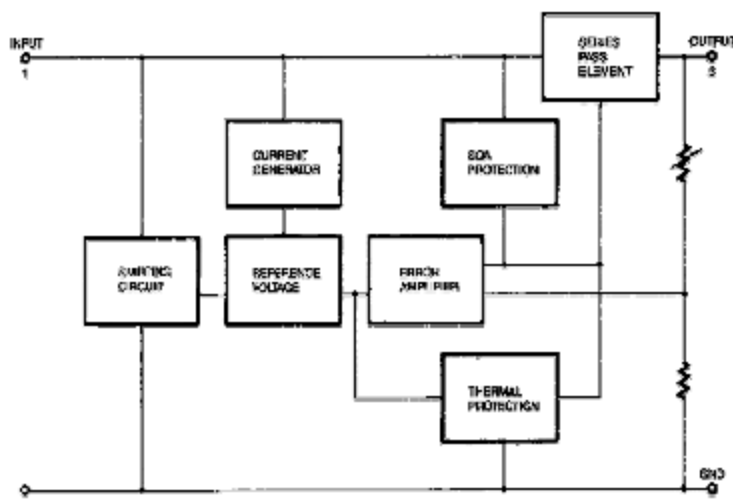


Package TO-220
(top view)



Package

Internal Block Diagram



Absolute Maximum Ratings

Maximum Input Voltage: 36V

Operation Junction Temperature Range: -40°C to 125 °

Electrical Characteristics

($V_{in}=10V$, $I_o=0.5A$, $T_j=25^\circ$, unless otherwise noted.)

Characteristic	Symbol	Min	Max	Unit
Output Voltage	V_o	-4.8	-5.2	Vdc
Line Regulation				
-7.0 Vdc $\leq V_i \leq$ -25 Vdc $I_o=0.1A$	Reg_{line}	-	50	mV
-7.0 Vdc $\leq V_i \leq$ -25 Vdc $I_o=0.5A$		-	100	
-8.0 Vdc $\leq V_i \leq$ -12 Vdc $I_o=0.1A$		-	25	
-8.0 Vdc $\leq V_i \leq$ -12 Vdc $I_o=0.5A$		-	50	
Load Regulation				
5.0 mA $\leq I_o \leq$ 1.5 A	Reg_{load}	-	100	mV
250 mA $\leq I_o \leq$ 750 mA		-	50	
Quiescent Current	I_Q	-	8.0	mA
Quiescent Current Change				
-7.0 Vdc $\leq V_i \leq$ -25 Vdc	ΔI_Q	-	1.3	mA
5.0 mA $\leq I_o \leq$ 1.0A		-	0.5	

Typical Applications Circuit

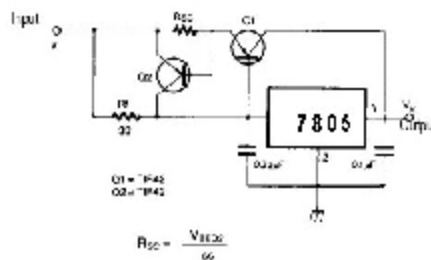
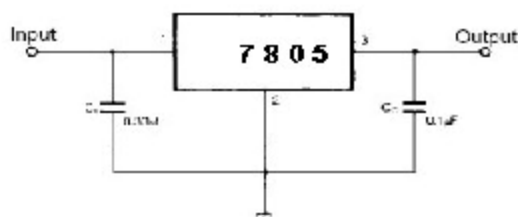
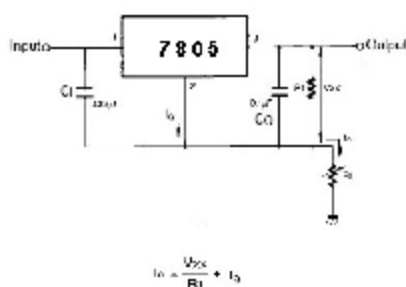


Figure 1. High Output Current with Short Circuit Protection


Figure 2. DC Parameters

Figure 3. Constant Current Regulator

Ordering Information

ORDERING NUMBER	PACKAGE	MARKING
7805	TO - 220	ESTE K

Address : 北京市海淀区永定路 88 号长银大厦 6A06-6A07

Rm 6A07, Changyin Office Building, No.88, Yong Ding Road, Hai Dian District, Beijing

Postalcode:100039

Tel: 86-010-58895780 / 81 / 82 / 83 / 84 Fax: 010-58895793

Http://www.estek.com.cn

Email:sales@estek.com.cn

REV No:01-060801

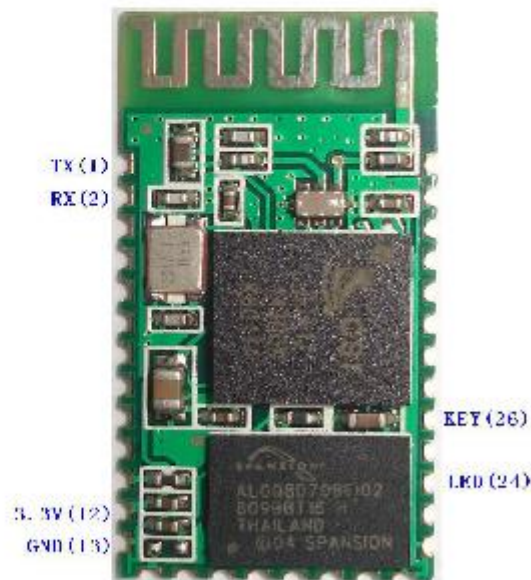
Bluetooth to serial HC-06 wireless module

Product Description:

- 1, Mainstream CSR Bluetooth chip, Bluetooth V2.0 protocol standards
- 2, serial module operating voltage 3.3V.
- 3, the baud rate for 1200, 2400, 4800, 9600, 19200, 38400 and users can be set
- 4, core module size: 28mm x 15 mm x 2.35mm.
- 5, the working current: 40mA
- 6, Sleep current: <1mA
- 7, for the GPS navigation system, utility meter reading system, the industrial field, collecting and controlling system.
- 8, with a Bluetooth laptop computer to the Bluetooth adapter, PDA and other devices to seamlessly connect

The module's host and slave, the host and slave pairing communication from the machine and from the machine or between the host and the host can not communicate, communication function and computers, mobile phones and other Bluetooth pairing purchase default slave , requires that the host needs to be indicated]

Main distinction: 1, if the chip is not specified on, the lights flash slow main fast from; September 2,2009, all manufactured host will be playing in the IC a hook or paste There are the "main" characters, there is no hook or not affixed to the word "master" is the slave. The date of manufacture can be obtained from the Bluetooth address]





The factory default parameters:

Slave, baud rate: 9600, n, 8,1. Passkey: 1234; need host mode, indicate when the orders.

Second, AT command set as follows:

1, test communications

Send: AT (return OK, one second left and right)

Back: OK

2, change the Bluetooth serial communication baud rate

Send: AT + BAUD1

Back to: OK1200

Send: AT + BAUD2

Back to: OK2400

```

.....
1 ----- 1200
2 ----- 2400
3 ----- 4800
4 ----- 9600
5 ----- 19200
6 ----- 38400
7 ----- 57600
8 ----- 115200
9 ----- 230400
A ----- 460800
B ----- 921600
C ----- 1382400
  
```

Not recommended to use more than 115200 baud rate, signal interference causes the system to instability.

The settings over 115,200 with a computer is not available, use microcontroller programming in higher than 115200 to use this baud rate and re-issue the AT command set low baud rate

AT command set baud rate, the next power do not need to set up and can be powered down to save the baud rate.

3, change the Bluetooth name

Send: AT + NAMEname

Back to: OKname

Parameter name: To set the current name, the name of the Bluetooth search. 20 characters or less.

Example: Sending AT + NAMEbill_gates

Back OKname

The Bluetooth name changed to bill_gates

The parameters can be powered down to save, simply modify the time. PDA the end refresh can see the Bluetooth name changed.

4, change the Bluetooth pairing password

Send: AT + PINxxxx

Returns: OKsetpin



Parameter xxxx: To set a passcode, 4 bytes, this command can be used from the machine or host. The slave adapter or mobile phone pops up to enter when pairing the password window, manually enter this parameter can be connected from the machine. Host in the main Bluetooth module connected digital camera, digital camera from the machine, find the camera pairing password, and then set up the White Bluetooth module, the main Bluetooth module can automatically connect the camera.

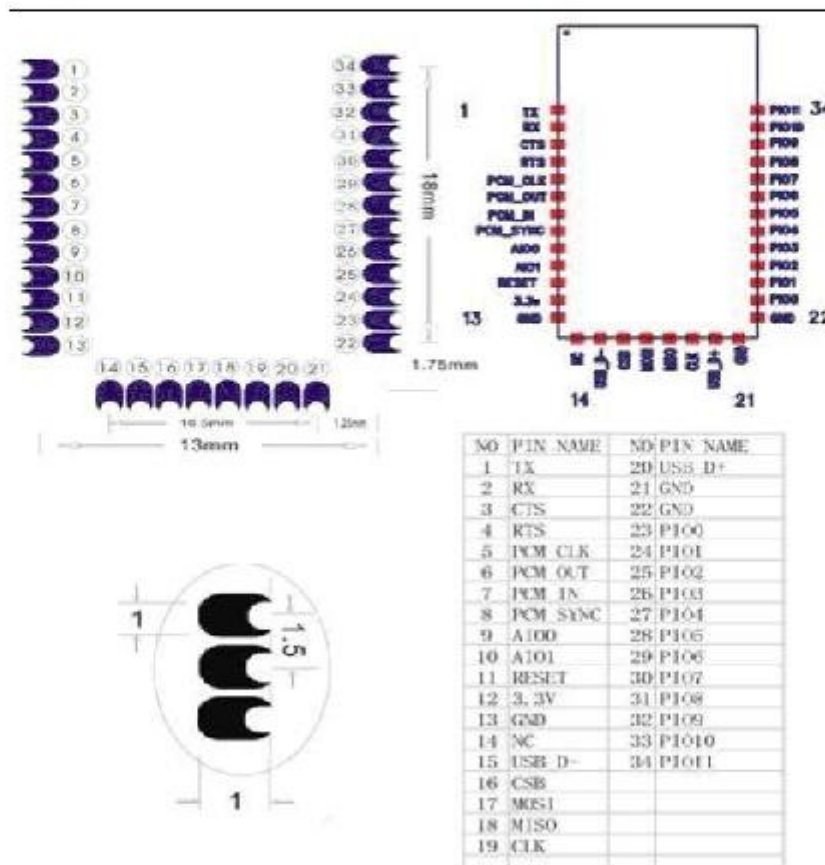
Example: Sending AT + PIN8888

Back OKsetpin

The Bluetooth pairing password to 8888, module paired at the factory default password is 1234.

The parameters can be powered down to save, simply modify the time.

Package pin diagram:

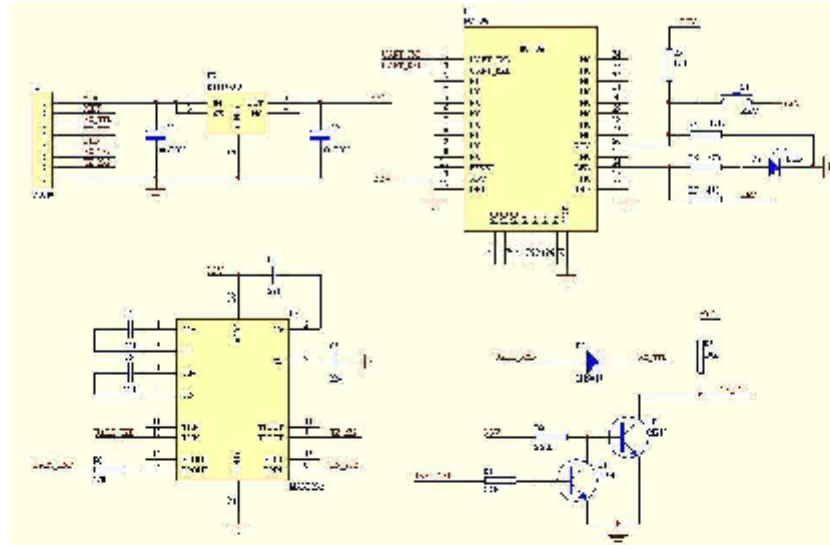


The old customer please refer to (schematics, wiring diagrams, etc.)

This module is for upgrade BCM_LV module version



The figure below shows the core module peripheral circuit schematic diagram of the typical



application

If you use the factory default baud rate, do not want to modify the contents of the baud rate below do not need to care about:

The AT command for the master and slave. When used in pairs, the master and slave can be different baud rate can also transfer data, but the host and the device connected to the host baud rate to be the same, but also slave and slave devices connected to the same baud rate .

Setting mode:

Initial communication parameters for 9600, N, 8,1, before pairing (pairing indicator flashes) Send to modify the baud rate command

Before sending AT commands to ensure that the hardware is connected as follows:

The eight-pin interface with base plate, the first pin external power supply (3.3 to 5V), the sixth leg connected to the computer COM1 port (DB9 male) pin, pin 7 computer COM1 port pin, the eighth pin is connected to the computer COM1 port 5 feet. Using HyperTerminal or serial debugging assistant to open the COM1 port of the computer, enter the text "AT" to manually send. Special note: master and slave pair can communicate between master and slave baud rate is required to be consistent between master and slave is to take the Bluetooth protocol, rather than serial port protocol.

1, test communications

Send: AT (return OK, one second left and right)

Back: OK

2, change the Bluetooth serial communication baud rate

Send: AT + BAUD1



Back to: OK1200

Send: AT + BAUD2

Back to: OK2400

.....

1 ----- 1200

2 ----- 2400

3 ----- 4800

4 ----- 9600

5 ----- 19200

6 ----- 38400

7 ----- 57600

8 ----- 115200

9 ----- 230400

A ----- 460800

B ----- 921600

C ----- 1382400

Not recommended to use more than 115200 baud rate, signal interference causes the system to instability.

The settings over 115,200 with a computer is not available, use microcontroller programming in higher than 115200 to use this baud rate and re-issue the AT command set low baud rate

AT command set baud rate, the next power do not need to set up and can be powered down to save the baud rate.

3, change the Bluetooth name (February 2008 after 24 new features)

Send: AT + NAMEname

Back to: OKname

Parameter name: To set the current name, the name of the Bluetooth search. 20 characters or less.

Example: Sending AT + NAMEbill_gates

Back OKname

The Bluetooth name changed to bill_gates

The parameters can be powered down to save, simply modify the time. PDA the end refresh can see the Bluetooth name changed.

4, change the Bluetooth pairing password

Send: AT + PINxxxx

Returns: OKsetpin

Parameter xxxx: To set a passcode, 4 bytes, this command can be used from the machine or host.

The slave adapter or mobile phone pops up to enter the pairing password window, then manually enter this parameter

The number can be connected to the slave. Host in the main Bluetooth module connected digital camera, digital camera from the machine, find the pairing password of the camera, and then set up the White Bluetooth module, the master Bluetooth module can be.

CRONOGRAMA GENERAL DE ACTIVIDADES

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Cronograma											
							tri 2, 2018	abr	may	jun	tri 3, 2018	jul	ago	sep				
1		ELABORACION DEL PLAN	10 días	sáb 28/4/18	jue 10/5/18													
2		ENTREGA DEL PLAN	5 días	mar 8/5/18	lun 14/5/18													
3		ELABORACION DE A PROPUESTA	5 días	jue 10/5/18	mié 16/5/18													
4		ENTREGA DE LA PROPUESTA	5 días	mar 15/5/18	lun 21/5/18													
5		DISEÑO DEL PROYECTO	7 días	mar 22/5/18	mié 30/5/18													
6		SIMULACION Y MONTAJE DEL PROYECTO	15 días	mar 5/6/18	lun 25/6/18													
7		PRUEBAS FINALES Y ELABORACION DEL INFORME TECNICO	30 días	mié 20/6/18	mar 31/7/18													
8		ELABORACION DE MANUALES TECNICOS DE USUARIO	7 días	lun 30/7/18	mar 7/8/18													
9		ELABORACION DE PRESENTACION DIGITAL	7 días	mié 8/8/18	jue 16/8/18													
10		ENTREGA DEL PROYECTO	5 días	vie 17/8/18	jue 23/8/18													

Proyecto: proyecto cris RFID1 Fecha: mar 25/9/18	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha límite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
Hito inactivo		solo fin				

MANUAL DEL USUARIO



1. ÍNDICE.

1.	ÍNDICE.....	1
2.	Contenido del producto	2
3.	Indicaciones de Seguridad.....	2
4.	PRECAUCIONES DE USO.....	3
4.1	Encendido del dispositivo.....	3
4.2	Impermeabilidad.....	3
4.3	Úselo adecuadamente.....	3
4.4	Acerca del dispositivo.....	3
5	INTRODUCCIÓN.....	4
5.1	Descripción de accesorios y partes.....	4
5.2	Vista frontal y lateral del dispositivo.....	4
5.3	Conector de alimentación	4
5.4	Software de instalación.....	5
6.	INSTALACIÓN DEL SOFTWARE.....	5
6.1	Requerimientos básicos.....	5
6.2	Instalación.....	5
7	GUÍA DE USO.....	6
7.1	Activación del dispositivo.....	6
7.2	Inicio de compras	8
8	SOLUCIÓN DE PROBLEMAS COMÚNES.....	11
9	EMAIL Y TELÉFONOS DE SOPORTE TÉCNICO.....	12

1. Contenido del producto

- 1 dispositivo de comunicación
- 1 batería recargable
- 1 aplicación Android
- 1 Aplicación Web
- 1 manual de usuario
- 1 manual técnico

2. Indicaciones de Seguridad.

- El dispositivo de conexión está diseñado bajo reglas técnicas. Se han realizado pruebas de verificación y ha salido de fábrica en buen estado.
- Mantenga el dispositivo alejado de dispositivos que generen interferencias eléctricas.
- Si el dispositivo no funciona adecuadamente desconecte la batería y comuníquese con el centro de soporte técnico autorizado.
- **IMPORTANTE** para evitar riesgo de descarga eléctrica, no exponer al dispositivo a la humedad.

3. PRECAUSIONES DE USO.

4.1 Encendido del dispositivo.

Oprima el botón que se indica en la figura, llevándolo de la posición cero “0” hacia la posición uno “1”, recuerde que este paso solamente inicia la alimentación del dispositivo.



Figura No. 1: Interruptor de Encendido/Apagado del dispositivo

4.2 Impermeabilidad.

No sumerja el dispositivo en cualquier líquido o fluido, esto puede causar daños en la estructura interna del dispositivo, ya que no es a prueba de agua.

De requerir limpieza, hágalo con un paño suave y humedecido.

4.3 Úselo adecuadamente.

No utilice el dispositivo sin antes leer y comprender el presente manual.

4.4 Acerca del dispositivo.

Recuerde el modo de funcionamiento:

Primero permite hacer una consulta del producto y precio y desde el teléfono celular por medio de la aplicación móvil previamente instalada en su teléfono puede ir agregando o borrando los productos a la lista de compras, una vez listo los productos a comprar mediante su aplicación puede generar su factura de compra.

Recuerde que los componentes del dispositivo pueden ser ingeridos por menores de 5 años, manténgalo fuera del alcance de estos.

5 INTRODUCCIÓN.

5.1 Descripción de accesorios y partes.

Accesorios:

- Batería de 9V recargable
- Cargador de 9 Voltios.
- Archivo con extensión .apk

Partes:

- Ranura de acceso para cambio de batería.
- Placa electrónica con 02 módulos.

5.2 Vista frontal y lateral del dispositivo.



Figura No. 2: Vista frontal y lateral del dispositivo

5.3 Conector de alimentación

Asegúrese de identificar y conectar de manera adecuada la polaridad de la batería al dispositivo como se indica en la siguiente figura.



Figura No. 3: Vista del conector de batería

5.4 Software de instalación.

El archivo con la extensión .apk le será instalado al momento de la adquisición del producto, al igual que le será enviado a su correo personal, como respaldo.

En caso de requerir nuevamente el archivo mencionado anteriormente, escríbanos a los correos o llámenos a los números que se indican al final del presente manual.

6. INSTALACIÓN DEL SOFTWARE.

6.1 Requerimientos básicos.

- Teléfono celular con sistema operativo ANDROID
- Teléfono con bluetooth incorporado.
- Teléfono celular que soporte aplicaciones móviles.
- Procesador de 1.2 GHz como mínimo.
- Más de 2.5M libres en memoria de almacenamiento.

6.2 Instalación.

- Abra la carpeta con el nombre Compras.
- Encontrará un archivo .apk, que debe ejecutarlo para que inicie el proceso de instalación.

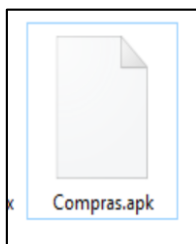


Figura No. 4: Archivo .apk para iniciar la configuración

- Presione el botón instalar como se muestra a continuación

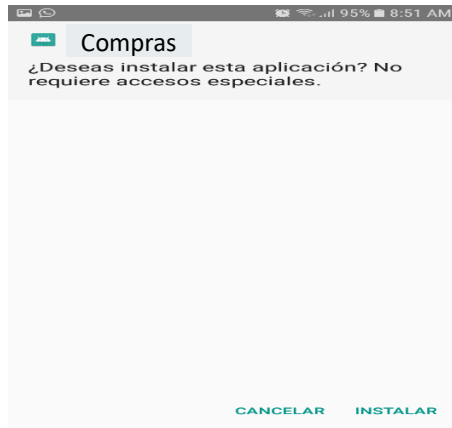


Figura No.5: Proceso de instalación de la aplicación móvil.

7 GUÍA DE USO.

Con todos los antecedentes descritos anteriormente, usted está en condiciones de operar de manera adecuada el sistema. Así que a continuación se le indicará paso a paso la manera correcta para usar el equipo.

7.1 Activación del dispositivo.

- Con el dispositivo colocado en el carro del supermercado, proceda a encenderlo.



Figura No. 6 Dispositivo en el carro de compras.

- Inmediatamente active el bluetooth de su teléfono móvil en donde se encuentra instalada la aplicación.

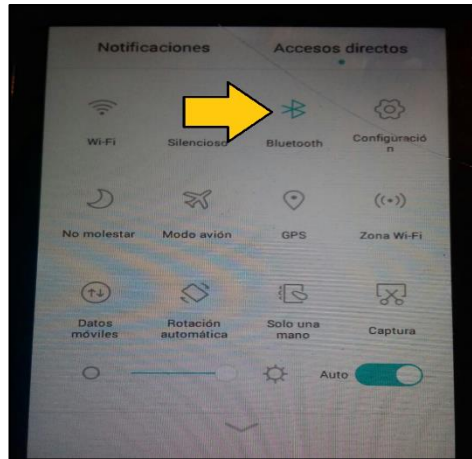


Figura No.7: Búsqueda de dispositivos bluetooth dentro de una zona.

- Busque los nombres de los dispositivos bluetooth cercanos a su móvil y seleccione aquel que dice ASISTENTE COMPRAS

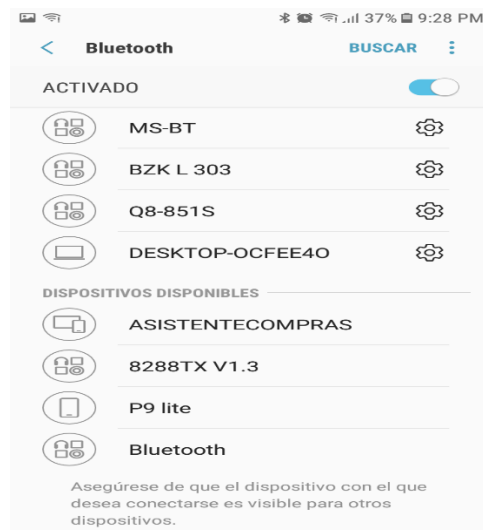


Figura No. 8: Emparejado con el HC-06

- Inmediatamente le pedirá una clave de vinculación, usted debe digitar 1234 para entablar un enlace entre su móvil y el dispositivo.
- Seguido de esto inicie la aplicación que previamente se instaló e inmediatamente observará la siguiente pantalla.

Tienda Virtual

Cédula

INGRESAR



Figura No.9: Pantalla principal de la aplicación móvil.

- Observará que la pantalla tiene un espacio destinado a ingresar el número de cedula o RUC del usuario Cédula .
- Además observará el botón **INGRESAR** , una vez que haya colocado el número de cedula del usuario presione el botón Ingresar y se le mostrara a una nueva pantalla.

7.2 Inicio de compras

- Una vez presionado **INGRESAR** en la pantalla anterior se desplegara la siguiente pantalla.

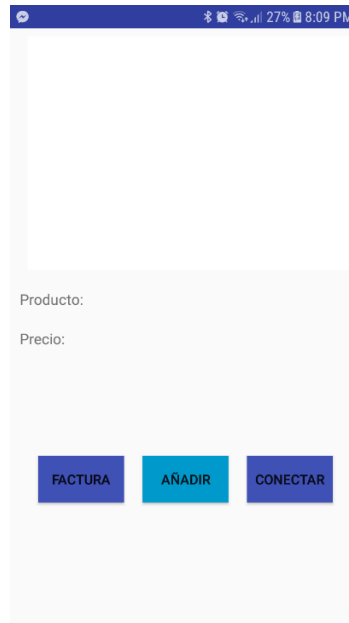



Figura No. 10. Pantalla de selección de la aplicación

- Una vez activa la pantalla presionamos  y el dispositivo se conectara con el móvil.
- Una vez conectado el dispositivo con el móvil procedemos a pasar el producto que contiene la tag por el dispositivo y se desplegara la siguiente pantalla que nos permite consultar el precio del producto como se muestra en la figura No. 11.

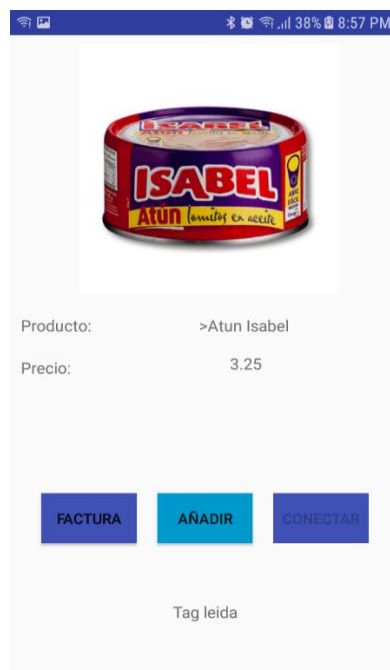


Figura No. 11: Consulta del precio del producto

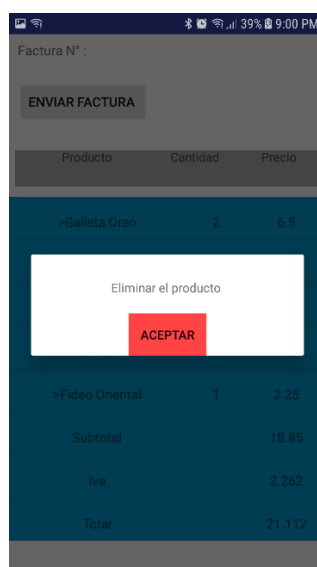
- Si deseamos comprar el producto presionamos **AÑADIR** y el producto se añadirá a la factura de compra como se muestra en la figura No.12.



Producto	Cantidad	Precio
>Atun Isabel	1	3.25
Subtotal		3.25
Iva		0.39
Total		3.6399999

Figura No.12. Pantalla de la factura

- Si desea comprar más de una unidad por producto siga pasando el producto por el dispositivo y añadiéndolo a la factura de compra.
- Si desea quitar un producto de su factura solamente presione sobre el nombre del producto que desea eliminar y se desplegara una pequeña pantalla que dice eliminar el producto, como se muestra en la figura No. 13



Producto	Cantidad	Precio
>Galleta Oreo	2	6.5
>Fideo Onemal	1	2.25
Subtotal		18.85
Iva		2.262
Total		21.112

Figura No. 13: Pantalla de eliminar un producto

- Una vez seleccionado el producto a ser eliminado presionamos **ACEPTAR** y el producto se borrara de la factura.
- Una vez agregado a la factura todos los productos que desee comprar presione **ENVIAR FACTURA** y se generará su factura, como se muestra en la siguiente figura No.14.

Factura N° : 41

ENVIAR FACTURA

Producto	Cantidad	Precio
>Galleta Oreo	2	6.5
>Cafe Nescafe	1	4.50
>Leche 1L	1	0.85
>Fideo Oriental	1	2.25
Subtotal		14.1
Iva		1.6920002
Total		15.792001

Figura No.14: Factura generada por el usuario.

NOTA: No olvide nunca, primero, encender el dispositivo y mantenerlo así para que interactúen tanto el celular y el artefacto.

8 SOLUCIÓN DE PROBLEMAS COMÚNES.

A continuación se le darán a conocer una serie de soluciones breves que le permitirán deducir si su dispositivo funciona de manera adecuada o es momento de cambiarlo o reclamar su garantía.

- Es muy importante que antes de iniciar la compra encienda la alimentación del dispositivo, pues sin éste paso no logrará ubicar su dispositivo, y aunque el bluetooth de su móvil esté encendido no le permitirá realizar la búsqueda
- Si su dispositivo no enciende, revise el estado de su batería y de ser el caso cárguela, revise que este colocada de la manera correcta.
- Si usted inicio su aplicación y no recibe respuesta en su celular, asegúrese de haber vinculado de antemano el bluetooth del dispositivo con su móvil, como se indicó en las figuras 7 y 8.

- No permita que los niños manipulen el dispositivo, podrían causar averías en el funcionamiento.

9 EMAIL Y TELÉFONOS DE SOPORTE TÉCNICO.

Para otras inquietudes, soluciones o consejos, contáctenos a los siguientes números telefónicos o escribanos a los correos que se detallan a continuación.

C&R Electronics

mcorella85@hotmail.com

2884097

0988469203

Corella Cristina

Ing. Electrónica.

MANUAL TÉCNICO



1. OBJETIVOS DEL MANUAL Y ALCANCE DEL SISTEMA

Éste manual tiene por objetivo informar las características técnicas, ya sean eléctricas o electrónicas del sistema, así como resolver problemas técnicos que se presenten en el equipo.

Las limitaciones que presenta el sistema, tiene que ver con respecto a la distancia, puesto que el dispositivo, tiene un alcance efectivo de 10m para el enlace con Smartphone, tanto en línea directa como en presencia de obstáculos.

Está diseñado específicamente para acoplarse a la base superior de un carro de compras de un supermercado. Adicional a esto es capaz de trabajar ininterrumpidamente, siempre y cuando la batería haya sido debidamente cargada.

2. DESCRIPCIÓN DE BASE DE DATOS.

a) Especificaciones Técnicas:

Sistema operativo móvil Android.- Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y BlackBerry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma. El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.)

- Código abierto.
- Núcleo basado en el Kernel de Linux.
- Adaptable a muchas pantallas y resoluciones.
- Utiliza SQLite (sistema de gestión de bases de datos relacional) para el almacenamiento de datos.
- Ofrece diferentes formas de mensajería.
- Navegador web basado en WebKit (es una plataforma para aplicaciones que funciona como base para el navegador web).
- Soporte de Java y muchos formatos multimedia.
- Soporte de HTML, HTML5, Adobe Flash Player, etc.
- Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del
- software.
- Catálogo de aplicaciones gratuitas o pagadas en el que pueden ser descargadas e instaladas (Google Play).
- Bluetooth.
- Google Talk desde su versión HoneyComb, para realizar video llamadas.

Modulo Lector RFID RC522.- Es un módulo lector y grabador de Tags RFID, compatible con casi cualquier microcontrolador, arduino o tarjeta de desarrollo. Utiliza también un sistema avanzado de modulación y demodulación para todo tipo de dispositivo pasivos de 13,56Mhz.

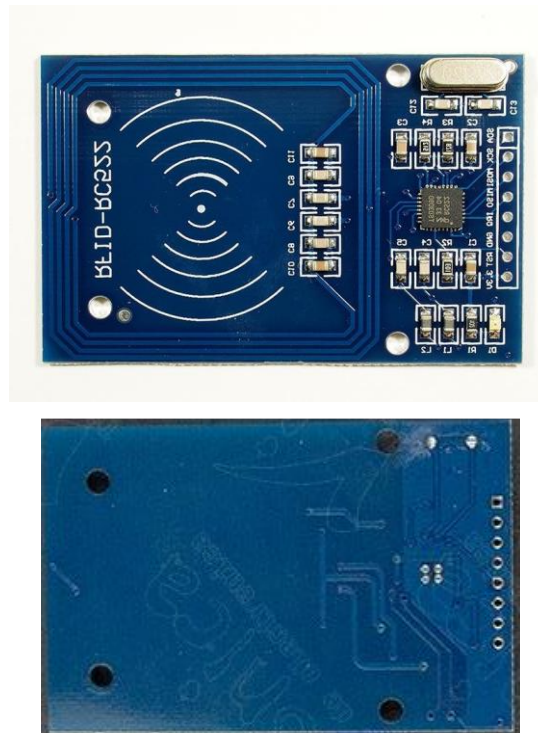


Figura No. 4: Vista superior e inferior del RFID RC522.

- Voltaje de operación: 3,3V
- Corriente de operación: 13mA~26mA
- Tensión de funcionamiento: 5V.
- Isb de stand by: 10-13mA a 3,3V
- Ism de sleep-mode: <80uA
- Im máxima: 30mA.
- Frecuencia de operación: 13,56MHz
- Distancia de lectura: 0 a 60mm.
- Protocolo de comunicación: SPI
- Velocidad de datos máxima: 10Mbit/s.
- Dimensiones: 40 x 60mm
- Máxima velocidad de SPI: 10 Mbit/s

MIFARE1 S50.- MIFARE es una tecnología que ha tenido un gran éxito en el mundo entero, se las conoce también por el nombre “tarjetas inteligentes” o “Tags”, son tarjetas de pvc que incluyen en su interior una antena que permite la comunicación de dicha tarjeta con un lector o receptor de radiofrecuencia (RFID) a una distancia que varía entre 2 cm y 10 cm, estas Tags trabajan en un rango de frecuencia desde 125 Khz a 13,56 Mhz.



Figura No. 02. Mifare1 S50

- Tamaño pequeño, robustez y durabilidad, no se descolora
- A prueba de agua, a prueba de golpes, anticorrosión
- Disponible en diferentes colores
- Disponible en diferentes diseños
- Disponible en LF 125Khz y HF 13.56MHz
- Disponible en diferentes chips (TK4100, ATA5567, MIFARE 1 S50, MIFARE 1 S70, Ultraligero, Desfire, I-CODE SLI)
- Color personalizado y marca de inyección de tinta para el número de identificación o logotipo disponible a petición

Módulo Bluetooth HC-06.- El módulo Bluetooth HC-06 viene configurado de fábrica para trabajar como esclavo. Ideal para aplicaciones inalámbricas, utiliza el protocolo UART RS 232 serial, de fácil, implementación con PC, microcontroladores y Arduino.

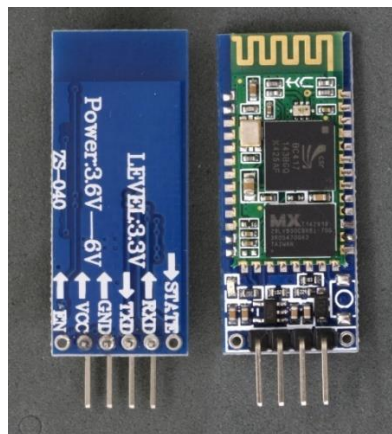


Figura No. 03: Vista frontal y posterior del HC-06.

- Compatible con el protocolo Bluetooth V2.0
- Voltaje de alimentación: 3.3VDC a 6VDC.
- Voltaje de operación: 3.3VDC.
- Baud rate ajustable: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
- Baud rate: 9600
- Tamaño: 1,73 in x 0,63 in x 0,28 in (4,4 cm x 1,6 cm x 0,7 cm)
- Corriente de operación: < 40 mA
- Corriente modo sleep: < 1mA.
- Frecuencia 2,4 GHz

Regulador 7805.- Dispositivo electrónico que tiene la capacidad de regular voltaje positivo de 5V a 1^a de corriente, muy utilizado con arduino y microcontroladores.

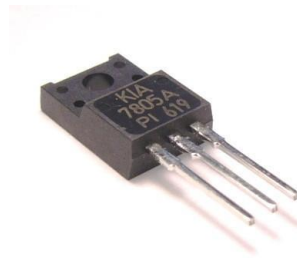


Figura No.04: Vista frontal del regulador 7805.

- Voltaje de salida: +5 V
- Corriente de salida max.: 1 A
- Protección contra sobrecarga térmica
- Limitación interna de corriente contra cortocircuitos
- Voltaje de entrada máximo: 35 V
- Tolerancia: 4%
- Caída de voltaje: 2 V típico
- Encapsulado TO-220

3. DIAGRAMAS ESQUEMÁTICOS.

A continuación se muestra el diagrama de los elementos que intervienen en la comunicación del dispositivo con el celular.

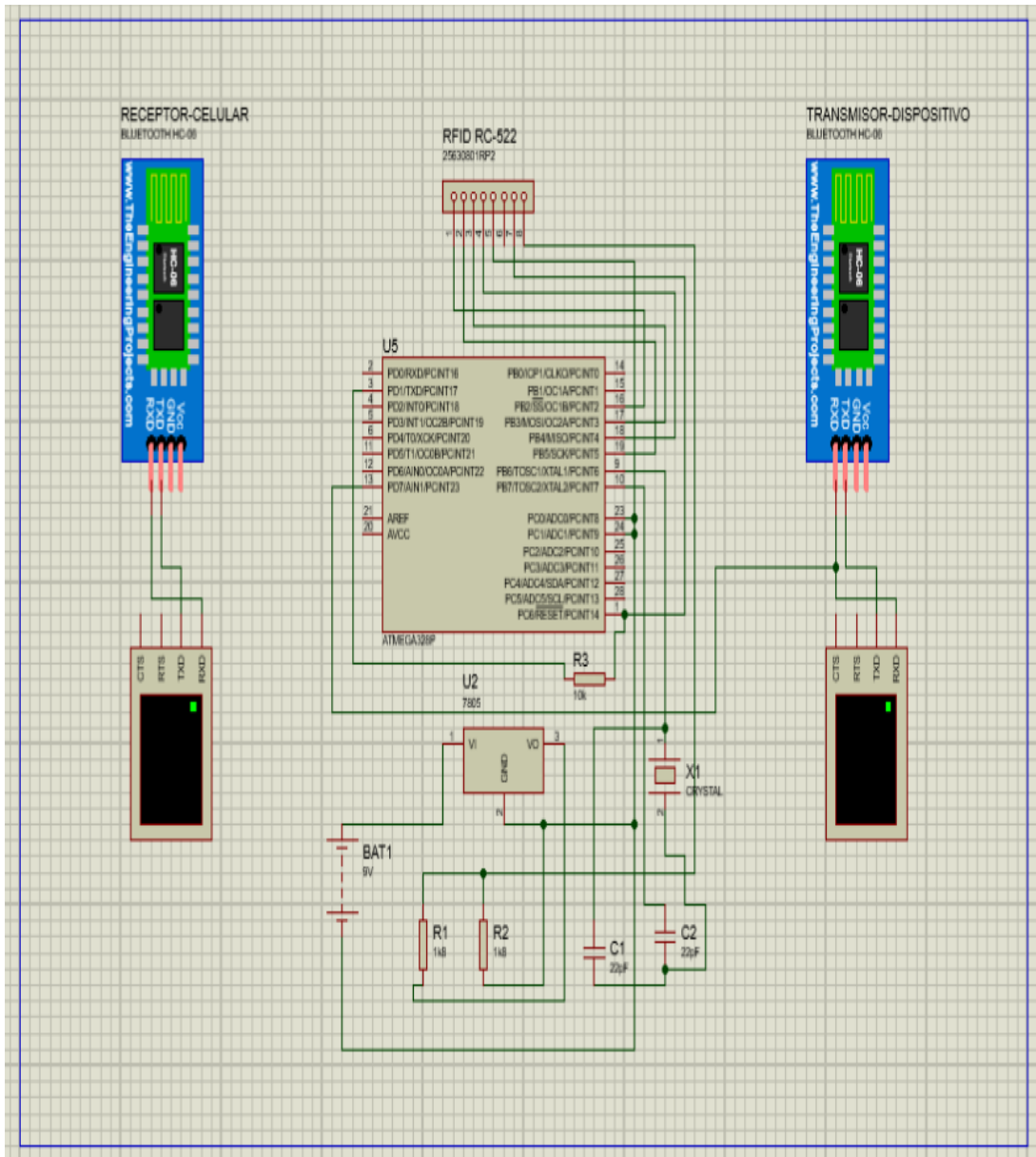


Figura No. 5: Esquemático para prueba de enlace. Bluetooth

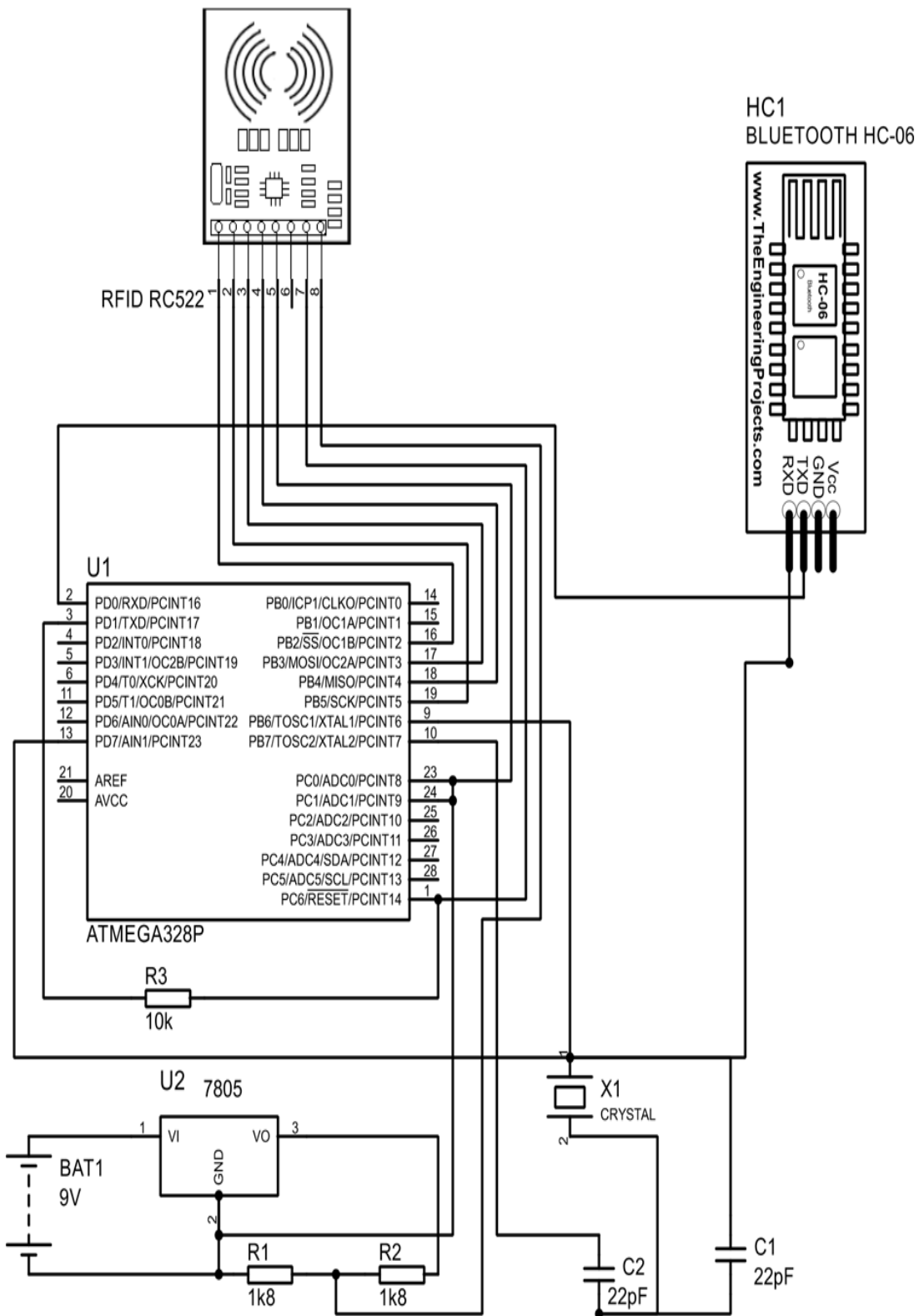


Figura No. 6: Diagrama electrónico para elaboración de la placa electrónica.

4. DIAGRAMAS PCB.

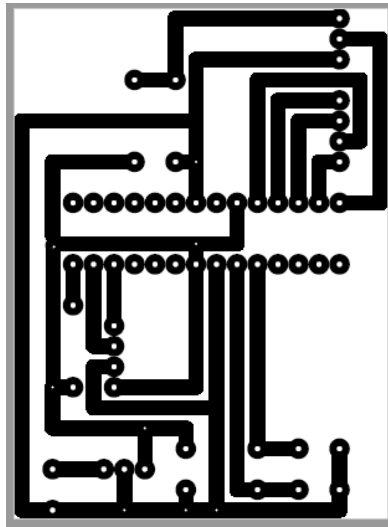


Figura No. 7: Diagrama a utilizarse en elaboración de la placa electrónica

5. GUÍA RÁPIDA DE MANTENIMIENTO.

El único elemento a retirar por parte del técnico es la batería de 9V, ya que tanto el microcontrolador ATmega 328P como el módulo HC-06, llevan un lenguaje de programación específico y si llegase a ser modificado, el sistema funcionará a medias o simplemente no funcionará.

Con respecto a la batería, vale recalcar que no se debe confundir los cables que entregan voltaje con aquellos que sirven de la alimentación del circuito.



Figura No. 8: Batería 9V para suministrar voltaje al sistema.

En caso de requerir mayor información o cambio de alguna de sus partes, revise el numeral 7 donde encontrará correos electrónicos y números telefónicos para soporte técnico especializado.

6. GUÍA RÁPIDA PARA SOLUCIÓN DE PROBLEMAS.

A continuación se le darán a conocer una serie de soluciones breves que le permitirán deducir si su dispositivo funciona de manera adecuada o es momento de cambiarlo o reclamar su garantía.

- Es muy importante que antes de iniciar el proceso active el botón de alimentación del dispositivo.
- Si su dispositivo no enciende, revise el estado de su batería y de ser el caso cárguela y colóquela de manera correcta.
- Si usted inicio su aplicación y no recibe respuesta en su celular, asegúrese de haber vinculado de antemano el bluetooth del dispositivo con su móvil., si aun habiendo realizado esta verificación no hay conexión bluetooth revise el bluetooth del dispositivo y de ser el caso cámbielo.
- No permita que niños manipules el dispositivo, ya que podrían causar averías en el funcionamiento.

7. EMAIL Y TELÉFONOS DE SOPORTE TÉCNICO.

Para otras inquietudes, soluciones o consejos, que no sean proporcionados en el presente manual contáctenos a las siguientes direcciones:

C&R Electronics

mcorella85@hotmail.com

2884097

0988469203

Corella Cristina