



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y
CONTROL**

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

AUTORES:

**GUARNIZO POMA, JOSÉ JAVIER
VALLEJO MANCERO, BERNARDO JAVIER**

**TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
FACTURACIÓN EN PISTA PARA ESTACIONES DE SERVICIO**

**DIECTOR: ING. ALULEMA, DARWIN
CODIRECTOR: ING. GRANIZO, EVELIO**

SANGOLQUÍ, JULIO 2014

Certificado de Tutoría

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICADO

Ing. Darwin Alulema

Ing. Evelio Granizo

CERTIFICAN

Que el trabajo titulado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE FACTURACIÓN EN PISTA PARA ESTACIONES DE SERVICIO”, realizado por José Javier Guarnizo Poma y Bernardo Javier Vallejo Mancero, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la institución, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas - ESPE

Debido a que se trata de un trabajo de investigación recomiendan su publicación. El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a José Javier Guarnizo Poma y Bernardo Javier Vallejo Mancero que lo entreguen al Ingeniero Luis Orozco, en su calidad de Coordinador de la Carrera.

Sangolquí, 3 de julio de 2014

Ing. Darwin Alulema
DIRECTOR

Ing. Evelio Granizo
CODIRECTOR

Declaración de Responsabilidad

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

DECLARACIÓN DE RESPONSABILIDAD

JOSÉ JAVIER GUARNIZO POMA

BERNARDO JAVIER VALLEJO MANCERO

DECLARAMOS QUE:

El proyecto de grado denominado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE FACTURACIÓN EN PISTA PARA ESTACIONES DE SERVICIO”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie, de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 3 de julio de 2014

José Javier Guarnizo Poma

Bernardo Javier Vallejo Mancero

Autorización de Publicación

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Nosotros, José Javier Guarnizo Poma y Bernardo Javier Vallejo Mancero

Autorizamos a la Universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la Institución del trabajo “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE FACTURACIÓN EN PISTA PARA ESTACIONES DE SERVICIO”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, 3 de julio de 2014

José Javier Guarnizo Poma

Bernardo Javier Vallejo Mancero

DEDICATORIA

A mis padres Francisco y Fanny por ser siempre mi gran ejemplo a seguir.

José Guarnizo

A mis padres, por todo su esfuerzo y dedicación; a mis hermanos, por ser con quienes siempre puedo contar; y finalmente a todas las personas importantes en vida, en especial a aquella que estuvo a mi lado en el reto más difícil que se me ha presentado.

Bernardo Vallejo

AGRADECIMIENTO

A mis padres por su amor, apoyo y comprensión y sobre todo por su paciencia en esta etapa de mi vida. Un agradecimiento especial para mis amigos Paul y Henry por su ayuda en el proyecto. Agradezco también a mi director y codirector por su guía y consejos para el desarrollo del proyecto.

José Guarnizo

En primer lugar, a mis padres y hermanos por su apoyo durante cada etapa de mi vida, siendo el pilar fundamental para superar cada nuevo reto.

A mis docentes con mención especial a mi director y codirector, por brindarme parte de su conocimiento y valores, necesarios para cumplir este proyecto y dar comienzo a la vida profesional.

Y por último, a cada uno de mis compañeros y amigos, por las buenas experiencias que hemos compartido.

Bernardo Vallejo

ÍNDICE DE CONTENIDOS

DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDOS.....	vi
INDICE DE FIGURAS	xi
ÍNDICE DE TABLAS	xii
RESUMEN.....	xiii
CAPÍTULO 1: DESCRIPCIÓN DEL PROYECTO.....	1
1.1 ANTECEDENTES.....	1
1.2 JUSTIFICACIÓN E IMPORTANCIA	2
1.3 ALCANCE.....	3
1.4 OBJETIVOS.....	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos.....	4
CAPÍTULO 2: MARCO TEÓRICO.....	5
2.1 INTERFACES DE ENTRADA Y SALIDA	5
2.1.1 Teclados.....	5
2.1.2 Pantalla GLCD.....	7
2.1.3 Impresora Matricial	7
2.2 MÓDULOS RFID, ARDUINO Y MICROCONTROLADORES DE GAMA ALTA ..	8
2.2.1 Módulos RFID.....	8
2.2.2 Tecnología Arduino.....	9
2.2.3 Microcontroladores PIC de gama alta	12
2.3 DISEÑO DE PCB	13
2.3.1 Consideraciones Preliminares	13
2.3.2 Materiales	13
2.3.3 Consideraciones Eléctricas.....	14
2.4 WEB SERVICES.....	15
2.4.1 Definición.....	15
2.4.2 Consumir Web Services	16
CAPÍTULO 3: DESARROLLO DEL DISPOSITIVO.....	17
3.1 DISEÑO DE HARDWARE	17

3.1.1 Diagrama de Bloques del Hardware del Sistema	18
3.1.2 Componentes del Hardware	18
3.1.3 Diseño de la PCB (Printed Circuit Board).....	29
3.2 DISEÑO DE SOFTWARE	35
3.2.1 Diagrama de flujo Principal del microcontrolador	36
3.2.2 Diagramas de flujo del Programa Principal	37
3.2.3 Subrutinas principales.....	45
3.2.4 Método de validación de número de cédula y RUC.	51
3.3 DISEÑO DE LA CARCASA	52
3.3.1 Diseño de la base de la carcasa	53
3.3.2 Diseño de la tapa de la carcasa	54
3.3.3 Ensamblaje final y fabricación.....	56
3.4 COSTO DE FABRICACIÓN	56
CAPÍTULO 4: MÓDULO DE COMUNICACIÓN	58
4.1 TOPOLOGÍA DE RED.....	58
4.2 DESCRIPCIÓN DEL HARDWARE	60
4.2.1 Diagrama de Bloques	60
4.2.2 Componentes de Hardware	61
4.3 TRAMAS DE COMUNICACIÓN	62
4.3.1 Tramas del dispositivo facturador al servidor	63
4.3.2 Tramas del servidor al dispositivo facturador	64
4.4 DISEÑO DEL SOFTWARE	65
4.4.1 Diagrama de Flujo del Programa	65
4.4.2 Diagrama de Flujo de Interrupción de Puerto Serie	67
4.5 COSTO DE FABRICACIÓN	68
CAPÍTULO 5: DISEÑO DE SOFTWARE Y BASE DE DATOS	69
5.1 DISEÑO DE SOFTWARE	69
5.1.1 Consideraciones Preliminares	69
5.1.2 Arquitectura de la aplicación Cliente-Servidor.....	72
5.1.3 Distribución de pantallas de la aplicación Cliente-Servidor	73
5.1.4 Diagramas de Flujo.....	75
5.1.5 Diagramas de Clases.....	80
5.2 DISEÑO DE LA BASE DE DATOS	82
5.2.1 Diagrama de Casos de Uso.....	82

5.2.2 Diagrama de Clases	83
5.2.3 Modelo Lógico y Físico de la Base de Datos	85
5.2.4 Diccionario de la Base de Datos	87
5.3 REQUISITOS PARA LA INSTALACIÓN DE LAS APLICACIONES.....	88
5.3.1 Aplicación Administrador de Datos	88
5.3.2 Aplicación Cliente-Servidor	89
5.4 COMPARACIÓN DE FUNCIONES DEL SISTEMA DE FACTURACIÓN DESARROLLADO CON SISTEMAS COMERCIALES.....	89
CAPÍTULO 6: PRUEBAS Y ANÁLISIS DE RESULTADOS.....	91
6.1 CONECTIVIDAD DE LOS MÓDULOS DE COMUNICACIÓN A LA RED	92
6.2 IMPRESIÓN DE FACTURAS VÁLIDAS.	93
6.3 MÓDULO RFID.....	95
6.4 TIEMPO DE FACTURACIÓN.....	97
CAPÍTULO 7: CONCLUSIONES Y RECOMENDACIONES.....	99
7.1 CONCLUSIONES	99
7.2 RECOMENDACIONES.....	101
BIBLIOGRAFÍA.....	103

INDICE DE FIGURAS

Figura 1. Diagrama eléctrico de un teclado matricial de 4x4.	5
Figura 2. Estructura de un GLCD (Ibrahim, 2008)	7
Figura 3. Distribución del circuito impreso (IPC, 1998).	14
Figura 4. Relación entre ancho de pista, corriente y temperatura	15
Figura 5. Diagrama de Bloques del Hardware del Sistema	18
Figura 6. PIC 18F67J11 (Microchip, 2009)	19
Figura 7. Esquema del Banco de Capacitores.....	20
Figura 8. Esquema de conexiones Iniciales del Microcontrolador	21
Figura 9. Teclado Matricial vista Frontal	21
Figura 10. Teclado Matricial diagrama de conexiones.....	23
Figura 11. Diagrama de conexión del conector macho del teclado al microcontrolador.	23
Figura 12. LCD Grafico 128x64 pixeles (Vishay, 2002)	25
Figura 13. Diagrama de conexión del GLCD al Microcontrolador.....	26
Figura 14. Módulo RFID ID-12LA (id-innovations, 2013)	27
Figura 15. Diagrama de conexión del módulo RFID	27
Figura 16. Cargador Universal de 5V y 2A (Importec, 2013)	28
Figura 17. Diagrama de conexión de Regulador de Voltaje.....	29
Figura 18. Diagrama del Circuito Electrónico.....	32
Figura 19. Circuito Impreso Capa Superior.....	33
Figura 20. Circuito Impreso Capa Inferior.	34
Figura 21. Circuito Impreso en 3 dimensiones.....	34
Figura 22. Circuito Impreso Placa de Alimentación y Comunicación	35
Figura 23. Diagrama de Flujo del Programa Principal.	36
Figura 24. Diagrama de Flujo rutina Inicialización	37
Figura 25. Diagrama de Flujo Inicio De Sesión.....	38
Figura 26. Diagrama de Flujo Menú.....	40
Figura 27. Diagrama de flujo selección de Cliente.....	41
Figura 28. Diagrama de flujo búsqueda y datos del Cliente.....	42
Figura 29. Diagrama de flujo proceso de venta	43

Figura 30. Diagrama de flujo forma de pago cliente Normal.....	44
Figura 31. Subrutina interrupción serial RFID.....	46
Figura 32. Subrutina interrupción serial del Servidor.	47
Figura 33. Subrutina del Teclado.....	48
Figura 34. Diagrama de flujo de la subrutina de gráficos en el GLCD.	50
Figura 35. Diagrama de Flujo – Validación de Cédula y RUC.	52
Figura 36. Base de la Carcasa.....	53
Figura 37. Dimensiones de la Base de la Carcasa.	54
Figura 38. Tapa de la Carcasa.....	55
Figura 39. Dimensiones de la Tapa de la Carcasa	55
Figura 40. Ensamblaje final	56
Figura 41. Topología de red del sistema de facturación en pista.....	59
Figura 42. Diagrama de Bloques del Hardware del Módulo de Comunicación.	60
Figura 43. Tarjeta Arduino Mega 2560 (Arduino, 2014).....	61
Figura 44. Tarjeta Arduino Ethernet Shield (Arduino, 2014).	62
Figura 45. Estructura de una trama	63
Figura 46. Diagrama de Flujo del Programa en Arduino.....	66
Figura 47. Diagrama de Flujo de la Interrupción del Puerto Serial.....	67
Figura 48. Aplicaciones del software	70
Figura 49. Arquitectura de la aplicación Administración.	72
Figura 50. Esquema 1.....	73
Figura 51. Esquema 2.....	74
Figura 52. Esquema 3.....	75
Figura 53. Esquema 4.....	75
Figura 54. Diagrama de Flujo de la aplicación Administrador de Datos.....	76
Figura 55. Consumo de Web Service	77
Figura 56. Aplicación PHP	77
Figura 57. Diagrama de Flujo de la aplicación Cliente-Servidor	79
Figura 58. Diagrama de clases de la aplicación Administrador de Datos. ...	81
Figura 59. Diagrama de clases de la aplicación Administración.	82
Figura 60. Diagrama de Casos de Uso.....	83
Figura 62. Modelo Lógico de la Base de Datos	86

Figura 63. Modelo Físico de la Base de Datos	87
Figura 64. Prueba de estado de comunicación módulo 1	92
Figura 65. Prueba de estado de comunicación módulo 2	93
Figura 66. Prueba de estado de comunicación módulo 3	93
Figura 67. Facturación de un día promedio de trabajo	94
Figura 68. Resultado prueba de reconocimiento de los dispositivos.	96
Figura 69. Comparación de tiempo de facturación Ingreso manual vs. Sistema de facturación instalado.	97

ÍNDICE DE TABLAS

Tabla 1. Tipos de etiquetas RFID	8
Tabla 2. Comparación entre tarjetas Arduino.	10
Tabla 3. Tipos de datos soportados en Arduino.....	12
Tabla 4. División de teclas por categorías del teclado matricial.....	22
Tabla 5. Conexión pines del teclado matricial al microcontrolador.	24
Tabla 6. Conexión pines de GLCD a microcontrolador.....	26
Tabla 7. Conexión pines módulo RFID	28
Tabla 8. Temperaturas máximas y mínimas por regiones del Ecuador	30
Tabla 9. Consumo de Corriente del dispositivo	30
Tabla 10. Parámetros para el cálculo del ancho de pista	31
Tabla 11. Costo de fabricación de dispositivo facturador.....	57
Tabla 12. Tabla de tramas enviadas del dispositivo facturador al servidor	64
Tabla 13. Tramas enviadas desde el servidor al dispositivo facturador.....	65
Tabla 14. Costo de fabricación módulo de comunicación.....	68
Tabla 15. Tareas y Acciones realizadas por el software.....	71
Tabla 16. Comparación sistema desarrollado con otras marcas comerciales.....	90
Tabla 17. Resultados prueba de funcionamiento módulo RFID.....	96

RESUMEN

El presente trabajo recapitula el desarrollo del diseño e implementación de un sistema de facturación en pista para estaciones de servicio. El proyecto se dio debido a la necesidad de las estaciones de servicio de eliminar la figura de consumidor final en sus ventas, de acuerdo a lo establecido por la resolución 00382 del SRI, y garantizar la emisión de facturas correctas con el fin de evitar multas a futuro por el SRI. Para lograr lo anterior, se diseñó y fabricó un dispositivo facturador que consta de un teclado alfanumérico, un LCD gráfico de 128x64 píxeles y una tarjeta de control; este dispositivo permite el ingreso de los datos del cliente y se encuentra ubicado en las pistas de la estación de servicio. Los dispositivos facturadores se encuentran conectados sobre una red Ethernet con topología tipo estrella, gracias a la plataforma Arduino, donde el nodo central de la red es un servidor en el cual se encuentra la base de datos que almacena todas las ventas, clientes y empleados de la estación de servicio. En el servidor se implementa un software en lenguaje JAVA que permite la interacción entre los dispositivos facturadores con la base de datos y las impresoras, ubicadas también en pista; además, el software permite realizar las operaciones de búsqueda, modificación, ingreso y eliminación sobre la base de datos. Finalmente, se obtuvo un sistema que garantiza la legibilidad de la información, acelera el proceso de facturación y almacena de forma estructurada la información de la estación.

PALABRAS CLAVE: SISTEMA DE FACTURACIÓN, BASE DE DATOS, JAVA, MICROCONTROLADOR, RFID, PCB, ARDUINO.

ABSTRACT

This paper summarizes the development of the design and implementation of a billing system on track for service stations. The project was given because of the need to service stations to eliminate the position of end consumers in their sales, as established by resolution 00382 of SRI, and ensure the issue of correct bills in order to avoid future fines by the SRI. To achieve this, we designed and manufacture a billing device consisting of an alphanumeric keypad, a graphic LCD 128x64 pixels and a control card; this device allows the entry of customer details and is located on the tracks of the service station. Billers devices are connected over an Ethernet network with star topology type, thanks to the Arduino platform, where the central node of the network is a server on which the database that stores all sales, customers and employees of the service station is. On the server a software in Java language is implemented that allows interaction between billers devices database and printers, also located on track; in addition, the software allows searching, modification, and disposal income on database. Finally, a system that ensures the legibility of the information was obtained, thus, the system accelerates the billing process and store information of the station in a structured way.

KEYWORDS: BILLING SYSTEM, DATABASE, JAVA, MICROCONTROLLER, RFID, PCB ARDUINO.

CAPÍTULO 1

DESCRIPCIÓN DEL PROYECTO

1.1 Antecedentes

Teleautomatismos V&P S.A., empresa ecuatoriana dedicada en la rama de Ingeniería Electrónica, vio la necesidad de crear un sistema de facturación en pista para las estaciones de servicio, que permita asegurar el correcto ingreso de datos tales como, número de cédula o RUC necesarios para la elaboración de facturas.

Hasta a mediados del año 2013 no era obligación por parte de las estaciones de servicio entregar una factura al consumidor a menos que el propio consumidor la pidiera. Sin embargo, a partir del 19 de agosto del 2013 todas tienen la obligación de emitir facturas debido a la resolución 00382 del SRI (ver *Anexo I*) que dice en su artículo 1:

“Disponer que en las facturas que se emitan con ocasión de transferencias de combustibles líquidos derivados de hidrocarburos (CLDH) y biocombustibles, se incorpore como requisito de llenado, adicionalmente a los establecidos en el Reglamento de Comprobantes de Venta, Retención y Documentos Complementarios, el número de cédula de identidad o ciudadanía del adquirente” (Maldonado, 2013).

En la actualidad, algunas empresas ecuatorianas ya ofrecen la solución para la problemática descrita anteriormente. Sin embargo, su solución es poco práctica porque los equipos utilizados están sobredimensionados, como es el caso de colocar computadoras en cada isla de la estación; otras empresas brindan una solución práctica pero su costo es muy elevado.

1.2 Justificación e Importancia

“Existen más de 1000 estaciones de servicio en el país” (Maldonado, 2013) y la mayoría realiza la facturación a mano. Esto se explica porque hasta el año pasado, no era una obligación emitir una factura para cada cliente y eran muy pocos los que solicitaban una.

En una estación de servicio promedio se realizan aproximadamente quinientas ventas al día, y el realizar la factura a mano ocasiona demoras en la atención al cliente o errores de escritura. Esto es un problema puesto que una factura con datos erróneos provoca una multa por parte del SRI y frente a una ganancia de centavos por galón vendido, cualquier multa presenta pérdida para la estación.

Además se ha visto que varias estaciones, en especial las pequeñas que solo tienen un empleado en pista, han tenido que contratar un mayor número de personal. Lo han hecho porque el facturar a mano genera demoras, provocando colas de automóviles en espera en las estaciones de servicio; y muchos de sus clientes deciden no esperar y se marchan, lo cual repercute en sus ganancias ya que el número de clientes que atienden diariamente disminuye.

Con este proyecto se garantiza la legitimidad de los datos que se ingresan, validando con sus respectivos algoritmos números de cédula y/o RUC. Además al ser un sistema digital no se cometen errores de lectura cuando se revisan las facturas, algo que si pasa cuando se factura a mano porque a veces no se entiende la letra de otra persona. Por otra parte, al ir almacenando clientes en una base de datos el proceso de facturación se vuelve más eficiente, debido a que no es necesario volver a ingresar los datos completos del cliente cada vez que va a la estación.

Además, dentro del software que se desarrolló en este proyecto se brindan aplicaciones útiles para una estación de servicio. Lo que le permitirá a la vez, no sólo facturar de una manera más eficiente sino también, llevar un mejor control administrativo de la estación.

1.3 Alcance

Por lo antes expuesto se diseñó un sistema de facturación que en general, facilite al operario en pista el ingreso de datos necesarios para emitir una factura pre impresa y se almacene en una base de datos estructurada. La impresión la realiza una impresora matricial ubicada en cada isla.

Para la facturación en pista el sistema cuenta con un dispositivo ubicado en cada dispensador de combustible. Tal dispositivo interactúa con el operario a través de un teclado alfanumérico y una pantalla LCD, además de tarjetas RFID, que se otorgará a clientes asiduos. El dispositivo cuenta con una serie de distintos menús que son fáciles de entender para el operario, y otorgan la mayor cantidad de datos tanto para facturar, como para llevar un control en la base de datos de todas las ventas y transacciones que ocurren en la estación de servicio.

Con el fin de abaratar costos y realizar una aplicación especializada para estaciones de servicio para el dispositivo que se describe, se diseñó e implementó la tarjeta electrónica de control y la carcasa. El teclado alfanumérico fue provisto por la empresa Teleautomatismos V&P.

La comunicación de todo el sistema se la realizó sobre una red TCP/IP Ethernet y su gestión la realiza un software instalado en el servidor. Los dispositivos se conectan vía serial a la tarjeta Arduino mega la que a su vez se conecta a la red a través de una tarjeta Arduino Ethernet Shield. Las impresoras se encuentran en la misma red y son controladas desde el software en el servidor permitiendo la impresión en pista.

El software de gestión se desarrolló en software open-source y es el encargado de interactuar con la base de datos que está en el mismo servidor. La base de datos se diseñó en base a las necesidades de las estaciones de servicio. Además de eso, el software cumple con funciones administrativas como: control de personal, administración de clientes, elaboración de reportes y cierre de caja.

1.4 Objetivos

1.4.1 Objetivo General

- Diseñar e Implementar un sistema de facturación en pista para estaciones de servicio.

1.4.2 Objetivos Específicos

- Realizar un estudio sobre la tecnología Arduino, RFID y microcontroladores de gama media-alta.
- Diseñar una tarjeta electrónica que permita la conexión de un teclado matricial, un GLCD y posea puertos seriales para módulo RFID y comunicación.
- Desarrollar un programa para el microcontrolador de la tarjeta electrónica que realice el manejo del GLCD a través del teclado matricial, e interactúe con la tarjeta Arduino y el módulo RFID a través de comunicación serial.
- Desarrollar una base de datos para el almacenamiento de las facturas así como de clientes, productos, precios y empleados.
- Desarrollar un software que permita la gestión de comunicaciones entre los equipos diseñados e impresoras en pista y la base de datos.
- Desarrollar un software que permita funciones administrativas como: control de personal, administración de clientes y cierres de caja.

CAPÍTULO 2

MARCO TEÓRICO

2.1 INTERFACES DE ENTRADA Y SALIDA

2.1.1 Teclados

2.1.1.1 Teclado Matricial

“Un teclado matricial está compuesto por teclas interconectadas formando una matriz. Las teclas son simples interruptores mecánicos y cada una ocupa la intersección de una fila con una columna” (Valdés Pérez & Pallàs Areny, 2007).

En la *Figura 1* se muestra la conexión eléctrica de un teclado matricial típico de 4 filas x 4 columnas. Para decodificar cuál tecla ha sido pulsada se lee las columnas del teclado o líneas de retorno que se mantienen en un nivel lógico ‘1’ o 5 voltios debido a las resistencias pull-up, mientras que en las filas o líneas de exploración se va colocando un nivel de ‘0’ lógico o 0 voltios sucesivamente. De esta forma, sólo si la tecla ha sido pulsada existirá en las líneas de retorno un ‘0’ lógico.

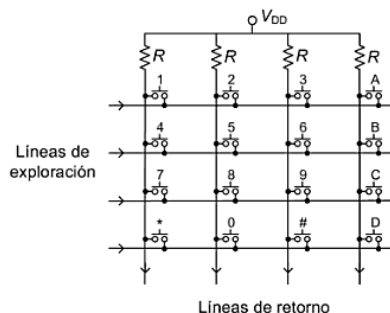


Figura 1. Diagrama eléctrico de un teclado matricial de 4x4. (Valdés Pérez & Pallàs Areny, 2007)

2.1.1.2 Teclados de Membrana.

Entre los tipos de teclados matriciales más usados en el mercado destacan los mecánicos, los cuales consisten en un resorte que devuelve la tecla a su posición original después de ser presionada; sin embargo, existen otros tipos como lo son los teclados de membrana, siendo sus principales ventajas:

- Facilidad para combinar estéticamente la presentación gráfica.
- Posibilidad de uso en ambientes agresivos.
- Gráfica inalterable, pues se imprime en el reverso del material del frente.
- Ahorro de espacio ya que el teclado está contenido en una estructura de apenas 1mm de espesor.
- Facilidad de montaje.

Para su fabricación se deben tomar en cuenta ciertos criterios, tales como:

- Sensación Táctil.- Se refiere al tipo de relieve que las teclas tendrán, existiendo tres tipos: domo metálico, burbuja y sin sensación táctil. En la primera se logra el relieve insertando una cúpula de metal mientras que en la de burbuja se inserta aire frío logrando así una deformación, cabe destacar que la de domo metálico es la que mayor robustez presenta.
- Materiales del frontal.- Se refiere al tipo de material que tendrá la lámina frontal, siendo los más usados policarbonato y poliéster. El poliéster es sumamente resistente a la fatiga y a productos químicos y presenta una mayor robustez frente al policarbonato.
- Conexionado.- Se refiere al diagrama eléctrico del teclado.
- Blindajes.- Se refiere a agregar o no una placa electrostática para minimizar posibles interferencias o ruido electrónico (GUEMISA, 2014).

2.1.2 Pantalla GLCD

Una pantalla gráfica de cristal líquido o GLCD es una pantalla usualmente monocromática en la que a diferencia de los LCD típicos de 16x2, su unidad mínima de gráfico no es un carácter sino un pixel; permitiendo desplegar caracteres de diferente tamaño o incluso imágenes. El tamaño comúnmente usado en este tipo de pantalla es el de 128x64 pixeles y son ampliamente usados en consolas de videojuegos o en teléfonos celulares.

El funcionamiento del GLCD de 128x64 está dividido lógicamente en dos mitades, existiendo un microcontrolador para cada mitad, uno para la mitad derecha y otro para la mitad izquierda. Cada mitad de la pantalla consiste en 8 páginas donde cada página tiene 8 bits de alto por 8 bytes de ancho (Ibrahim, 2008). Por lo tanto, cada mitad posee 64x64 bits para ser graficados tal como se observa en la Figura 2. En semejanza con los LCD de 16x2 los GLCD de 128x64 también poseen un puerto de datos de 8 bits, con la diferencia que existe pines adicionales para seleccionar con qué microcontrolador se está trabajando al momento.

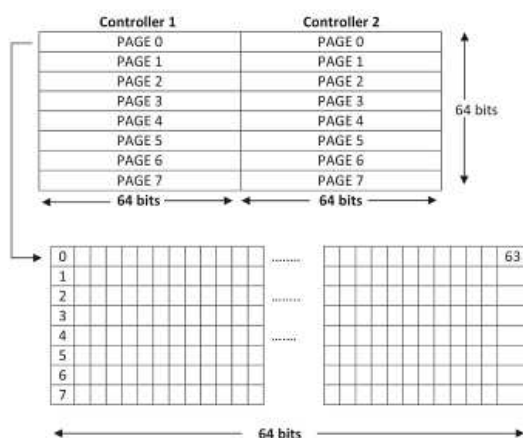


Figura 2. Estructura de un GLCD (Ibrahim, 2008)

2.1.3 Impresora Matricial

Aunque es un modelo antiguo aún es muy utilizado en el mercado siendo más visto en puntos de cobro. Es bastante ruidosa y debe tener un rollo de papel continuo para su correcto funcionamiento.

“Para imprimir utiliza una cita entintada (similar a la de las máquinas de escribir) y un cabezal con una serie de tiras metálicas o agujas que golpean la hoja a través de la cinta, de modo que la imagen o el texto es una composición de puntos creados por esos choques.” (Gallego Cano, 2005)

2.2 MÓDULOS RFID, ARDUINO Y MICROCONTROLADORES DE GAMA ALTA

2.2.1 Módulos RFID

La tecnología RFID utiliza ondas de radio para la identificación de objetos a los cuales previamente se les ha asignado un tag, la identificación la realiza un receptor o lector RFID que se encuentra fijo. “Un tag o etiqueta RFID consiste en un microchip que va adjunto a una antena de radio y que va a servir para identificar unívocamente al elemento portador de la etiqueta” (Blásquez del Toro, 2014); existen tres tipos de tags o etiquetas RFID y describen en la *Tabla 1*. Por otro lado, el lector recoge la información de las etiquetas o tags, envía comandos, datos y la energía necesaria para activar el circuito integrado del tag mediante una antena, gracias a esto no se necesita una línea de vista entre emisor y receptor, es suficiente con que ambos estén próximos.

Tabla 1. Tipos de etiquetas RFID (Blásquez del Toro, 2014)

TIPO	DESCRIPCIÓN
Pasiva	No llevan una fuente de alimentación. Utilizan para responder la energía inducida por la señal de escaneo del lector. Su alcance de lectura es de 6 metros como máximo. Son pequeñas. Operan en la frecuencia de 124 o 135 KHz.
Semi-pasiva	Incluyen una pequeña batería por lo que no requieren capturar la energía de la señal del lector.
Activa	Llevan su propia fuente de alimentación. Mayor alcance en frecuencia de 455 MHz a 2.45 GHz. Mayor alcance hasta 100 metros.

2.2.1.1 Protocolos de Comunicación.

Los lectores o receptores RFID comunican el tag leído a través de un protocolo de comunicación, siendo los más usados: Wiegand26 y ASCII que se analizan a continuación:

1. El protocolo Wiegand 26 ocupa tres líneas de transmisión, una para enviar los unos lógicos llamada DATA1, otra para los ceros lógicos llamada DATA0 y una línea de referencia o GND. Para enviar un uno o cero lógico lo que hace es mantener en bajo sólo la línea del dato por 50 us mientras la otra línea está en alto. La separación entre cada pulso es de 2 ms y ocupa niveles TTL para la transmisión. El formato de transmisión Wiegand 26 es el siguiente:
 - El primer bit B0, es la paridad Par de los primeros 12 bits transmitidos (B1:12).
 - Los 8 siguientes, B1:B8 son un Byte al que llaman Facility Code.
 - Los 16 siguientes: B9:B24 son dos Bytes al que llaman User Code
 - El último bit, B25, es la paridad Impar de los últimos 12 bits transmitidos. (El Protocolo Wiegand, 2014).
2. El modo de transmisión ASCII no representa un protocolo de comunicación en sí, simplemente se refiere que los datos transmitidos están en formato ASCII. El modo de transmisión es serial TTL y usualmente a una velocidad típica de 9600 baudios.

2.2.2 Tecnología Arduino

Arduino es una plataforma electrónica para la creación de prototipos basada en software y hardware open-source. El microcontrolador de la placa Arduino se programa en el lenguaje de programación de Arduino a través de su propio entorno de desarrollo basado en Wiring/ Processing, y no es necesario que se encuentre conectado siempre al computador ya que los programas cargados en la placa Arduino se almacenan en la memoria flash de la placa. (Arduino, 2014).

2.2.2.1 Hardware de Arduino

Dentro de la arquitectura del hardware de Arduino su microcontrolador siempre es uno de la marca Atmel AVR, esto es tanto para las tarjetas como para los shields; además las tarjetas Arduino cuentan con una interfaz USB para la comunicación con el IDE de Arduino en la PC. En la Tabla 2 se realiza una comparación entre las tarjetas Arduino más populares como son la UNO, Leonardo, Mega 2560 y la DUE. Todas estas tarjetas operan con un voltaje de 5V y pueden ser alimentadas a través de la conexión USB o con una fuente externa de 7 a 12 voltios.

Tabla 2. Comparación entre tarjetas Arduino.

CARACTERISTICAS	UNO	Leonardo	Mega 2560	DUE
Microcontrolador	Atmega328	Atmega32U4	Atmega2560	AT91SAM3x8E
Velocidad de reloj	16 MHz	16 MHz	16 MHz	84 MHz
Pines digitales E/S	14	20	54	54
Entradas analógicas	6	12	16	12
Salidas analógicas	0	0	0	2
Puertos Serie	1	1	4	4
Memoria FLASH	32 KB	32KB	256 KB	512KB
Memoria SRAM	2 KB	2.5 KB	8 KB	96 KB
Memoria EEPROM	1 KB	1KB	4 KB	0 KB

2.2.2.2 Software de Arduino

Debido a que Arduino es una plataforma abierta existen varios compiladores e IDE's (entornos de desarrollo), para este proyecto se elige el propio IDE de Arduino por su amplia documentación en línea. El IDE de Arduino se encuentra disponible para Windows, Linux y Mac OS X; gratuitamente en su página web oficial. El lenguaje de programación Arduino es un lenguaje de alto nivel y se estructura de la siguiente forma:

```
//Declaración de librerías a utilizarse  
//Declaración de variables  
  
void setup () {  
//Inicialización de puertos, variables, librerías. Solo corre una vez.  
}  
  
void loop () {  
//Programa Principal  
}
```

El lenguaje posee también las estructuras clásicas de control tales como condicionales y lazos. A continuación se da un listado de las mismas:

- if
- if...else
- for
- switch case
- while
- do..while
- break
- continue
- return
- goto

Además, el lenguaje de programación Arduino tiene sus propias funciones para configuraciones, lectura o escritura de puertos, comunicación serie, Ethernet, entre otros. Los tipos de variables que soporta el lenguaje Arduino se listan en la Tabla 3.

Tabla 3. Tipos de datos soportados en Arduino

Tipo de Dato	Descripción
boolean	Tiene dos valores verdadero o falso. Ocupa 1 byte de memoria.
char	Almacena un dato tipo carácter. Ocupa 1 byte de memoria. Va de -128 a 127.
unsigned char	Almacena un dato tipo carácter. Ocupa 1 byte de memoria. Va de 0 a 255.
byte	Almacena un número de 8 bit sin signo.
int	Almacena un valor entero de 16 bits con signo en Arduino UNO. Va desde -32768 hasta 32767.
unsigned int	Almacena un valor entero de 16 bits sin signo en Arduino UNO. Va desde 0 hasta 65536.
word	Igual a unsigned int.
long	Almacena un valor entero de 32 bits con signo.
unsigned long	Almacena un valor entero de 32 bits sin signo.
short	Almacena un valor entero de 16 bits con signo.
float	Almacena valores flotantes. Ocupa 4 bytes de memoria.
double	Igual que float.
String	Dato tipo Objeto.
array	Permite crear arreglos de variables.

2.2.3 Microcontroladores PIC de gama alta

Los Microcontroladores PIC de gama alta se caracterizan por un juego de instrucciones de 16 bits y un sistema avanzado de atención a las interrupciones, que incluye, aparte de las interrupciones internas varias entradas para interrupciones externas. Además, permiten trabajar a velocidades superiores que la de los PIC de gama media, como son los PIC de la familia 16, tienen un mayor número de timers, su pila es más grande, soportan interfaces para comunicación RS-485, algunos incluso tienen una arquitectura abierta que admite la ampliación de memoria de programa y de datos.

“La familia 18 de los PIC pertenece a la gama alta, en su gran mayoría tienen memoria de programa de tipo FLASH. Poseen además un juego de 77 instrucciones de 16 bits. Varios PIC18 están diseñados para trabajar con tensiones bajas (2.0V a 3,6V) y con corrientes inferiores a 2mA”. (Valdés Pérez & Pallàs Areny, 2007).

2.3 DISEÑO DE PCB

La metodología adoptada para el diseño de PCB se basa en la norma IPC-2221, la misma que dicta una serie de normas y guías para el diseño integro de una tarjeta electrónica.

2.3.1 Consideraciones Preliminares

En primer lugar para ubicar los elementos en una PCB se debe tomar en cuenta lo siguiente:

- Condiciones ambientales del entorno, como temperatura, calor generado por los componentes, ventilación y vibración
- Si el ensamblado es para ser mantenido o reparado.
- Si la instalación afectará el tamaño y la locación de los agujeros de montaje.
- Limitaciones de la manufactura como ancho de pista y forma y tipo de placa.
- Tecnología de ensamblaje utilizada, montaje superficial, perforado y mixto
- Selección de materiales

Posterior a esto se debe tener el diagrama esquemático/lógico en donde se muestra todas las interconexiones entre los elementos del circuito. El esquemático debe mostrar bien las áreas críticas del circuito, la distribución de la toma de poder y masa, la ubicación de puntos de prueba y la pre asignación de la ubicación de los conectores de entradas y salidas. Además se debe realizar un listado con los materiales que se ocupan en la PCB, debidamente etiquetados y referenciados en el diseño.

2.3.2 Materiales

Existen varias tipos de materiales para PCB desde los estándares hasta los altamente sofisticados y especializados. Para elegir un tipo de material se

debe tomar en cuenta temperatura (soldadura y operativa), propiedades eléctricas, interconexiones y densidad del circuito.

2.3.3 Consideraciones Eléctricas

Dentro de las consideraciones eléctricas en el diseño de PCB las más relevantes son:

- Se debe tener en cuenta el mínimo de distanciamiento entre las pistas, para tensiones de hasta 15 voltios y pistas con recubrimiento se recomienda una distancia mínima de 0.1 mm entre pistas y de 0.13 mm entre las pistas terminales de salida.
- Se divide al circuito por frecuencias de operación de sus elementos, tal como se observa en la Figura 3 donde se han dividido en tres áreas: alta, media y baja frecuencia.

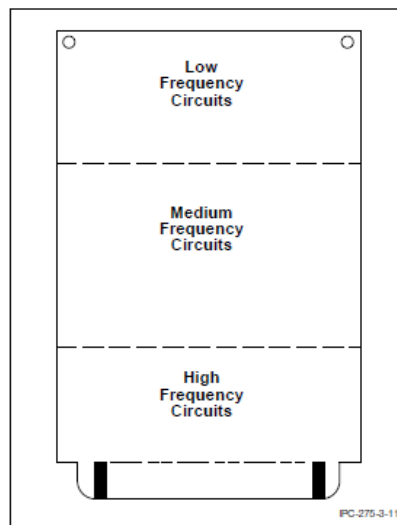


Figura 3. Distribución del circuito impreso (IPC, 1998).

- Se selecciona el ancho de pista adecuado dependiendo de la temperatura y corriente máxima que pasarán a través de la misma. En la Figura 4 se muestra la gráfica para pistas externas en donde se correlaciona el ancho de la pista con la corriente y temperatura máxima que soporta.

- Se recomienda que los cambios de línea o giros en las pistas no se hagan en 90°, sino se opte por un giro en 135°.

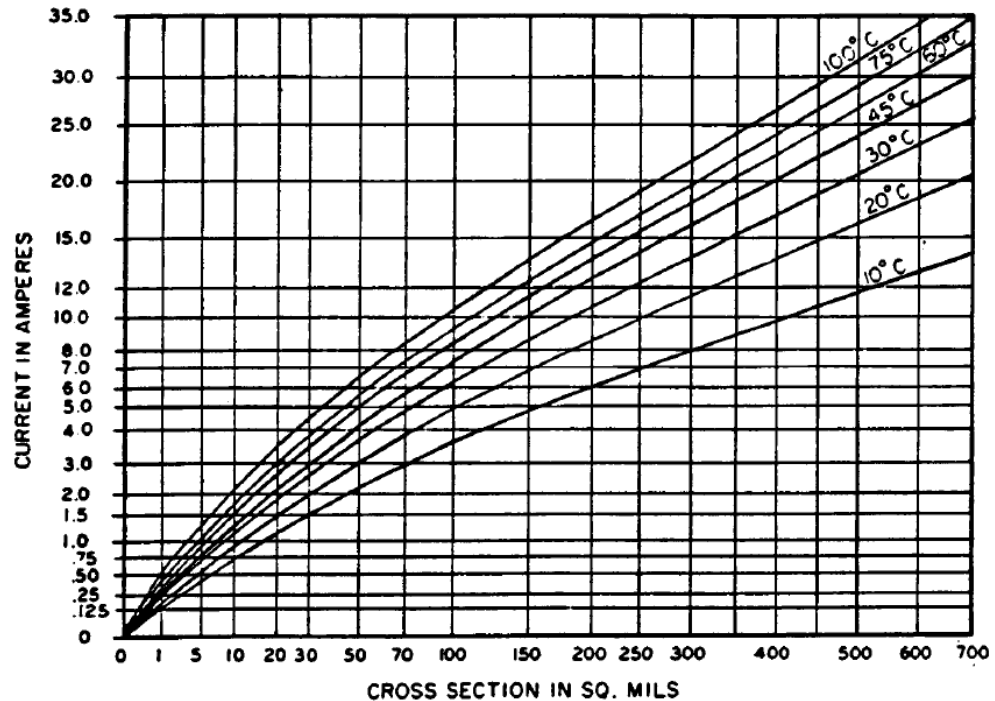


Figura 4. Relación entre ancho de pista, corriente y temperatura (IPC, 1998).

2.4 WEB SERVICES

2.4.1 Definición

“Un web service es una interfaz de red accesible a la funcionalidad de una aplicación, construida usando tecnologías estándar de internet”. (Snell & Tidwell, 2002). De esta forma, si existen dos aplicaciones escritas cada una en su propio lenguaje de programación, y una de ellas requiere los servicios de la otra lo puede realizar a través de un web service.

Como ya se ha mencionado los web services son una forma estandarizada de integrar aplicaciones WEB, lo realizan mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet. A continuación se define cada protocolo (Saffirio, 2014):

- XML, es usado para describir los datos.
- SOAP, se ocupa para la transferencia de los datos.
- WSDL, se emplea para describir los servicios disponibles
- UDDI, se ocupa para conocer cuáles son los servicios disponibles.

2.4.2 Consumir Web Services

Como ya se ha mencionado los web services se escriben en un lenguaje estándar y open source, por este motivo existen varios lenguajes de programación que han desarrollado funciones que facilitan al programador el consumo de los mismos. Entre los lenguajes más comunes están JAVA y PHP.

El lenguaje JAVA a través de sus IDE Netbeans o Eclipse proporciona al programador varias herramientas que simplifican la programación y facilitan el consumo de web services. En internet se encuentra una gran cantidad de tutoriales que muestran paso a paso como consumir un web service a través de estos IDE.

El lenguaje PHP permite que el código sea ejecutado en el servidor, el cliente solo recibirá el resultado de la ejecución del script y no será capaz de acceder al código subyacente. PHP es usado en aplicaciones WEB por lo que se embebe dentro de un código HTML, para delimitar de donde a donde es código PHP se utilizan las etiquetas: `<?php` y `?<`.

Para el presente proyecto se elige el lenguaje PHP para el consumo de web services porque la aplicación que se desarrolla es pequeña y no es compleja, por lo que no es necesario utilizar un lenguaje con tantas prestaciones como lo es Java.

CAPÍTULO 3

DESARROLLO DEL DISPOSITIVO

En este capítulo se describe el desarrollo del dispositivo de facturación en pista para estaciones de servicio, al que tiene acceso el usuario del sistema y se encuentra dividido en tres partes: hardware, software y el diseño de la carcasa en donde se alojan los componentes de hardware.

3.1 DISEÑO DE HARDWARE

El hardware del dispositivo está constituido por varios componentes con el fin de cumplir con las tareas necesarias para realizar la facturación en las estaciones de servicio.

En base a la necesidad de desarrollar un dispositivo portable de fácil operación y que permita la visualización de datos, se establecen los siguientes requisitos:

- Disponga de un microcontrolador que posea dos puertos independientes de comunicación serial, necesarios para la comunicación con el servidor y el módulo RFID.
- Permita la conexión de un teclado matricial alfanumérico de 6 filas por 5 columnas y de un LCD gráfico con una resolución de 128 x 64 píxeles.
- Admita la programación, comunicación y energización del dispositivo a través de un cable multipar enrollado.
- El tamaño del dispositivo a implementar debe ser compacto con el fin de que sea portable considerando las reglas necesarias para su desarrollo.

3.1.1 Diagrama de Bloques del Hardware del Sistema

Es una representación de las relaciones internas que existen entre los componentes que constituyen el hardware, como se muestra en la Figura 5.

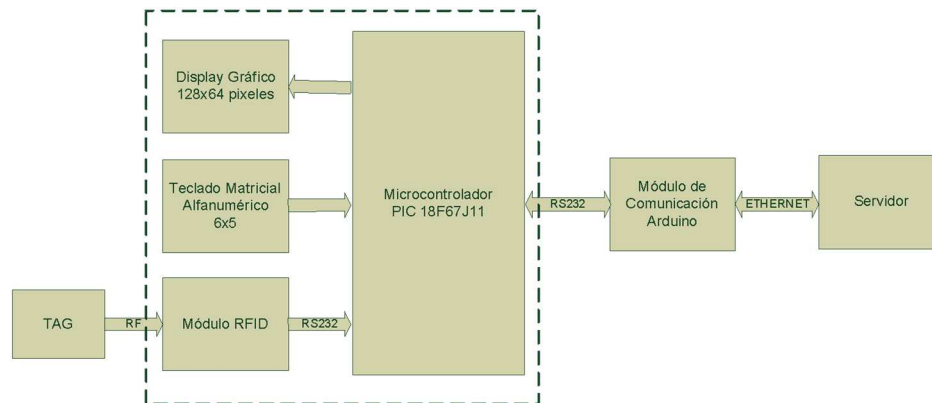


Figura 5. Diagrama de Bloques del Hardware del Sistema

3.1.2 Componentes del Hardware

3.1.2.1 Microcontrolador

El microcontrolador seleccionado es el PIC18F67J11 de la empresa MICROCHIP (Figura 6), que posee las características necesarias para realizar la aplicación deseada, a continuación se listan las más relevantes:

- Rango de voltaje de operación de 2.0 a 3.6 voltios. La tolerancia de las entradas digitales es de 5 voltios.
- Posee 64 pines y es de montaje superficial.
- El puerto B puede ser configurado vía Software para activar las resistencias pull-ups internas en todas las entradas.
- Tiene 2 canales de USART para la comunicación entre módulos.
- La memoria de programa es de 128 kB.
- Su memoria RAM es de 3904 bytes.

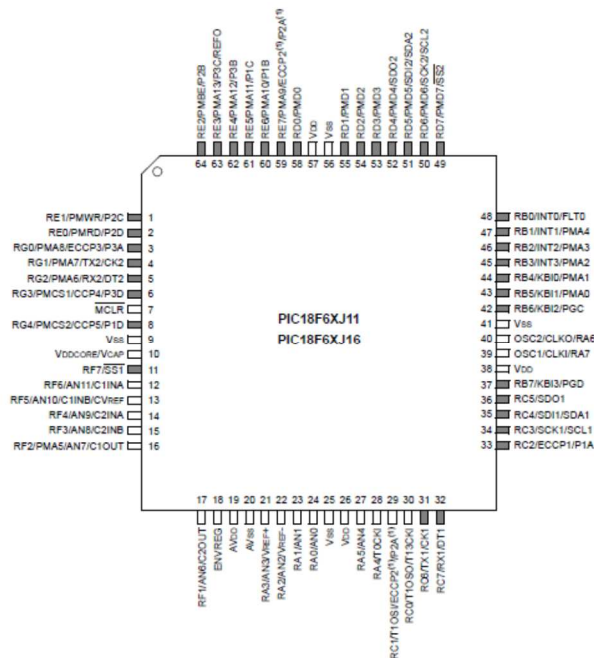


Figura 6. PIC 18F67J11 (Microchip, 2009)

Para el correcto funcionamiento del microcontrolador, el fabricante recomienda hacer las siguientes conexiones antes de proceder con la aplicación:

- Todos los pines de VDD (pines 10, 26, 38, 57) y VSS (pines 9, 25, 41, 56) deben utilizar capacitores de desacople. Para cada par de pines de alimentación se utiliza 3 capacitores en paralelo como se muestra en la Figura 7: el primero tiene un valor de 0,1uF por recomendación del fabricante; el segundo se lo coloca junto al primero para mejorar el manejo de ruido de alta frecuencia, su valor es de 1nF; y por último un tercer capacitor de 47uF para satisfacer la pérdida aceptable de tensión del dispositivo, debido a la relación entre la fuente que suministra energía y la potencia consumida por el circuito.
- Los pines AVDD (pin19) y AVSS (pin 20) deben utilizar capacitores de desacople independientemente de si se utiliza o no las funciones analógicas del microcontrolador.
- El pin ENVREG (pin 18) se conecta a VDD para activar el regulador de voltaje interno del chip, con la finalidad de que su núcleo lógico interno trabaje con un voltaje superior al nominal de 2,5 voltios.

- Los pines OSC1 (pin 40) y OSC2 (pin 39) se conectan a un cristal de 12 MHz y dos capacitores de 22 pF para permitir al microcontrolador utilizar un oscilador externo.
- El pin MCLR (pin 7) se conecta a VDD a través de una resistencia para la energización del microcontrolador y a un conector adicional con el fin de que pueda ser programado.
- Por último, los pines PGC (pin 42) y PGD (pin 37) utilizados para la programación ICSP (serial In-Circuit Programming) del microcontrolador, se conectan individualmente a una resistencia de protección de sobrecorrientes de 100 ohmios, y a un conector para su futura programación.

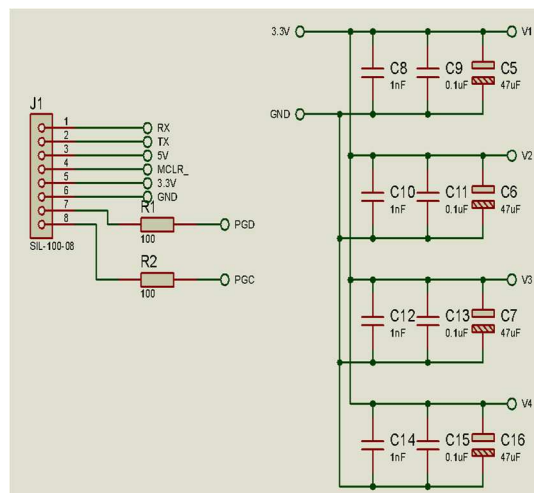


Figura 7. Esquema del Banco de Capacitores

En la Figura 8 se muestra el esquema final del microcontrolador PIC18F67J11 después de realizar las conexiones anteriormente mencionadas.

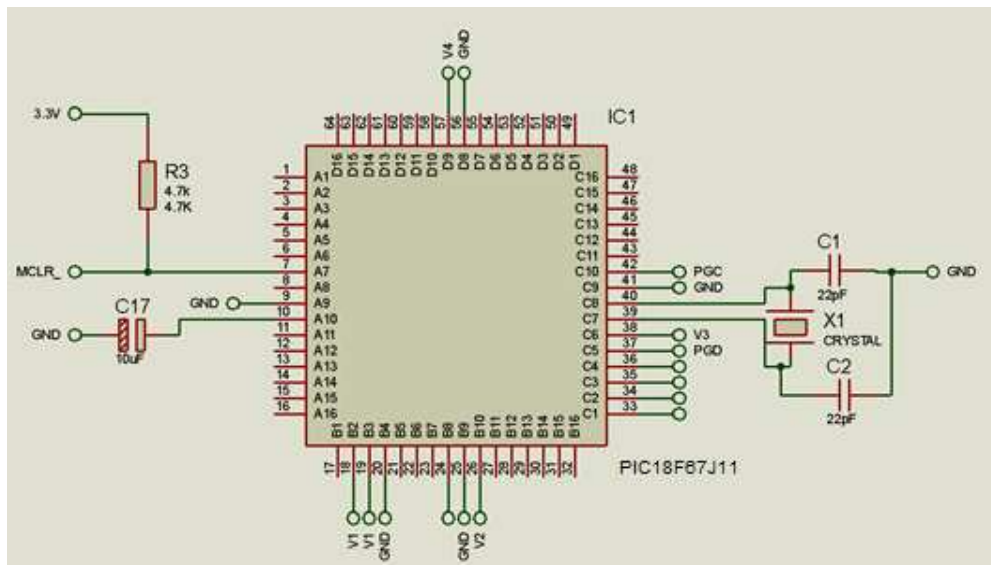


Figura 8. Esquema de conexiones Iniciales del Microcontrolador

3.1.2.2 Teclado matricial alfanumérico.

Es un teclado de membrana (Figura 9), diseñado en base a la necesidad de ingreso y validación de información, sus principales características son las siguientes:

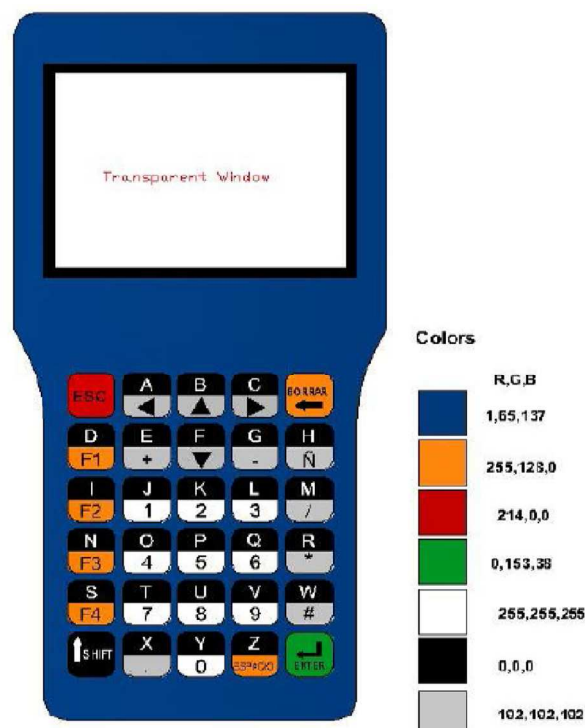


Figura 9. Teclado Matricial vista Frontal

- Teclado de membrana de 30 teclas alfanuméricas adhesivo, formado por 6 filas y 5 columnas.
- Elaborado en Autotex PET F150, tipo de poliéster para aplicaciones industriales.
- IP67 impermeabilidad contra polvo y agua.
- Teclas de domo metálico para alto uso con variedad de colores de acuerdo a su función.
- Ventana transparente para pantalla gráfica.
- Conector hembra con cable de 10cm de largo con 11 pines.
- Dimensiones exteriores 158.5 x 79.0 mm.

Las teclas se encuentran divididas de acuerdo a la categoría a la que pertenecen, tal como se observa en la Tabla 4.

Tabla 4. División de teclas por categorías del teclado matricial.

Categoría	Descripción	Teclas
Función	En pantallas que muestren varias opciones, permiten la selección de una de ellas.	F1,F2,F3,F4
Control	Realizan varios tipos de acciones explicadas en el Anexo D.	ENTER, Shift, ESC, Borrar
Numéricas	Realizan el ingreso de valores numéricos	Dígitos Del 0 al 9
Alfabéticas	Formadas por letras mayúsculas para el ingreso de información, se activan con la tecla Shift.	Letras de la 'A' a la 'Z'
Desplazamiento	Permiten desplazarse entre varias opciones hasta seleccionar la deseada	Flecha arriba, abajo, izquierda y derecha

En la Figura 10, se observa las conexiones internas del teclado, donde se visualiza su forma de matriz y la distribución de pines correspondiente a filas y a columnas; información que es necesaria para realizar la conexión al microcontrolador y para la programación de la rutina de barrido y decodificación.

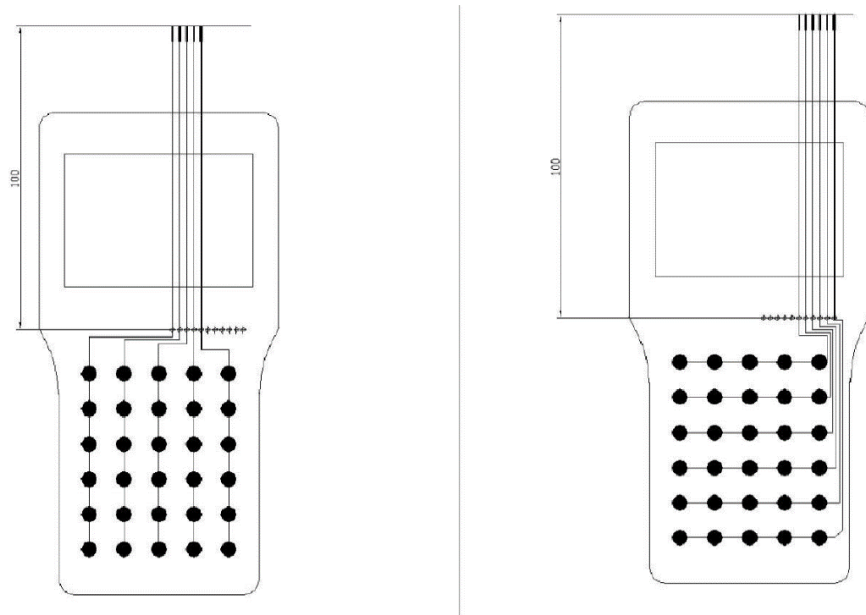


Figura 10. Teclado Matricial diagrama de conexiones.

En el diseño del conector del teclado para la PCB mostrado en la Figura 11, se detalla la conexión de los pines del teclado, las columnas se nombran con la etiqueta “TC” y las filas con la etiqueta “TF”. Las etiquetas van seguidas de un número que indica su posición y se las fue nombrando en orden de izquierda a derecha.

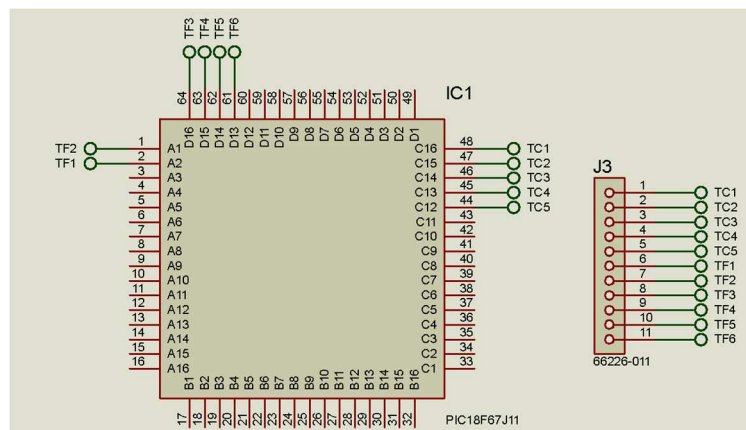


Figura 11. Diagrama de conexión del conector macho del teclado al microcontrolador.

Los pines del teclado matricial están conectados a los puertos B y E del microcontrolador, ocupando los 5 pines menos significativos del puerto B para

columnas y los 6 pines menos significativos del puerto E para las filas, así como se observa en la Tabla 5.

Tabla 5. Conexión pines del teclado matricial al microcontrolador.

Teclado		Microcontrolador	
Filas	Puerto	#Pin	
Pin 1	E0	1	
Pin 2	E1	2	
Pin 3	E2	64	
Pin 4	E3	63	
Pin 5	E4	62	
Pin 6	E5	61	
Columnas	Puerto	#Pin	
Pin 1	B0	48	
Pin 2	B1	47	
Pin 3	B2	46	
Pin 4	B3	45	
Pin 5	B4	44	

3.1.2.3 LCD gráfico de 128 x 64 pixeles.

Para mejorar la interacción del usuario del sistema con el equipo se escoge el LCD gráfico que se muestra en la Figura 12, el mismo que permite observar: letras, símbolos y gráficos. El LCD gráfico o también conocido como GLCD tiene una resolución de 128 x 64 pixeles y sus principales características son:

- Posee dos controladores KS0108, cada uno controla un área de 64 x 64 pixeles, además de una memoria RAM interna para el almacenamiento de los datos.
- Sus dimensiones externas son de 78 x 70 mm y su máxima área útil de visualización es de 38.5 x 56.27 mm.
- Su espesor es de 1.3 cm
- Opera a 5 voltios y 150 miliamperios.
- Cuenta con auto iluminación y fondo de color azul.



Figura 12. LCD Grafico 128x64 pixeles (Vishay, 2002)

El GLCD posee 20 pines divididos por su función en 3 grupos: datos, control y alimentación. A continuación se explica cada grupo:

- El primer grupo de pines lo constituyen los del bus de datos, en los cuales se lee y escribe los datos a transferir; son 8 en total y se los conecta directamente al puerto D del microcontrolador.
- El segundo grupo pertenece a los pines de control y está conformado por: CS1 (chip Select 1), CS2 (chip Select 2), D/I (Datos/instrucciones), R/W (leer/escribir), E (Enable), y RES (reset). Los pines del microcontrolador escogidos para su conexión corresponden a los pines más significativos del puerto F, todos se conectan directamente a excepción del pin E (Enable) que adicionalmente va conectado a un circuito de protección para evitar señales no deseadas.
- Por último, el tercer grupo que corresponde a los pines de alimentación del GLCD, se conectan a GND y a VCC (5v) respectivamente y el pin 5 (VR) se conecta a un divisor de tensión para controlar al nivel de contraste de la pantalla del GLCD.

En la *Figura 13* se muestra el diagrama de conexión del GLCD con el microcontrolador, por medio del conector etiquetado como J2, y en la *Tabla 6* se detalla dichas conexiones.

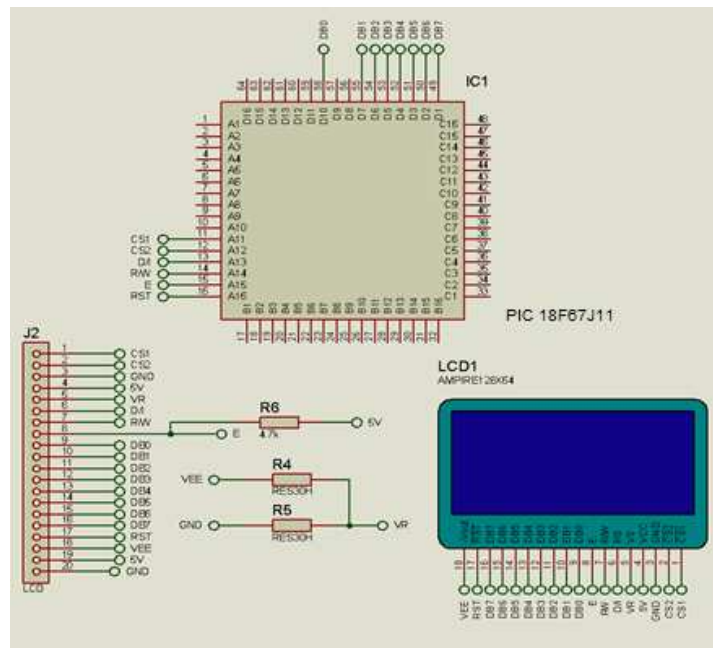


Figura 13. Diagrama de conexión del GLCD al Microcontrolador.

Tabla 6. Conexión pines de GLCD a microcontrolador

GLCD	Microcontrolador	
Pin	Puerto	# Pin
1 CS1	F7	11
2 CS2	F6	12
6 D/I	F5	13
7 R/W	F4	14
8 E	F3	15
9 DB0	D0	58
10 DB1	D1	55
11 DB2	D2	54
12 DB3	D3	53
13 DB4	D4	52
14 DB5	D5	51
15 DB6	D6	50
16 DB7	D7	49
17 RST	F2	16

3.1.2.4 Módulo RFID

El módulo RFID en el dispositivo facturador permite acelerar el proceso de venta a través del uso de tarjetas o llaveros de proximidad, usados por clientes registrados y usuarios del sistema; lo cual es un método alternativo al ingreso manual de información. En la *Figura 14*, se observa el módulo RFID ID-12LA

que se conecta en la PCB del dispositivo de facturación y se comunica con el microcontrolador, sus características principales son:

- Su voltaje de alimentación es de 2.8 a 5 V, con un consumo de corriente de 65 mA.
- Tiene una frecuencia nominal de 125 kHz.
- Puede trabajar con varios formatos de datos: ASCII, Wiegand26 y ABA Track2 magnéticos.
- Su rango de lectura sin obstáculos es de máximo 12 cm de distancia.
- Posee 11 pines y sus dimensiones externas son de 2.53 x 2.64 cm.



Figura 14. Módulo RFID ID-12LA (id-innovations, 2013)

La conexión de los pines del módulo RFID, que se observa en la Figura 15 y se detalla en la Tabla 7, se realiza de la siguiente forma:

- Para energizar el módulo RFID se debe conectar a Vcc (5v) los pines 2 y 11, y a GND el pin 1.
- Para la transmisión de datos se selecciona el formato ASCII conectando el pin 7 a GND.
- Para la comunicación con el microcontrolador se conecta directamente el pin 9 (D0) del módulo RFID al pin 5 (RX2) del microcontrolador, que es el pin de recepción de uno de los puertos de comunicación serial.

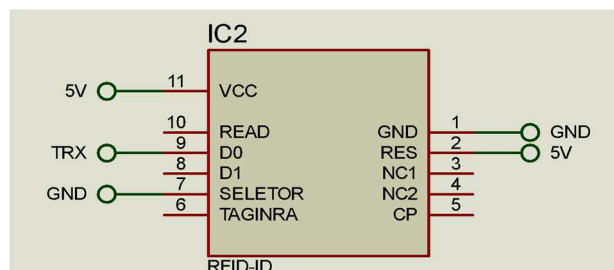


Figura 15. Diagrama de conexión del módulo RFID

Tabla 7. Conexión pines módulo RFID

Módulo RFID	Conexión
1 GND	0 v
2 RES	5v
7 SELECTOR	GND
9 D0	Puerto G2 pin 5
11 VCC	5v

3.1.2.5 Suministro de energía

El sistema de facturación en pista y el módulo de comunicación se encuentran alimentados por un adaptador de 5 voltios DC de 2 amperios (Figura 16), que se conecta a una toma de 110 voltios AC.



Figura 16. Cargador Universal de 5V y 2A (*Importec, 2013*)

La PCB del dispositivo necesita dos valores de voltaje de alimentación: 3.3 voltios para el microcontrolador y 5 voltios para los demás componentes que constituyen la PCB.

Para obtener los 3.3 voltios se implementa un circuito utilizando el regulador de voltaje LD1117v33, el cual tiene un voltaje de salida de 3.3 voltios y una corriente máxima de 800mA. En la *Figura 17* se observa el diseño del circuito del que forman parte el regulador de voltaje y dos capacitores uno de 10uF y otro de 47uF (valores recomendados por fabricante), conocidos como capacitores de carga necesarios para evitar fluctuaciones de voltaje.

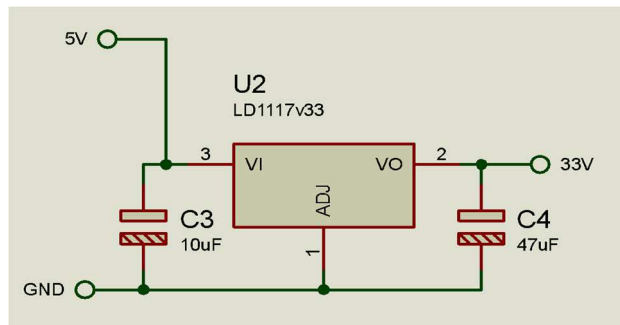


Figura 17. Diagrama de conexión de Regulador de Voltaje.

3.1.3 Diseño de la PCB (Printed Circuit Board)

Para la realización del circuito impreso del dispositivo de facturación en pista, se considera los criterios de diseño para PCBs y los requisitos de dimensiones; con la finalidad de que el dispositivo entregue una alta funcionalidad con un tamaño y peso reducido; los criterios aplicados son los siguientes:

- Realizar un diseño sencillo, usando pistas cortas y una distribución organizada de los componentes.
- Evitar pistas con ángulos de 90 grados, realizando ángulos de 135 grados para cuando sea necesario un giro.
- Implementar para los puntos de soldadura áreas circulares con el doble de ancho de la pista que terminan en él.
- Ubicar todos los componentes en una posición paralela a los bordes de la placa.
- Determinar el ancho de pista necesario para transportar la corriente máxima en base a la norma IPC-2221.

3.1.3.1 Cálculo de ancho de la pista

El cálculo del ancho de la pista se lo determina según la norma internacional IPC-2221, para lo cual es necesario conocer los siguientes parámetros: la corriente máxima que circulará a través de ellas, el incremento máximo de temperatura y por último el espesor de las pistas.

- En el diseño se considera la corriente máxima del regulador de voltaje como la corriente máxima que va a circular por las pistas;
- Para el incremento de temperatura se tomó en cuenta la temperatura máxima y mínima de las regiones del país y se realizó un promedio como se muestra en la Tabla 8.
- Por último, el espesor de la placa es determinado por la empresa que se encarga de la fabricación.

Tabla 8. Temperaturas máximas y mínimas por regiones del Ecuador
(INAMHI, 2008)

Región	Máxima (°C)	Mínima (°C)	Incremento máximo (°C)
Costa	35.2	18	17.2
Sierra	28.5	11.4	17.1
Oriente	32.5	15	17.5
Insular	30	19	11
Promedio			15.7

En la Tabla 9, se observa el consumo máximo de corriente del sistema, el cual es inferior al valor máximo de corriente soportado por el regulador seleccionado; lo cual permite, que el sistema trabaje sin riesgos de sobrecargas.

Tabla 9. Consumo de Corriente del dispositivo

Componente	Corriente
Microcontrolador	200mA
GLCD	150mA
RFID	65mA
Total	415mA

Los valores obtenidos para el cálculo del ancho de pista se muestran en la Tabla 10.

Tabla 10. Parámetros para el cálculo del ancho de pista

Variable	Descripción	Valor
I	Corriente máxima	800mA
ΔT	Incremento máximo de Temperatura	16°C
L	Espesor	1 onza/pie ²

Para el cálculo del ancho de pista se utiliza la Ecuación 1, obtenida de la norma IPC-2221.

$$\text{Ancho} = \frac{\left\{ \left[\frac{I}{k_1 * \Delta T^{k_2}} \right]^{k_3} \right\}}{L * 1,378} \quad (\text{Ecuación 1})$$

Donde:

- K1 es una constante definida por el estándar que vale 0,0150 cuando nuestra pista es interna y 0,0647 cuando es externa.
- K2 es otra constante que vale 0,5453 cuando la pista es interna y 0,4281 cuando la pista es externa.
- K3 es una constante más, que vale 0,7349 cuando la pista es interna y 0,6732 cuando es externa.

Reemplazando los valores en la Ecuación 1 se obtiene:

$$\text{Ancho pistas externas} = 5.44\text{mils}$$

$$\text{Ancho pistas internas} = 21.78\text{mils}$$

Analizando los resultados obtenidos se determina que los valores para el ancho de las pistas debe ser: para pistas externas de 5.44 mils (milésima de pulgada) que equivale a 0.14 mm, y para pistas internas de 21.78 mils que equivale a 0.6 mm.

3.1.3.2 Desarrollo de la PCB

Dentro del proceso de diseño primero se debe efectuar el diagrama esquemático del circuito electrónico, en el cual se especifican los

componentes electrónicos a utilizarse y sus respectivas interconexiones. En la Figura 18 se muestra el diseño finalizado, elaborado con la herramienta de software Isis de Proteus Design Suite, que es un Software que permite realizar el diseño del plano electrónico de un circuito.

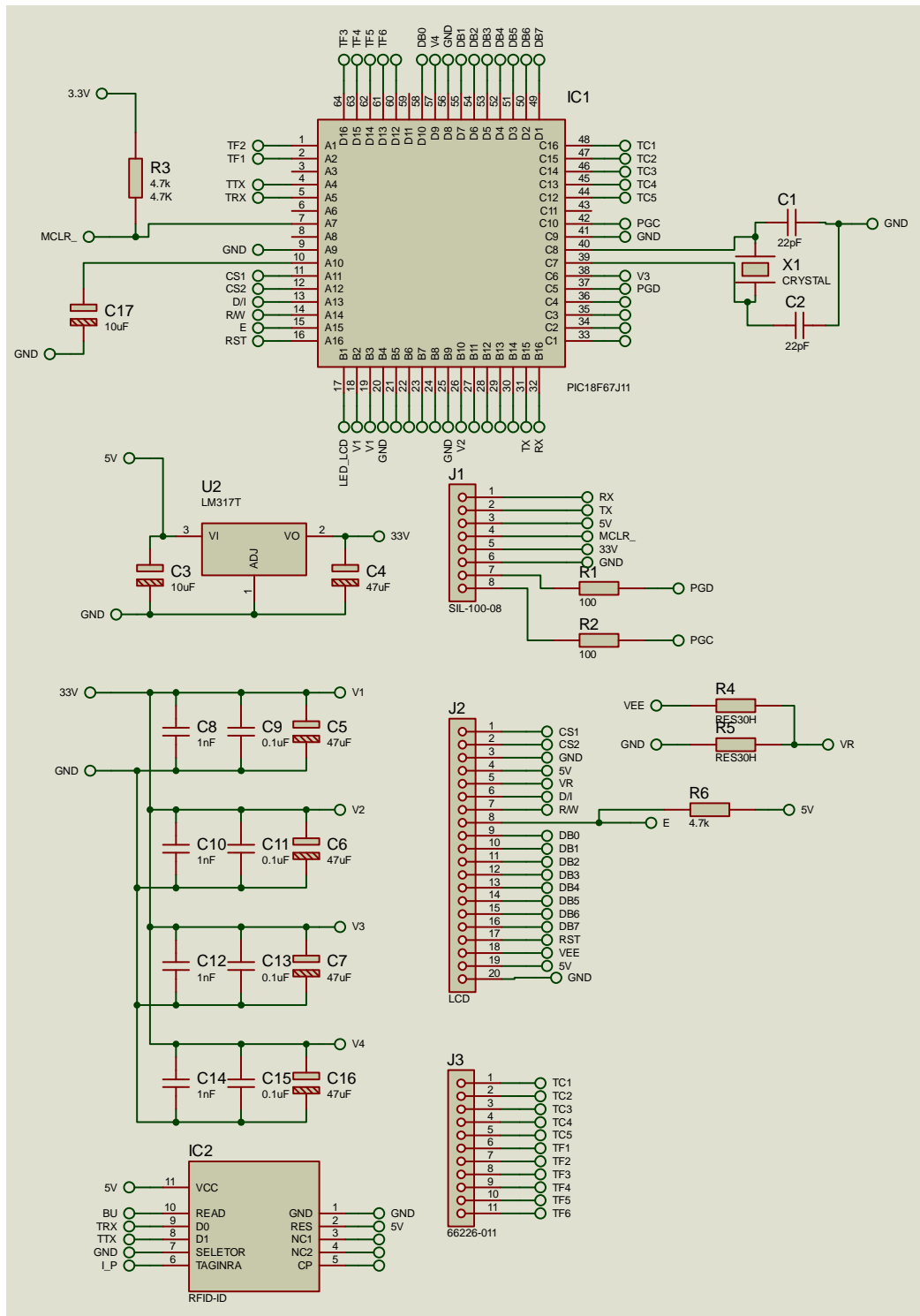


Figura 18. Diagrama del Circuito Electrónico.

Los componentes que forman parte de la placa electrónica son los siguientes:

- PIC18F67J11.
- Módulo RFID-12LA.
- Regulador de voltaje LD1117v33.
- Capacitores de Tantalio (1nF, 47nF, 0.1uF, 22pF).
- Resistencias.
- Cristal de 12Mhz.
- Conectores macho (11 y 20 pines).
- Jack Hembra RJ45.

A continuación, utilizando otra herramienta de software que forma parte de Proteus Design Suite denominado ARES, se realiza el Layout o dibujo físico de la placa, en el cual aparecen los componentes electrónicos mostrados en el diseño esquemático ya definidos su tamaño y posición final dentro de la placa, así como los caminos o pistas para la conexión de los pines de los distintos elementos.

Debido a la cantidad de conexiones a realizarse y el requisito de buscar una placa compacta, se selecciona un diseño a doble capa y se considera que algunos de los componentes electrónicos sean de montaje superficial. En la Figura 19, se observa el diseño de la primera capa o capa superior y en la Figura 20, se muestra el diseño de la segunda capa o capa inferior.

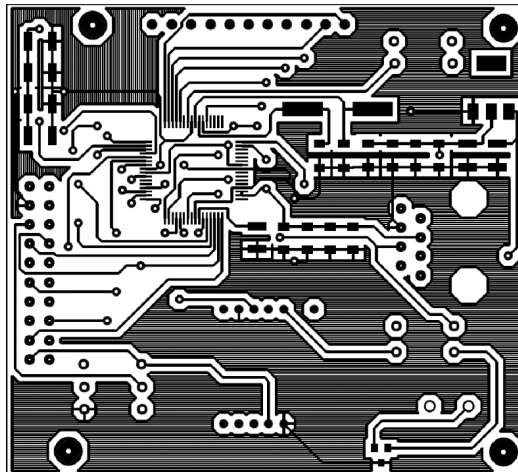


Figura 19. Circuito Impreso Capa Superior

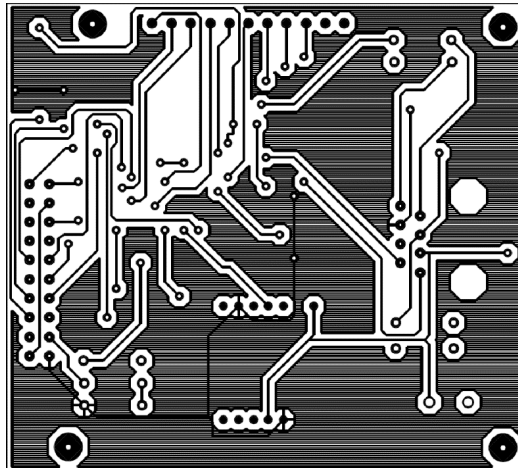


Figura 20. Circuito Impreso Capa Inferior.

En la Figura 21 se observa el diseño en tercera dimensión del circuito impreso con los elementos montados, algunos de ellos no tienen su forma real debido a que no pertenecen o no están disponibles dentro del software Proteus Design Suite. Las dimensiones finales de la placa diseñada son de 6,83 x 6,17cm.

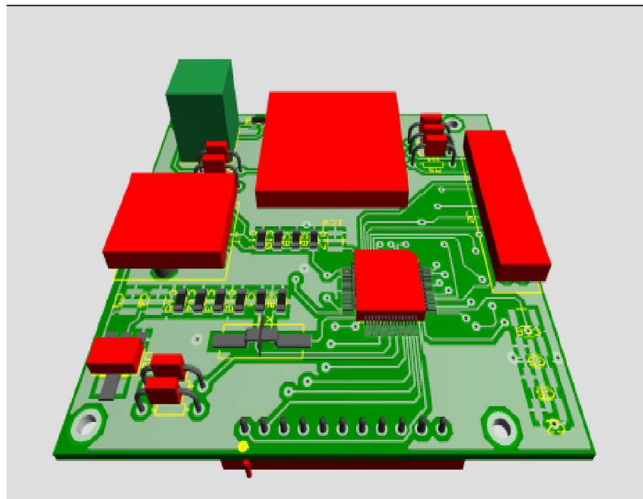


Figura 21. Circuito Impreso en 3 dimensiones.

Por último, se tiene las etapas de fabricación, ensamblaje y test, para lo cual se contrata una empresa dedicada a la fabricación de tarjetas electrónicas en máquinas CNC; las cuales entregan el producto finalizado.

Para la conexión del dispositivo al módulo de comunicación se desarrolla otra PCB (Figura 22), conectada directamente sobre el módulo de

comunicación, que permite a más de la comunicación entre dispositivos, servir de punto de conexión de la fuente de alimentación.

La placa está formada por dos conectores RJ45 hembra, un conector para alimentación y espadines para sujetarse a la placa de comunicación Arduino.

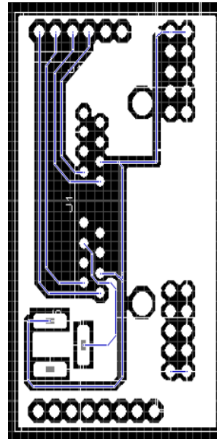


Figura 22. Circuito Impreso Placa de Alimentación y Comunicación

3.2 DISEÑO DE SOFTWARE

El desarrollo del software que gobierna al microcontrolador y sus funciones se lo realiza utilizando lenguaje C y el compilador PCWHD versión 5.008, dado que posee las librerías para el manejo del microcontrolador seleccionado.

Para la elaboración del programa primero se analiza los requisitos que este debe cumplir, los cuales son:

- Permitir la visualización de mensajes de información y selección en el GLCD, que sean claros y concisos, permitiendo al usuario del sistema estar seguro en que parte del proceso se encuentra.
- Realizar la lectura del teclado, reconociendo todos los tipos de teclas y funciones que estas cumplen, dando la oportunidad al usuario del sistema de interactuar con el dispositivo de una manera ágil y eficaz.
- Identificar al usuario del sistema y al cliente a través del ingreso de un código o Tarjeta RFID.

- Según la opción que se desee, permitir con un solo dispositivo facturador operar los dos lados de un dispensador de forma simultánea.
- Diferenciar el tipo de cliente (Normal, crédito o prepago, calibración y donación).
- Validar la información ingresada y permitir una comunicación bidireccional con el módulo de comunicación Arduino, a través del protocolo de comunicación serial.

3.2.1 Diagrama de flujo Principal del microcontrolador

En la Figura 23, se observa el diagrama de flujo principal del sistema de facturación, que se encuentra formado por varios procesos secuenciales en los que intervienen: el servidor, el cliente y el operador del sistema. Los procesos mostrados en el diagrama de flujo son explicados más adelante, y el código se encuentra en el *Anexo A.1*.

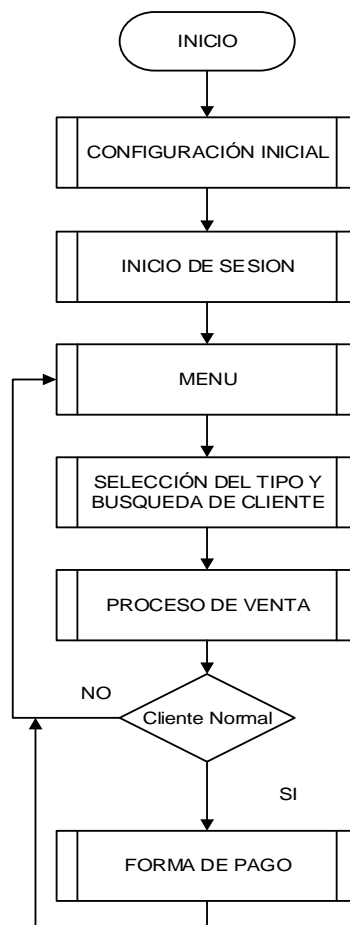


Figura 23. Diagrama de Flujo del Programa Principal.

3.2.2 Diagramas de flujo del Programa Principal

3.2.2.1 Diagrama de flujo del proceso de configuración inicial

En el diagrama de la Figura 24, se especifica el orden en que se realizan las declaraciones de configuración del microcontrolador. Las principales configuraciones a realizarse son las siguientes:

- Declaración de librerías a utilizar.
- Activación de los bits de configuración, tales como: los de protección de copia, y el Watchdog.
- Establecer la frecuencia del trabajo en 12MHz.
- Configurar los dos puertos de comunicación serial a 9600 baudios, con paridad nula, 8 bits, y 1bit de parada.
- Declaración de variables globales y métodos con los cuales se trabaja en la rutina principal.

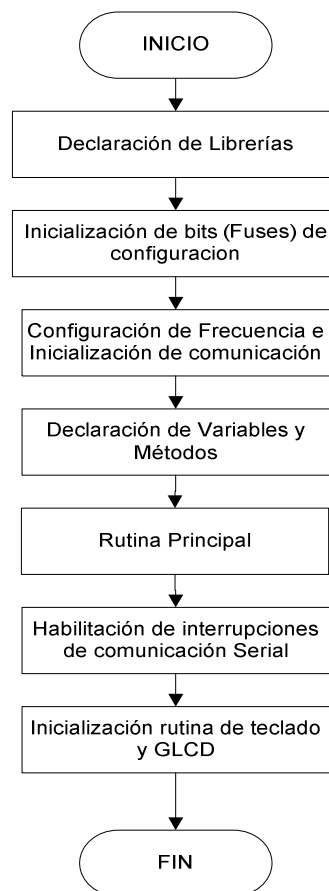


Figura 24. Diagrama de Flujo rutina Inicialización

3.2.2.2 Diagrama de flujo del proceso de inicio de sesión

En el diagrama de la Figura 25, se muestra el proceso de inicio de sesión que permite realizar un control de los usuarios que acceden al sistema.

El proceso inicia con la visualización del logo de la empresa en el GLCD, a continuación se puede iniciar sesión de dos formas: mediante una clave de usuario ingresada por teclado o a través de tarjetas de proximidad.

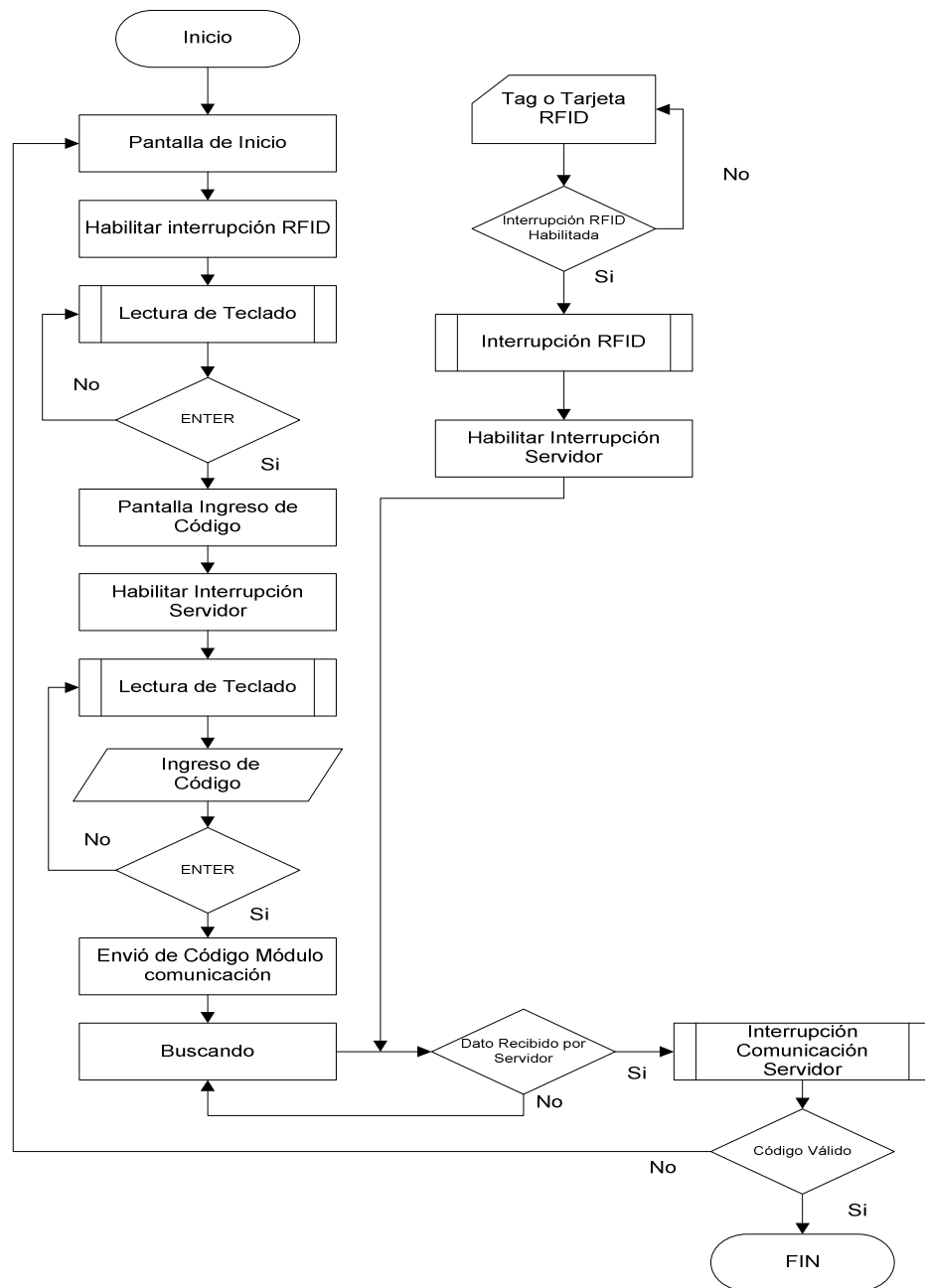


Figura 25. Diagrama de Flujo Inicio De Sesión

Para la opción de inicio de sesión con el uso de una clave, en primer lugar se debe presionar la tecla ENTER, la que despliega una pantalla que solicita el ingreso de un código a través del teclado, cuando el código es ingresado se presiona una segunda vez la tecla ENTER y el código es enviado al servidor para que sea validado. Si la respuesta del servidor es válida continúa a la pantalla de menú; caso contrario, retorna a la pantalla de inicio.

La segunda opción, es a través de una tarjeta de proximidad, que al entrar en contacto con el dispositivo activa la interrupción del puerto de comunicación serial que se encuentra conectado al módulo RFID, interrupción que lee el código de la tarjeta y lo envía al servidor a ser validado. Si la respuesta del servidor es afirmativa se despliega el mensaje de “Tarjeta valida” y el proceso continua a la pantalla de Menú; caso contrario, se muestra el mensaje “Tarjeta invalida” y se retorna a la pantalla de inicio.

3.2.2.3 Diagrama de flujo del proceso de la pantalla de Menú

La pantalla de menú es la primera pantalla de trabajo después del inicio de sesión, en la que se despliegan 4 opciones las cuales permiten seleccionar: el lado del dispensador de combustible con el que se va a trabajar ('A' o 'B'), un reinicio de todas las variables y el cierre de sesión (Figura 26).

Para seleccionar una de las opciones mencionadas se habilita la subrutina de lectura del teclado, las teclas de desplazamiento permiten ubicar el cursor sobre la opción deseada y se finaliza la selección presionando la tecla ENTER.

- Si la opción seleccionada es Lado A o Lado B el proceso continua a la pantalla de selección del tipo de Cliente.
- Si la opción seleccionada es limpiar se borra toda la información ingresada en los lados A y B, y retorna a la pantalla de Menú
- Y finalmente, si la opción seleccionada es Cierre de sesión, se muestra una nueva pantalla en la que solicita confirmación, si la respuesta es afirmativa retorna a la pantalla de inicio de proceso; caso contrario, retorna a la pantalla de Menú.

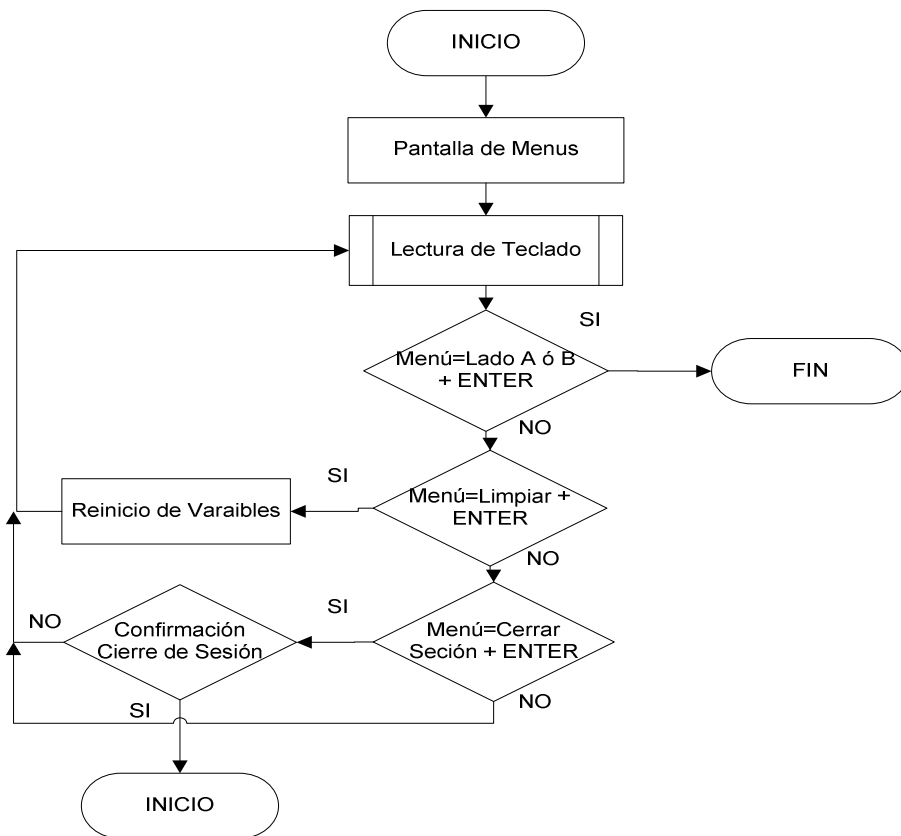


Figura 26. Diagrama de Flujo Menú.

3.2.2.4 Diagrama de Flujo del proceso de selección y búsqueda del cliente

El diagrama mostrado en la Figura 27, determina el proceso de selección e ingreso de información del cliente. Para empezar, después de seleccionar el lado de trabajo, se despliega una nueva pantalla en la que se visualiza los siguientes tipos de cliente:

- Clientes normales.
- Clientes a crédito.
- Calibración.
- Donaciones.

Una vez elegido el tipo de cliente se habilita la subrutina de lectura del teclado y la interrupción serial RFID, que constituyen los dos métodos de ingreso de información.

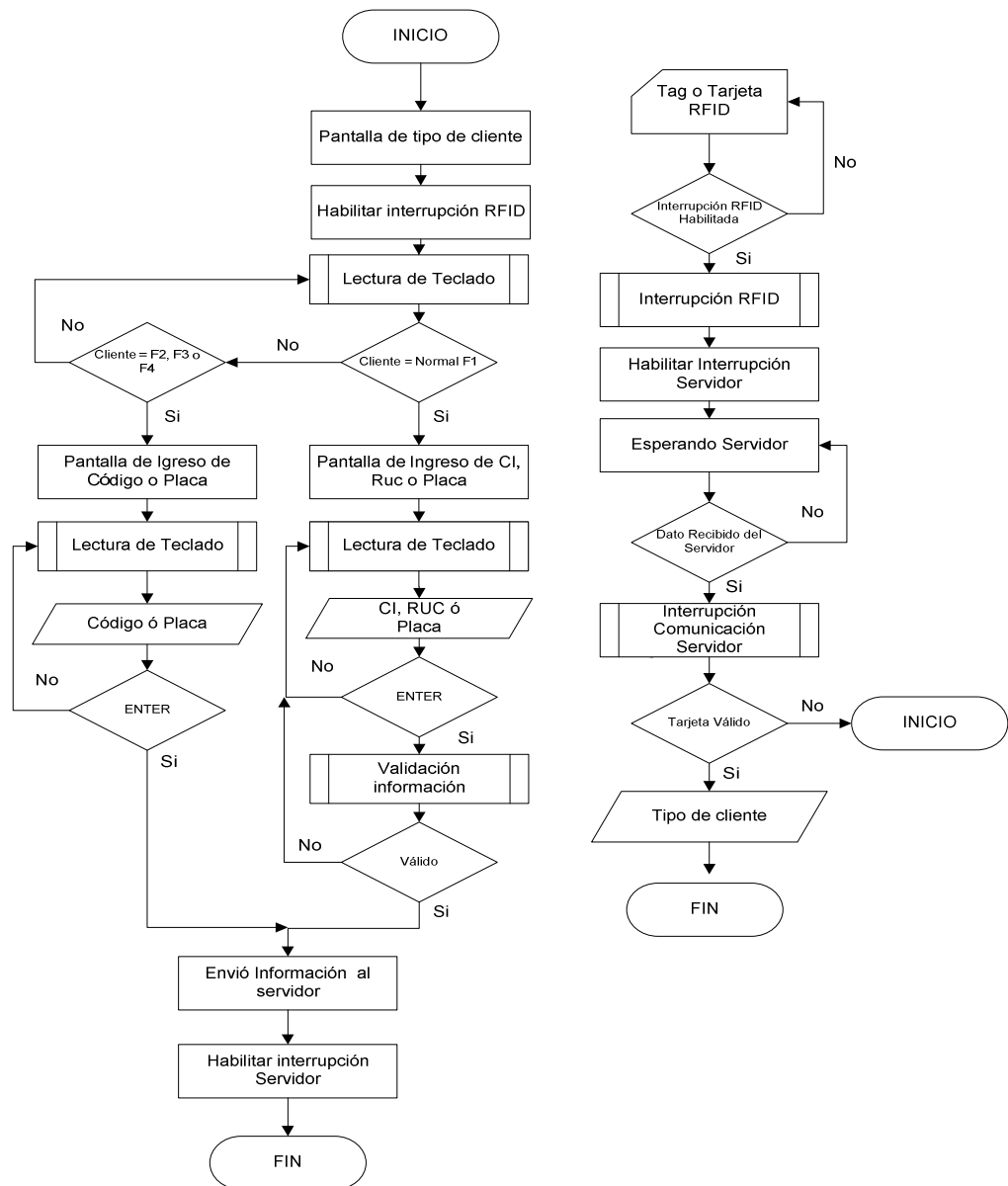


Figura 27. Diagrama de flujo selección de Cliente

Para el primer método se presiona una de las teclas de función (F1, F2, F3, F4) de acuerdo al tipo de cliente que se seleccione; acto seguido, aparece una nueva pantalla la cual solicita el ingreso de información que puede ser:

- El número de cédula, RUC o placa para clientes del tipo Normal.
- La placa o código para el resto de clientes.

A continuación, después de ser ingresada la información solicitada se presiona la tecla ENTER, la información es enviada al servidor y el dispositivo queda a la espera de su validación; en el caso de que la información ingresada

sea un número de cedula o RUC se realiza una validación previa antes de ser enviada al servidor.

El segundo método es a través del uso de una tarjeta de proximidad por parte del cliente. El cliente muestra su tarjeta al momento de realizar el despacho de combustible, al acercarla al dispositivo facturador activa la interrupción serial RFID, encargada de leer el código y enviarlo al servidor, el servidor confirma la validez y retorna el tipo de cliente al que pertenece la tarjeta.

Finalmente, después de utilizar cualquiera de los dos métodos antes mencionados, el proceso continúa a la pantalla de información, donde se muestran los datos del cliente (nombre, apellido, dirección y teléfono), que son necesarios para emitir la factura. El proceso se lo describe en el diagrama de la Figura 28.

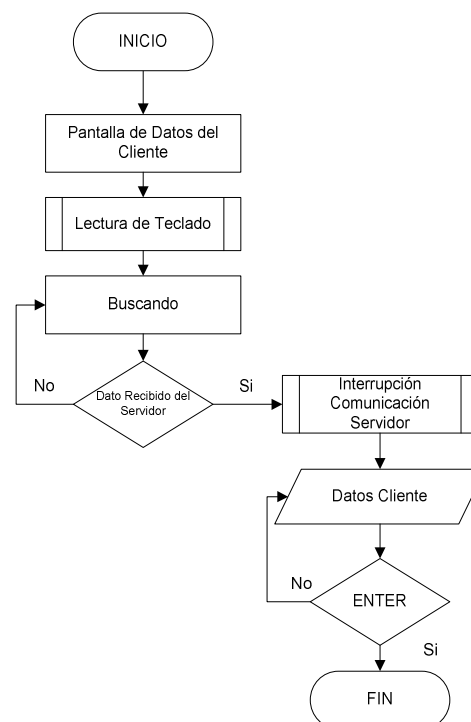


Figura 28. Diagrama de flujo búsqueda y datos del Cliente.

La pantalla de datos se muestra con los campos nombre, apellido, dirección y teléfono vacíos; y se visualiza el mensaje “Buscando”, que se mantiene hasta que el servidor envíe la información del cliente; acto seguido, se habilita la subrutina de lectura de teclado con el fin de completar la

información del cliente de ser necesario; y para concluir, se presiona la tecla ENTER que muestra una nueva pantalla continuando al proceso de venta.

3.2.2.5 Diagramas de Flujo proceso de venta

En la Figura 29, se observa el diagrama del proceso de venta en el cual se registran los siguientes datos: el producto o productos seleccionados, la cantidad de producto y la forma de pago; finalizando el proceso con la impresión del comprobante de venta.

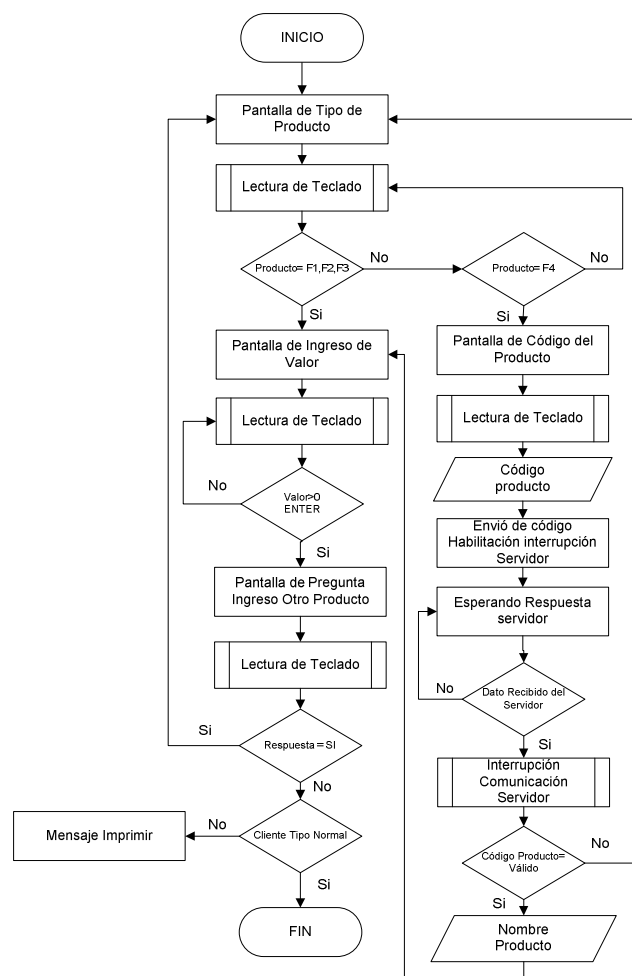


Figura 29. Diagrama de flujo proceso de venta

El proceso de venta inicia desplegando en pantalla una lista de 4 opciones de producto (Diesel, Super, Extra y Otros) que se seleccionan con las teclas de función (F1, F2, F3 y F4). Si se selecciona cualquiera de las 3 primeras

opciones el proceso de venta continua y pasa a la pantalla de ingreso de valor. Sin embargo, si se selecciona la opción F4 (Otros), se muestra una nueva pantalla que solicita el código del producto, una vez ingresado se envía al servidor a ser validado y retorne el nombre del mismo; finalmente, se pasa a la pantalla de ingreso de valor

En la pantalla de ingreso de valor se diferencia que para clientes normales y a crédito se ingresa un valor en dólares, mientras que para donaciones o calibraciones una cantidad en galones. Una vez que el valor es ingresado aparece en pantalla un nuevo mensaje, donde se formula la pregunta: ¿Desea otro producto?, si la respuesta es afirmativa se regresa a la pantalla de selección de producto; caso contrario, para clientes normales se procede a la forma de pago (Figura 30), y para los otros tipos de clientes se imprime el comprobante de venta.

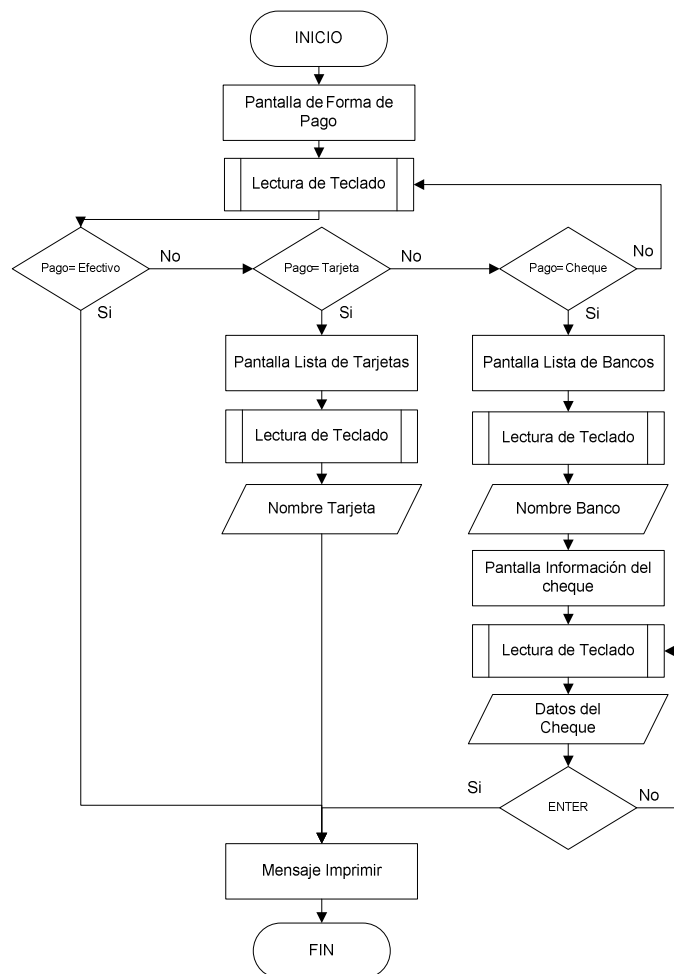


Figura 30. Diagrama de flujo forma de pago cliente Normal

En la pantalla de forma de pago, que se despliega para clientes normales, se solicita que indique una de las 3 formas de pago, que son: efectivo, tarjeta y cheque. Dependiendo de la elección el proceso continúa de la siguiente forma:

- Si se selecciona efectivo. se pasa a la pantalla de fin de proceso.
- Si se selecciona tarjeta, se muestra en pantalla una lista para elegir de nombres de tarjetas.
- Si se selecciona cheque, se muestra en pantalla una lista con los bancos más importantes del país y se solicita información adicional del cheque.

En la pantalla final se envía la información obtenida del proceso de venta al servidor y se despliega el mensaje “Imprimiendo”. Para finalizar, se imprime la factura y el dispositivo retorna a la pantalla de Menú Principal.

3.2.3 Subrutinas principales

El código se encuentra en el *Anexo A.2*.

3.2.3.1 Subrutina interrupción RFID

La interrupción RFID (Figura 31) es lanzada cuando se acerca una tarjeta de proximidad o TAG al dispositivo facturador y depende de si está o no habilitada en el programa principal.

Al aproximar una tarjeta al módulo se envía un total de 16 caracteres, el primero es un carácter de inicio de trama, los siguientes 10 pertenecen al código de la tarjeta y los últimos son de información adicional. Una vez, que se confirma el ingreso del total de caracteres se los ordena en un arreglo de caracteres (array) para luego enviarlos al servidor. Finalmente, se habilita la interrupción del servidor y se reinician todas las variables utilizadas.

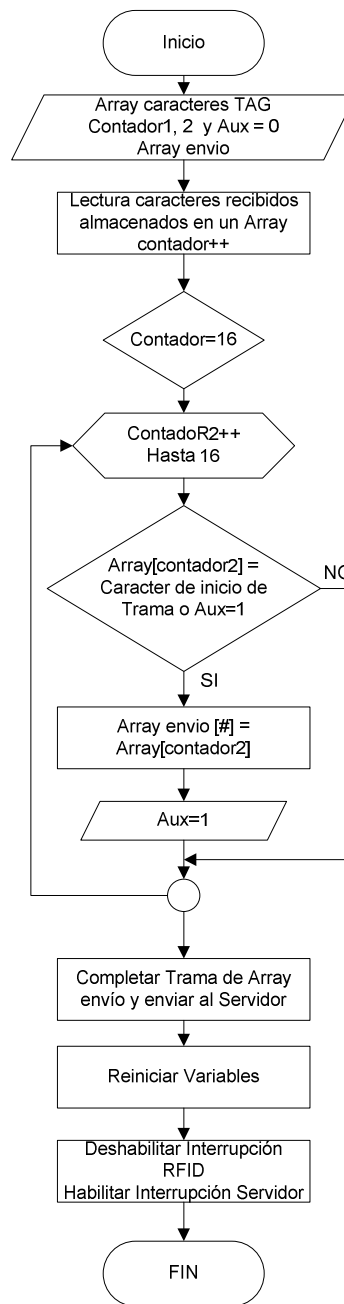


Figura 31. Subrutina interrupción serial RFID

3.2.3.2 Subrutina Interrupción del Servidor

Esta interrupción es lanzada cada vez que el servidor intenta comunicarse con el dispositivo móvil siempre y cuando la interrupción serial del servidor se encuentre habilitada. La rutina se observa en la Figura 32, y cumple la función de decodificar las tramas enviadas desde el servidor.

Según el caracter de inicio de trama que envía el servidor, se activa una bandera, esta determina qué hacer con los demás caracteres recibidos a continuación. Las funciones principales son de validación y retorno de datos:

- Validación de tarjetas RFID, inicio de sesión, información de cliente y producto adicional
- Retorno de información del tipo de cliente, datos del cliente y el nombre del producto.

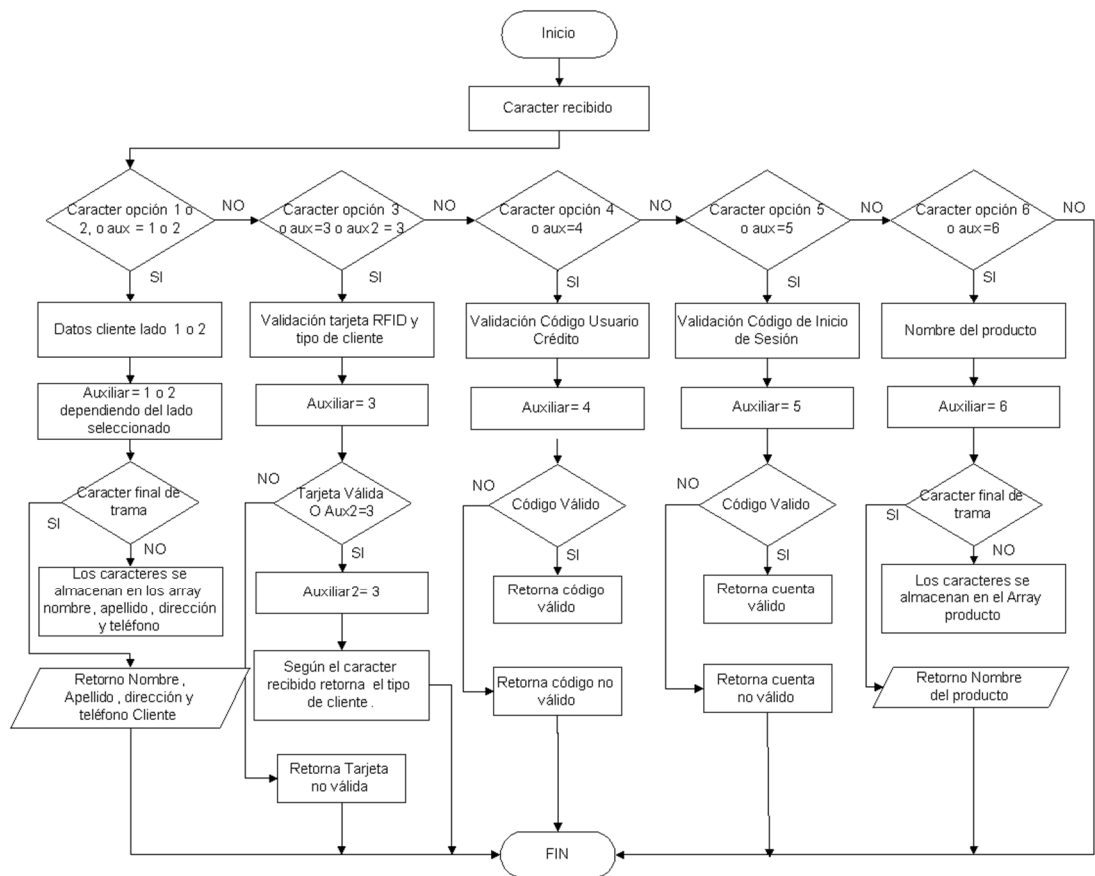


Figura 32. Subrutina interrupción serial del Servidor.

3.2.3.3 Subrutina de teclado

La subrutina del teclado que se observa en la Figura 33, es utilizada en cada uno de los subprocesos del programa principal, su función es permitir el uso del teclado matricial reconociendo y retornando el valor de la tecla presionada.

Para la selección del caracter se crean dos matrices de 30 símbolos, igual a la cantidad de teclas del teclado matricial, de las cuales la primera está formada por las letras mayúsculas y la segunda está formada por números, letras minúsculas y símbolos.

Mediante la subrutina se realiza el barrido de las filas y lectura de las columnas, obteniendo así la posición de la tecla presionada, esta posición es comparada en una de las dos matrices; si la tecla SHIFT fue presionada se compara con la primera matriz, caso contrario, con la segunda; la subrutina finaliza retornando el símbolo al que pertenece la posición obtenida en la matriz seleccionada.

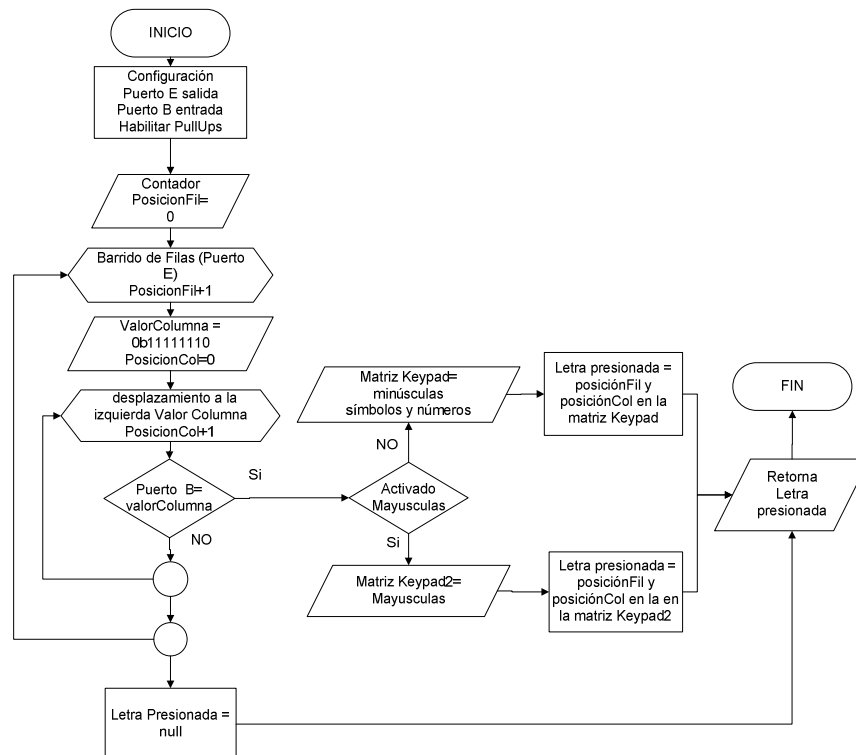


Figura 33. Subrutina del Teclado.

3.2.3.4 Subrutina de Inicialización del GLCD

Para el uso del GLCD se utiliza la librería CE_GLCD que permite la configuración inicial del GLCD. En primer lugar, se debe asociar los pines del microcontrolador que serán conectados a los pines de configuración del GLCD. Los principales métodos utilizados de la librería son:

- `Glcd_init(modos)`, se utiliza para inicializar el GLCD y debe ser declarado antes de cualquier otra función. Dentro de los valores admitidos para modos son (1,0).
- `Glcd_pixel(X,Y,color)`, selecciona el pixel del GLCD con el cual se va a trabajar, X y Y representan la posición del pixel y color admite los valores (1,0).
- `Glcd_fillScreen(color)`, cubre todo el LCD con el color dado, color admite los valores (1,0).
- `Glcd_update()`, escribe el dato mostrado en el display, en la memoria RAM del microprocesador del GLCD.

3.2.3.5 Subrutina de GLCD

En el diagrama de la Figura 34, se describe el proceso para graficar en el GLCD mediante el uso de algunos de los métodos de la librería `CE_GLCD`. Se inicia declarando las matrices que contienen los valores en bytes de los gráficos para mostrar en pantalla. Las matrices pertenecen a letras, símbolos, el logo de la empresa y 4 gráficos para la pantalla de Menú.

Se declara el método en el cual se debe considerar como valores necesarios la posición en la coordenada de X y la posición en la coordenada de Y, así como si los pixeles de la figura deben ir pintados o no.

Para dibujar en el GLCD se necesita las dimensiones del objeto a graficar tanto en X como en Y para delimitar el área de trabajo; cada valor dentro de las matrices de gráficos representa 8 pixeles consecutivos en forma vertical contados a partir de la posición dada por X y Y. Para graficar un símbolo se realiza dos barridos, el primero para la coordenada X, y el segundo para la coordenada Y, teniendo la relación de que por cada 8 incrementos en Y se incrementa un valor en X y se reinicia Y, hasta que X llegue a su límite y deba ser reiniciado; el proceso se repite hasta llegar a los límites de la figura establecido por el límite de la matriz.

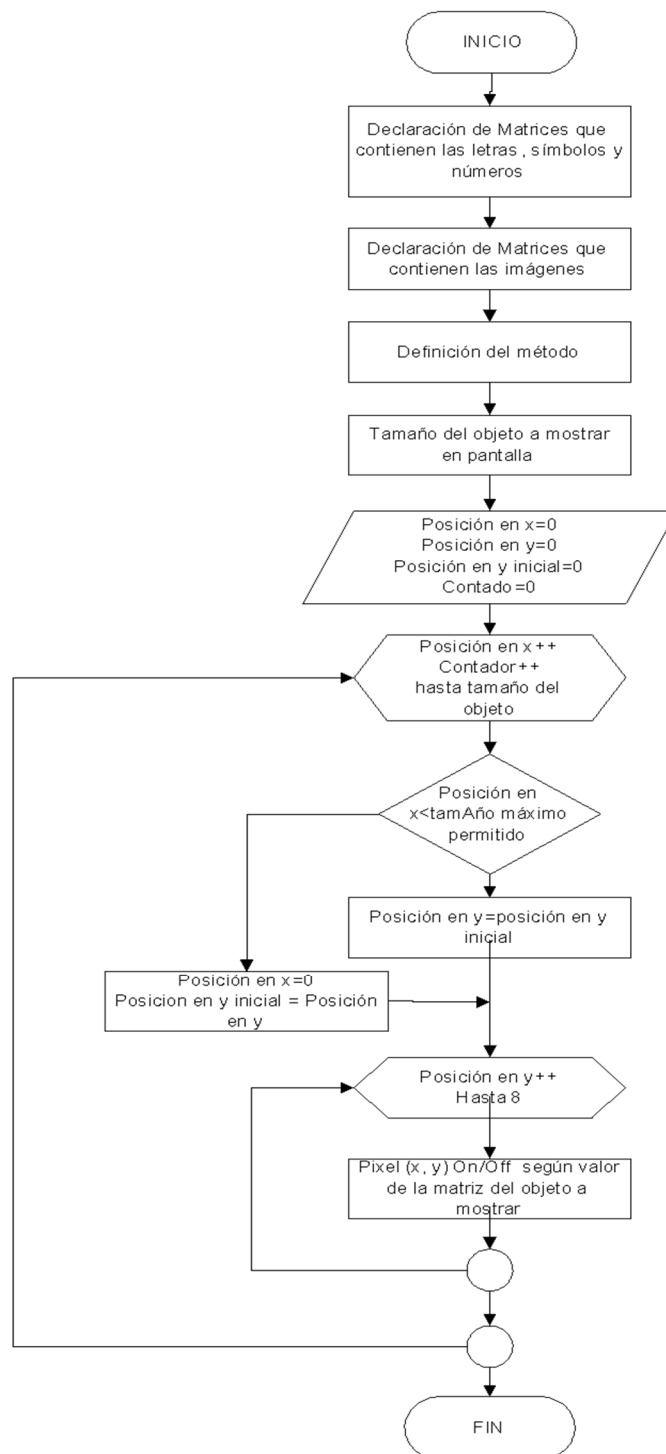


Figura 34. Diagrama de flujo de la subrutina de gráficos en el GLCD.

Utilizando el diagrama anterior, se desarrollan 6 métodos para poder utilizar de forma simple el GLCD:

- `glcd_rect(int x1, int y1, int x2, int y2, int fill, int color)`, grafica un rectángulo y los valores necesarios son la posiciones extremas en X y

Y, si el interior del rectángulo debe rellenarse (1/0) y si el área descrita debe ir pintada o no.

- `glcd_line(int x1, int y1, int x2, int y2, int color)`, grafica una línea, los valores necesarios son la posición inicial y final de la línea y si debe o no ir pintado.
- `glcd_text(int x, int y, int *texto, int size, int color)`, escribe una frase en pantalla, los valores necesarios son la posición inicial, el texto, el tamaño, y si van o no pintados.
- `glcd_char(int x, int y, char texto, int size, int color)`, escribe un carácter en pantalla, los valores necesarios son la posición inicial, el carácter, el tamaño, y si va o no pintado
- `glcd_image(int x, int y, int color)`, grafica la imagen del logo de la empresa en la posición indicada.
- `glcd_image2(int x, int y, int color, int im)` grafica la imagen seleccionada por el valor de `im` en la posición determinada.

3.2.4 Método de validación de número de cédula y RUC.

En la Figura 35, se muestra el método que valida si el número de cédula y RUC. Empieza con una validación del número de dígitos que ingresan en un arreglo tipo `char`, deben ser 10 para el caso de una cédula y 13 para un RUC (ver *Anexo A.3*). Dentro del método se dan tres casos:

- Para el caso de una cédula y de un RUC de persona natural (3er dígito del RUC es menor a 6) se procede a tomar los primeros 9 dígitos y aplicarles el algoritmo del módulo 10 y si el resultado es igual décimo dígito el método retorna verdadero.
- Para el caso de un RUC de persona jurídica (3er dígito del RUC es igual a 9) se procede a tomar los primeros 9 dígitos y aplicarles el algoritmo del módulo 11 y si el resultado es igual décimo dígito el método retorna verdadero.
- Para el caso de un RUC de empresas públicas (3er dígito del RUC es igual a 6) se procede a tomar los primeros 8 dígitos y aplicarles el algoritmo del módulo 11 y si el resultado es igual noveno dígito el

método retorna verdadero. El algoritmo de validación de cédula se explica completamente en el *Anexo J*.

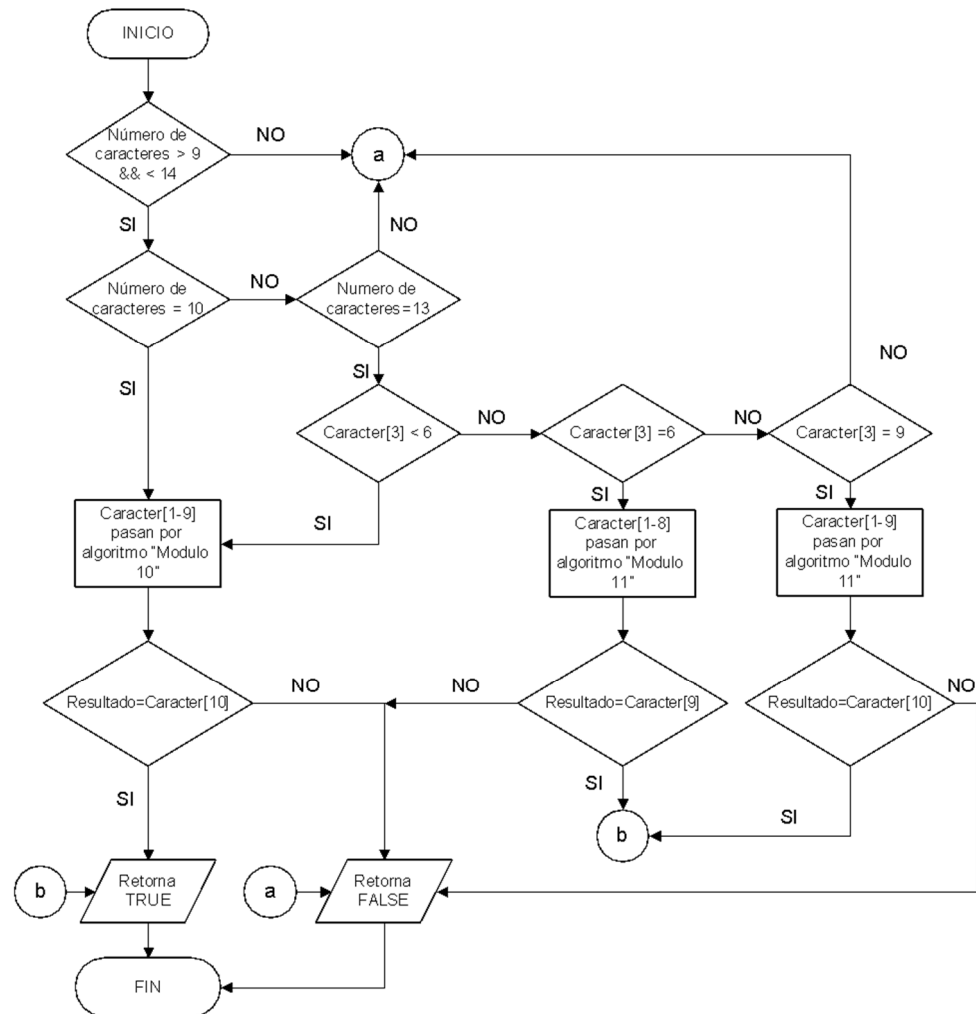


Figura 35. Diagrama de Flujo – Validación de Cédula y RUC.

3.3 DISEÑO DE LA CARCASA

Como finalización del desarrollo del dispositivo se realiza el diseño de una carcasa que contenga los elementos de hardware anteriormente mencionados (PCB, GLCD y Teclado). La carcasa se basa en diseños de controles móviles existentes en el mercado pero se ajusta al tamaño de los elementos de hardware con los que se cuenta. Para su diseño se toma en cuenta las siguientes características que debe cumplir:

- Permitir ser sujetado por una sola mano para ser operador con la otra.

- Su tamaño y forma debe ser similar al del teclado diseñado, debe seguir su contorno.
- Debe poseer el alto justo para q pueda albergar en su interior tanto la PCB diseñada así como el GLCD, sirviéndoles de protección.
- Tener los orificios necesarios para las conexiones de alimentación programación teclado y visualización del GLCD.
- Debe ser fabricado en un material resistente a golpes.

Una vez analizados los criterios para el diseño se utiliza la herramienta de Software SolidWorks 2012 con el cual se puede realizar el diseño 3D de la carcasa. El diseño se divide en dos partes: la base y la tapa.

3.3.1 Diseño de la base de la carcasa

En primer lugar se desarrolla la base de la carcasa (Figura 36), la cual sigue la forma del teclado. En su lado exterior posee varios surcos para evitar que al ser una superficie lisa pueda resbalarse de las manos del usuario y reciba algún golpe que pueda afectar a los componentes electrónicos. Internamente en la parte superior va el GLCD, por tal motivo se deja el espacio correspondiente y cilindros huecos para su sujeción mediante tornillos. En la parte inferior de igual manera se deja el espacio necesario para la PCB así como los orificios para sujetarla. Por último se deja un orificio en la parte inferior que permita conectar el cable multipar, el mismo que energiza y comunica al dispositivo de facturación en pista con el resto del sistema.

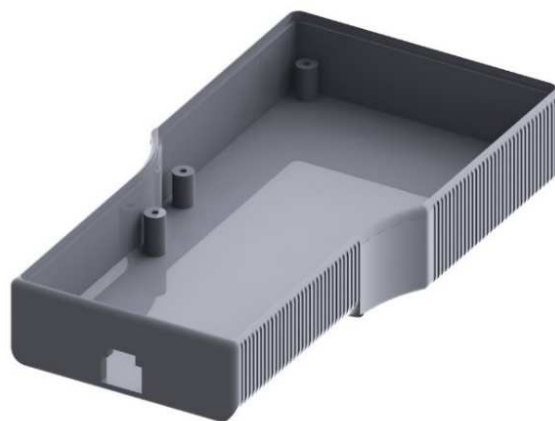


Figura 36. Base de la Carcasa

En la Figura 37 se observa una vista frontal y lateral de la carcasa en donde se indican sus dimensiones.

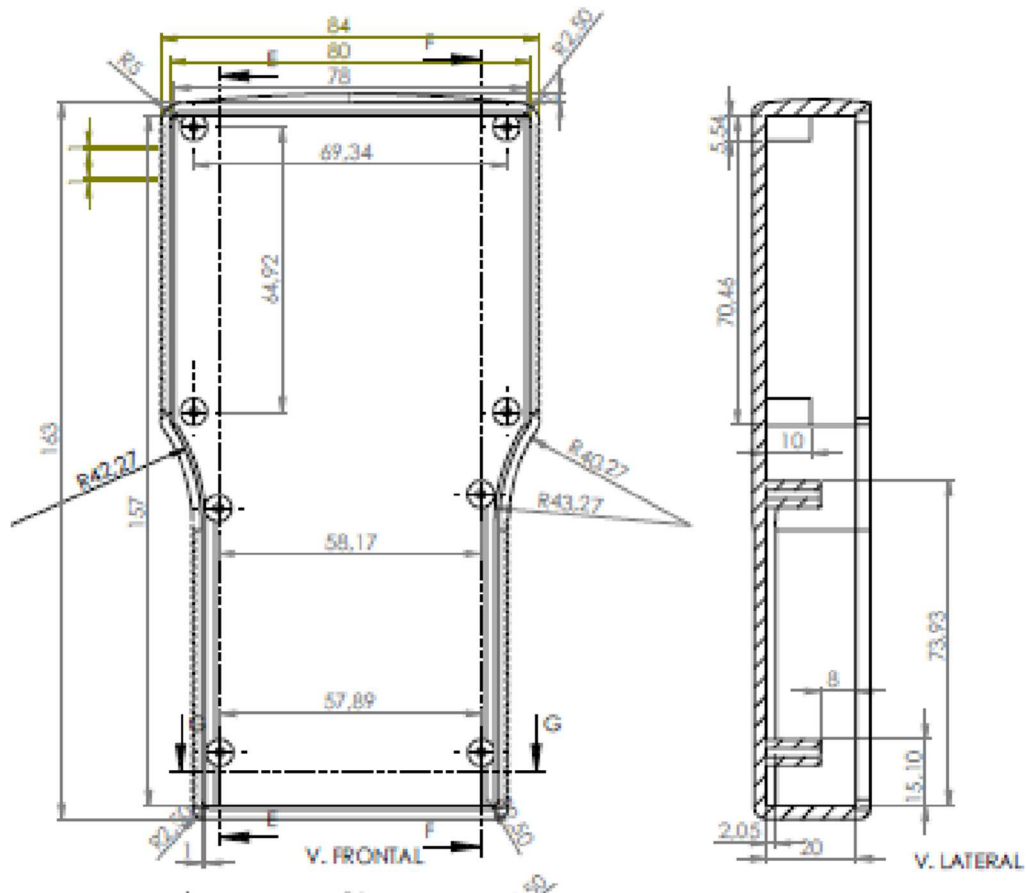


Figura 37. Dimensiones de la Base de la Carcasa.

3.3.2 Diseño de la tapa de la carcasa

El segundo elemento corresponde a la tapa de la carcasa (Figura 38) que es la superficie donde se coloca el teclado, tiene 6 orificios de 4 mm de radio que coinciden con los orificios de la base, son los encargados de mantener sujetas a las dos estructuras (base y tapa).

La tapa posee un corte en la parte superior para poder visualizar el GLCD y en la parte inferior un corte rectangular de menor tamaño utilizado para la conexión del bus del teclado con la PCB.



Figura 38. Tapa de la Carcasa.

Sus dimensiones pueden ser apreciadas en la Figura 39, que corresponden a una vista frontal y el espesor diseñado para la tapa es de 2mm.

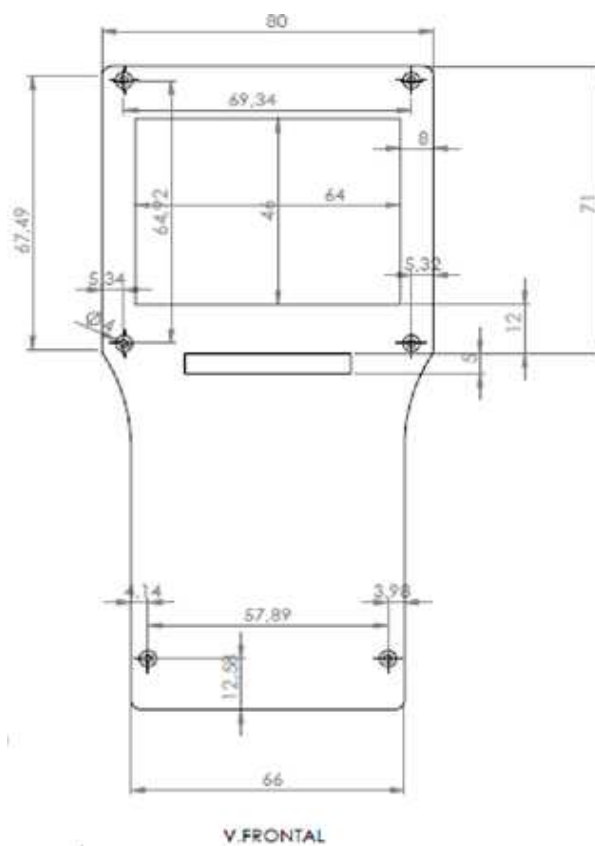


Figura 39. Dimensiones de la Tapa de la Carcasa

3.3.3 Ensamblaje final y fabricación

Una vez finalizado el diseño de las dos piezas se utiliza el software de diseño 3D SOLIDWORKS, para realizar el ensamblaje final (Figura 40), lo que permite verificar errores de dimensiones.



Figura 40. Ensamblaje final

Después de observar el ensamblaje final, se concluye que, no presenta errores y las dos piezas encajan correctamente, por lo que se guarda el archivo en formato STL, que es un formato de archivo informático de diseño asistido por computadora y es el formato con el que trabajan las impresoras 3D, servicio contratado para su fabricación.

El manual de usuario del dispositivo facturador va adjunto en el *Anexo D*.

3.4 COSTO DE FABRICACIÓN

En la *Tabla 11*, se muestra el costo total de fabricación de un dispositivo facturador, además, se detalla el costo individual de cada elemento o grupo de elementos que forman al dispositivo.

Tabla 11. Costo de fabricación de dispositivo facturador

Nombre	Modelo	Precio	Cantidad	Total
Microcontrolador	PIC18F67J11	2,68	1	2,68
Regulador de voltaje	LD1117v33.	0,42	1	0,42
Cable Flexible 3m	A2882R-10C	8,38	1	8,38
Cristal	FOXSDLF/080-20	0,22	1	0,22
Jack 8P8C	A31442-ND	1,37	1	1,37
Lector RFID	ID-12LA	30	1	30
LCD	JXD12864B	28,5	1	28,5
Teclado matricial		8	1	8
Capacitores		0,13	17	2,21
Resistencias		0,06	8	0,48
Espadines L		0,75	1	0,75
Espadines L doble		1	1	1
Cable 1 Pin F-F		0,16	20	3,2
Carcasa		22	1	22
Manufactura-Placa		13,65	1	13,65
Soldadura		36	1	36
Gastos de Importación		40	1	40
TOTAL				198,86

CAPÍTULO 4

MÓDULO DE COMUNICACIÓN

En el presente capítulo se muestra el desarrollo del módulo de comunicación que servirá de pasarela entre el dispositivo facturador en pista y el servidor; se encuentra dividido en tres partes: topología de red, descripción del hardware y diseño del software.

4.1 TOPOLOGÍA DE RED

Se entiende por topología de red la distribución física y lógica de los equipos que conforman la red y su forma de conexión (T. Eggeling, 2003). El sistema de facturación en pista para estaciones de servicio se implementa en una topología tipo estrella, esta topología presenta varias ventajas sobre otros tipos de topología, tales como:

- Reconfigurable, permite agregar o remover equipos sin afectar el funcionamiento de los otros equipos instalados.
- Tolerante, al existir un fallo o error en uno de los equipos, el resto se mantiene funcionando de forma normal.
- Cableado independiente, da mayor seguridad a la estación al evitar empalmes o conexiones intermedias cerca de los dispensadores, que puedan generar chispas capaces de provocar un incendio.

Gracias a la topología de red seleccionada el límite de surtidores a los que se puede instalar el sistema, depende de la capacidad máxima de conexiones que permita la configuración de red y de sus equipos (Conmutadores). En el país el número máximo de surtidores de combustible por estación de servicio es de 10, debido a esto no es necesario verificar ese límite.

En la *Figura 41*, se muestra la distribución de los equipos que forman parte del sistema completo de facturación en pista para estaciones de servicio. El sistema está constituido por las siguientes partes:

- Servidor, representa el nodo central del sistema, a través de él se realizan todas las comunicaciones y se procesa la información.
- Conmutador (Switch), encargado de centralizar el cableado de la red permite la interconexión de los equipos, retransmitiendo la información que llega a él.
- Internet, es necesario para realizar la consulta en línea de los datos del cliente al cual se va a realizar el despacho de combustible.
- Impresoras, imprimen la factura o el comprobante de venta; poseen un cableado independiente conectado directamente al conmutador, y se las ubica en pista.

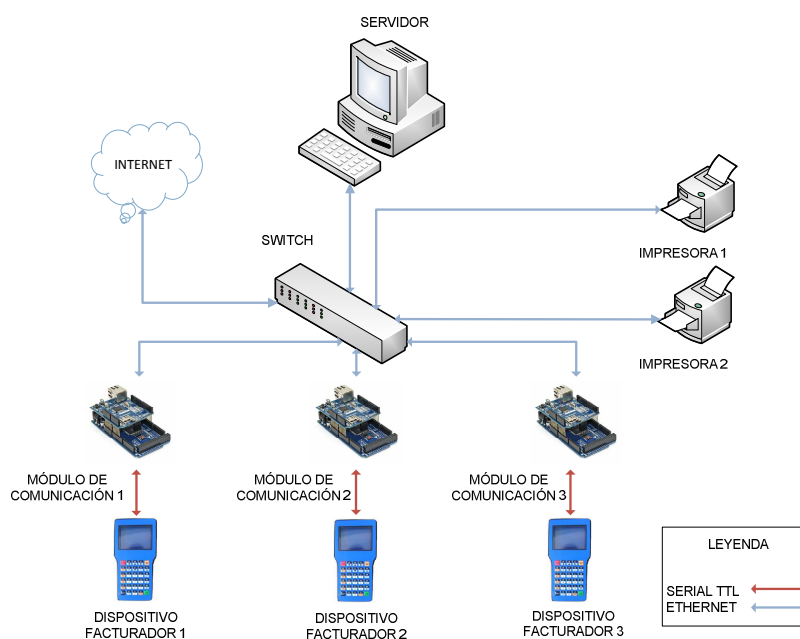


Figura 41. Topología de red del sistema de facturación en pista.

- Módulo de comunicación, tarjeta encargada de gestionar la comunicación entre el servidor y el dispositivo facturador.
- Dispositivo Facturador, control con pantalla gráfica para el ingreso y visualización de información del cliente y del proceso de facturación,

- Conexión Ethernet, se la realiza sobre cable UTP categoría 5 o superior. Sirve para la interconexión entre: servidor, switch, módulos de comunicación, impresoras y a internet.
- Conexión serial TTL, comunica el dispositivo facturador con el módulo de comunicación, se emplea un cable multipar enrollado para facilitar la movilidad y manipulación del dispositivo facturador.

4.2 DESCRIPCIÓN DEL HARDWARE

La función del módulo de comunicación es conectar al servidor con el dispositivo facturador en pista, el servidor posee una interfaz Ethernet mientras que el dispositivo una interfaz serial TTL; para llevar esto a cabo se opta por la tecnología Arduino que dentro de su gama de productos ofrece tarjetas con distintas interfaces de comunicación.

4.2.1 Diagrama de Bloques

El diagrama de la Figura 42 muestra una conexión en macro de todos los elementos involucrados en el sistema, en la parte enmarcada entre líneas punteadas se observa la interconexión de los elementos del módulo de comunicación con el resto de elementos del sistema.

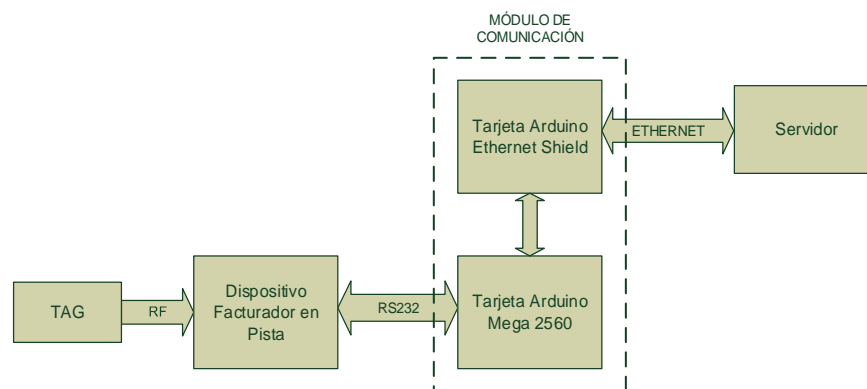


Figura 42. Diagrama de Bloques del Hardware del Módulo de Comunicación.

4.2.2 Componentes de Hardware

4.2.2.1 Tarjeta Arduino Mega 2560

En el capítulo 2 del presente texto se presenta un breve estudio sobre la tecnología Arduino y las principales características de varias tarjetas incluida la tarjeta Arduino Mega 2560, que se muestra en la Figura 43.

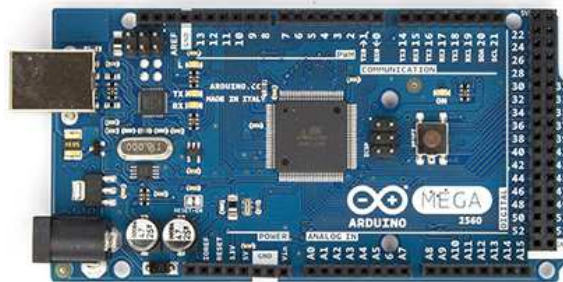


Figura 43. Tarjeta Arduino Mega 2560 (Arduino, 2014).

Para el proyecto se elige la tarjeta Arduino Mega 2560 por sus tres puertos seriales que ofrecen flexibilidad al sistema para futuras aplicaciones; por ejemplo lectura en tiempo real de las ventas en los surtidores de combustible, y además es compatible con la tarjeta Arduino Ethernet Shield cumpliendo así con los requisitos funcionales del módulo de comunicación.

La tarjeta se alimenta tanto por vía USB como por una fuente de alimentación externa. Si se utiliza la segunda opción se debe conectar los pines Vin y GND a una fuente de 9 a 12 V. Sin embargo, para el proyecto se emplea una fuente externa de 5V, conectada a los pines de 5V y GND de la tarjeta, para evitar el sobrecalentamiento del regulador de voltaje interno de la tarjeta.

El sistema de facturación entre sus requisitos principales, debe permitir el uso de un dispositivo facturador por cada surtidor o por cada lado del surtidor de combustible. Esto significa que el módulo de comunicación debe garantizar la conexión de máximo dos dispositivos facturadores, para esto se han elegido los puertos seriales 1 y 2 de la tarjeta Arduino Mega 2560; ambos puertos se

encuentran debidamente etiquetados en la tarjeta como: TX1, RX1, TX2 y RX2 respectivamente.

4.2.2.2 Tarjeta Arduino Ethernet Shield

La tarjeta Arduino Ethernet Shield, que se observa en la Figura 44, sirve de interfaz Ethernet para las tarjetas Arduino UNO y Mega. Para su funcionamiento es indispensable alguna de las tarjetas Arduino antes citadas; para el proyecto la Arduino Mega2560. Se conecta tipo torre, encima de la Arduino Mega, y queda sujeta a través de sus espadines.

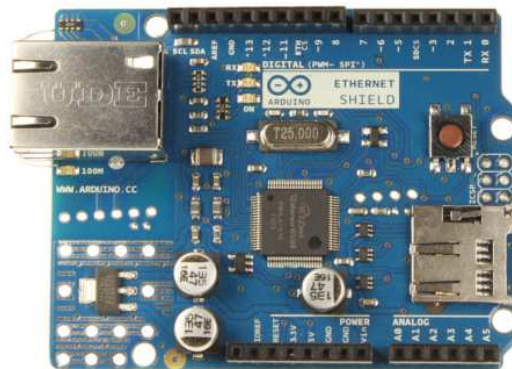


Figura 44. Tarjeta Arduino Ethernet Shield (Arduino, 2014).

Sus características principales son:

- Opera a un voltaje de 5V, suministrado por la tarjeta Arduino.
- Posee un controlador Ethernet, W5100, con un buffer de 16K.
- Velocidad de conexión: 10/100 Mb.
- Se conecta a la tarjeta Arduino a través del puerto SPI.

4.3 TRAMAS DE COMUNICACIÓN

Para la comunicación del sistema se diseñaron varias tramas de datos delimitadas por caracteres especiales y letras minúsculas. Estos caracteres facilitan la decodificación y permiten identificar que petición realiza el facturador móvil al servidor, y que información devuelve el servidor. Las tramas se han separado en dos tipos: del dispositivo facturador al servidor y

viceversa. Dichas tramas no se generan en el módulo de comunicación sino en su respectiva fuente, en el dispositivo facturador y en el servidor.

En la Figura 45 se muestra la estructura general de una trama de datos. El carácter de inicio de trama evita que el módulo de comunicación envíe datos basura a cualquiera de las dos partes; de esta forma se garantiza que el módulo de comunicación solo envíe datos cuando detecte el carácter de inicio y de fin de trama.

Además, la trama que va desde el dispositivo facturador hacia el servidor posee dos caracteres de fin de trama, el primero indica al servidor que acción realizar con los datos que le llegan, mientras que el segundo indica al módulo de comunicación, que sólo actúa como un cliente y no como un servidor, que la petición enviada espera una respuesta por parte del servidor y se debe dejar abierta la comunicación Ethernet.

Por otro lado, en la comunicación del servidor al dispositivo facturador el carácter de inicio de trama es el que identifica que acción debe realizar el dispositivo con los datos entrantes. Esto se debe a que varias tramas que envía el servidor son muy cortas y no poseen un carácter de fin de trama.

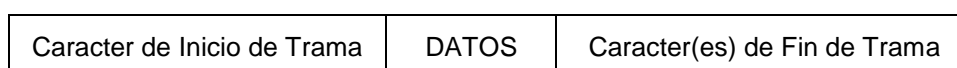


Figura 45. Estructura de una trama

4.3.1 Tramas del dispositivo facturador al servidor

El dispositivo facturador genera distintos tipos de tramas con el fin de solicitar información al servidor o enviar los datos de alguna venta realizada. En la Tabla 12 se muestran todas las tramas que el dispositivo facturador envía, como se observa en algunas el inicio de trama pueden ser distinto; esto permite identificar de qué lado del surtidor se despacha el combustible. Asimismo, se observa que el fin de trama lo componen dos caracteres, el primero permite al servidor identificar que petición es; mientras que el segundo, identifica que trama genera una respuesta en el servidor, en este

caso 'p'. En la descripción de la Tabla 12 se indica la acción que el servidor interpreta con el primer carácter de fin de trama.

Tabla 12. Tabla de tramas enviadas del dispositivo facturador al servidor

#	Descripción	Carácter Inicio Trama	Datos	Carácter Fin de Trama
1	Validación Código de Usuario	>	Código Numérico]p
2	Validación Código RFID de Usuario	>	Código RFID]p
3	Búsqueda de Cliente con tarjeta RFID	> ó <	Código RFID	{p
4	Busqueda Cliente Normal	> ó <	RUC ,Cédula o Placa	+p
5	Búsqueda Cliente Crédito, Donación o Calibración	> ó <	Tipo de Cliente y Código o Placa]p
6	Búsqueda de Producto	> ó <	Código Numérico	^p
7	Envío de datos ingresados de cliente	> ó <	Información Personal	%r
8	Envío datos de venta en efectivo	> ó <	Código Producto y Cantidad	&r
9	Envío datos de venta con tarjeta de crédito	> ó <	Código Producto, Cantidad y # de Tarjeta	&r
10	Envío datos de venta con cheque	> ó <	Código Producto, Cantidad e Información del Banco	&r
11	Cierre de sesión.	>	<Vacío>]r

4.3.2 Tramas del servidor al dispositivo facturador

En la Tabla 13 se describen las tramas que el servidor envía en respuesta a las peticiones del dispositivo facturador. Así como en el inciso anterior, los caracteres '>' y '<' identifican el lado del surtidor que solicita la información. En varios casos se ha suprimido el carácter de fin de trama debido al tamaño reducido de la misma; además, las tramas 2 y 3 de validación de clientes siempre van seguidas por la trama 1 de envío de información del cliente. En estas tramas el carácter de inicio de trama indica al dispositivo facturador que acción tomar con los dantos entrantes.

Tabla 13. Tramas enviadas desde el servidor al dispositivo facturador

#	Descripción	Carácter Inicio Trama	Datos	Carácter Fin Trama
1	Envío información de cliente	> ó <	Información Personal	%
2	Validación y Tipo de Cliente ingresado con tarjeta RFID	{	S o N para validación y N,C,X,D para tipo de cliente	
3	Validación de código de Cliente Crédito, Calibración o Donación	[S o N para validación	
4	Validación de Usuario por tarjeta RFID	}	S o N para validación	
5	Validación de Usuario por código]	S o N para validación	
6	Envío información de producto	#	Información Producto	=

4.4 DISEÑO DEL SOFTWARE

4.4.1 Diagrama de Flujo del Programa

El programa diseñado para la tarjeta Arduino permite que la misma opere como una interfaz entre la comunicación serial y Ethernet del sistema de facturación. El proceso, que se observa en la *Figura 46*, empieza con la declaración de las librerías para la comunicación Ethernet y la declaración de variables del programa. Las variables se han dividido en dos categorías: de red y del programa, de esta forma se explica que las variables de red: mac, ip, server, gateway y subnet deben ser declaradas e inicializadas para el correcto funcionamiento del módulo Ethernet. Además, se declara el método void setup() en el que se inicializan los puertos seriales a una velocidad de 9600 baudios y también el puerto Ethernet con las variables de red ya declaradas anteriormente.

La comunicación del sistema siempre inicia con una petición por parte del facturador móvil hacia el servidor; debido a esto, cada vez que se active la interrupción serial de la tarjeta Arduino implica que el facturador móvil ha enviado una trama y la variable bandera_serial se activa. A continuación, se envía los datos recibidos por el puerto Ethernet y dependiendo del último carácter de la trama, que indica si se espera o no una respuesta por parte del

servidor, se mantiene o no abierta la comunicación Ethernet; caso contrario, el programa espera recibir otra trama por el puerto serial.

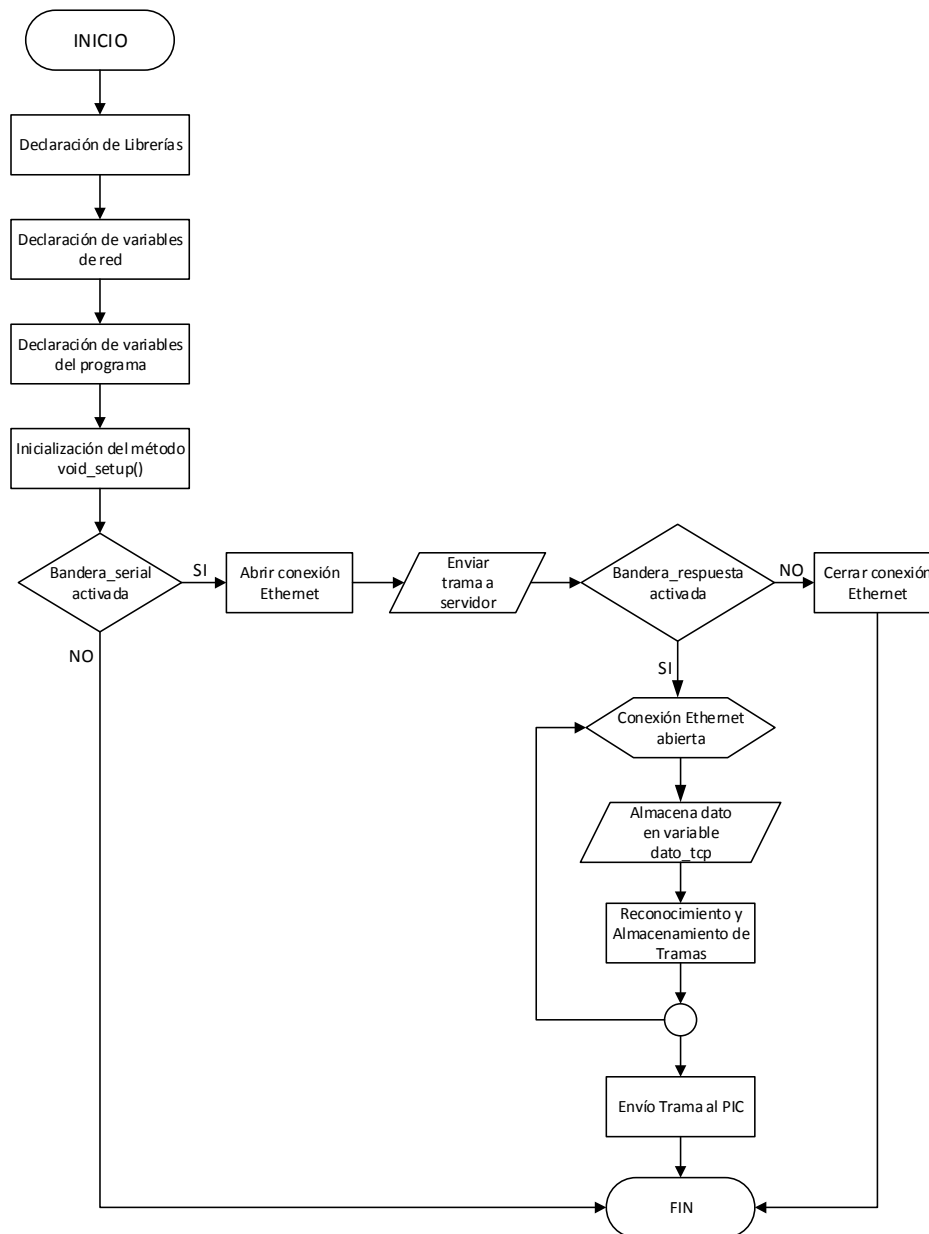


Figura 46. Diagrama de Flujo del Programa en Arduino

En caso de que se espere una respuesta por parte del servidor, el programa entra al bucle y dentro del mismo realiza el proceso de reconocimiento y almacenamiento de tramas; en este proceso el programa identifica las tramas enviadas del servidor. Como ya se dijo anteriormente las tramas 2 y 3 de la Tabla 13 van seguidas por la trama 1, cuando el programa las identifica da un retardo de 2,5 segundos antes de enviar la trama 1 para

que el dispositivo facturador alcance a graficar las distintas pantallas. En caso que el programa reciba cualquier otra trama del servidor, las almacena hasta leer el caracter de fin de trama y las envía por el puerto serial.

4.4.2 Diagrama de Flujo de Interrupción de Puerto Serie

En la atención a la interrupción del puerto serie, Figura 47, se empieza por la detección del carácter de inicio de trama que se observa en la Tabla 12, este caracter puede ser '>' o '<'. La interrupción sigue almacenando los datos entrantes en un buffer hasta detectar los caracteres de fin de trama, que determinan si la trama a enviarse al servidor genera una respuesta de vuelta; estos caracteres 'r' y 'p' solo sirven para el módulo de comunicación por lo que no pasan al servidor. El caracter 'p' activa la variable bandera_respuesta, que lleva al programa principal a un bucle que espera la respuesta del servidor; mientras que el caracter 'r' indica que no se espera nada por parte del servidor, por lo tanto se puede cerrar la comunicación Ethernet.

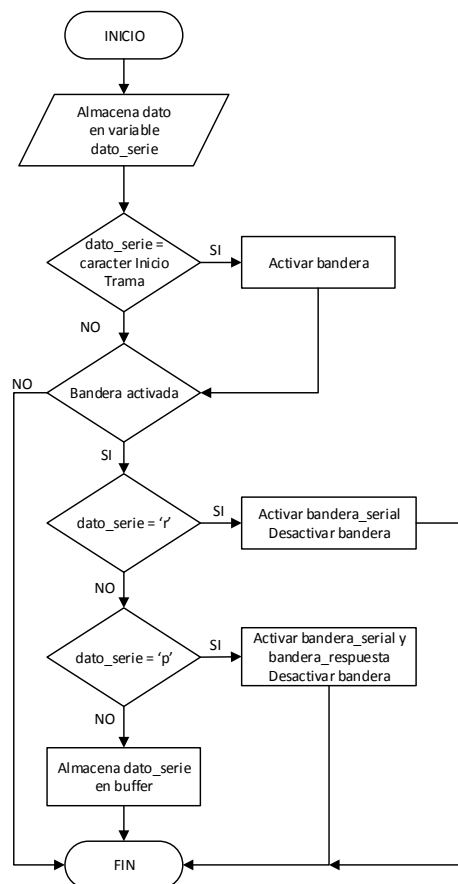


Figura 47. Diagrama de Flujo de la Interrupción del Puerto Serial

El código de programación en Arduino se encuentra en el *Anexo B*.

4.5 COSTO DE FABRICACIÓN

En la Tabla 14, se muestra el costo total de fabricación de un módulo de comunicación, además, se detalla el costo individual de cada elemento que posee el módulo.

Tabla 14. Costo de fabricación módulo de comunicación

Nombre	Modelo	Precio	Cantidad	Total
Tarjeta Arduino	Mega 2560	73	1	73
Arduino Ethernet Shield	W5100	25	1	25
Placa		13,65	1	13,65
Jack RJ45 doble		2,93	1	2,93
Soldadura		10	1	10
Gastos de Importación		5	1	5
Adaptador		20	1	20
TOTAL				149,58

En el *Anexo F* se muestra una cotización para una estación de servicio promedio de 4 servidores, indicando el costo total del sistema de facturación para dicha estación.

CAPÍTULO 5

DISEÑO DE SOFTWARE Y BASE DE DATOS

En el presente capítulo se describe el diseño del software que permite la interacción entre el dispositivo facturador y el usuario del sistema con la base de datos, se explica también el diseño de la base de datos, instalada en el servidor; y por último se muestra un manual de instalación para el software.

5.1 DISEÑO DE SOFTWARE

5.1.1 Consideraciones Preliminares

El software se implementa en la plataforma JAVA a través del IDE Netbeans versión 7.1. Se elige JAVA por ser un lenguaje de programación open-source y este IDE porque facilita el desarrollo de interfaces gráficas.

El software es el nodo central en el sistema de facturación porque controla todo el flujo de comunicación del sistema y siempre debe estar operativo. A fin de evitar que los usuarios cierren el software, y por ende el sistema de facturación deje de funcionar, se lo divide en dos aplicaciones denominadas:

- Administrador de datos.
- Cliente-Servidor.

La aplicación Administrador de Datos permite la interacción entre los dispositivos facturadores y la base de datos, mientras que la aplicación Cliente-Servidor permite la interacción entre el usuario del sistema y la base de datos, tal como se observa en la Figura 48. Debido a esto, ambas aplicaciones se conectan independientemente con la base de datos pero sólo la aplicación Cliente-Servidor posee una interfaz gráfica.

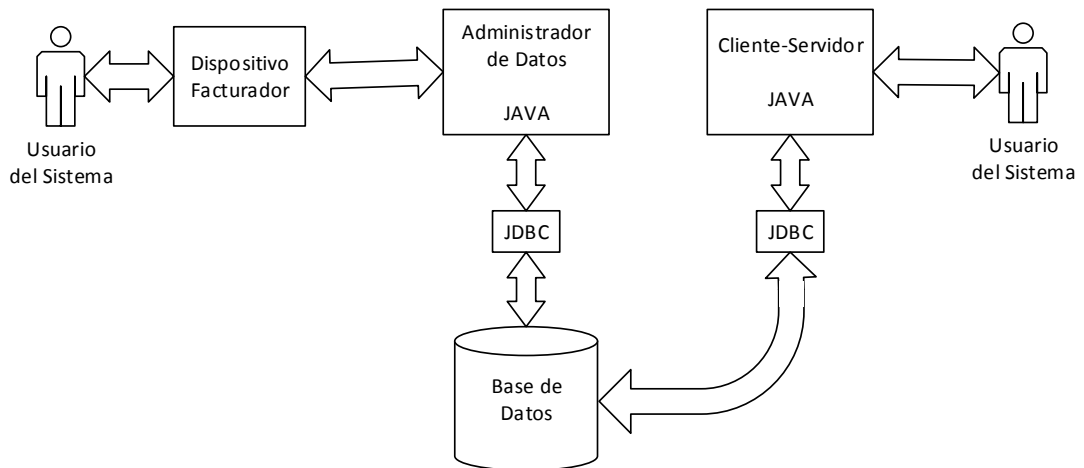


Figura 48. Aplicaciones del software

5.1.1.1 Funciones de la aplicación Administrador de Datos

Esta aplicación se ejecuta en segundo plano y arranca cada vez que se enciende el servidor, no posee una interfaz gráfica sólo un icono en el área de notificación que indica si se está ejecutando. Cumple con las siguientes funciones:

1. Conecta con la base de datos del servidor.
2. Funciona como servidor en la arquitectura cliente – servidor del sistema de facturación, permitiendo al dispositivo facturador realizar consultas, ingresos y modificaciones en la base de datos.
3. Direcciona que impresora imprime la factura enviada por el dispositivo facturador.
4. Implementa un método para consumir el web service “Merlyna”. Este web service, provisto por una empresa ecuatoriana, permite obtener datos como nombre o razón social a través de números de cédula, RUC o placas de vehículos; pero para este caso solo se utiliza el servicio de consulta por número de cédula.
5. Utiliza Merlyna en caso de no encontrar los datos del cliente en la base de datos.

5.1.1.2 Funciones de la aplicación Cliente-Servidor

Esta aplicación posee una interfaz gráfica que realiza las siguientes funciones:

1. Conecta con la base de datos del servidor
2. Permite el manejo de la información almacenada en la base de datos para las tareas presentadas en la Tabla 15, donde se explica todas las acciones que la interfaz gráfica debe cumplir y se las ha agrupado en tareas para su mejor comprensión.
3. Restringe el acceso, a través de una contraseña, a cualquier usuario que intente utilizar el software. Además, debe garantizar dos niveles de acceso, 0 y 1, donde los usuarios de nivel 0 solo puedan realizar la tarea "Reportes" y los de nivel 1 todas sin excepción; garantizando la inalterabilidad de la base de datos por usuarios no deseados.

Tabla 15. Tareas y Acciones realizadas por el software

TAREAS	ACCIONES
Hacer Reportes	Generar reporte de todos los productos registrados. Generar reporte de impresoras configuradas. Generar reporte de empleados registrados. Generar reporte de turnos creados. Generar reporte de ventas, ya sea en un período de tiempo o de un determinado cliente.
Administración de Clientes	Ingresar nuevos clientes, ya sean normales o a crédito. Si son de crédito se debe permitir el ingreso de dinero abonado. Editar información de clientes registrados.
Anulación de Facturas	Anular alguna factura realizado por el dispositivo facturador.
Cerrar Caja	Ingresar en detalle el dinero efectivo del total de ventas del día Mostrar en una tabla todas las transacciones del día en cheque o tarjeta. Solicitar nuevamente contraseña para identificación de usuario. Si el usuario es nivel 1 puede modificar todos los cierre de caja de los usuarios de nivel 0.
Configuración	Modificar precios de productos. Ingresar nuevos productos. Asociar las impresoras con los dispositivos facturadores. Asociar los dispensadores con los dispositivos facturadores, y asignar un número en cada manguera. Ingresar y modificar nuevos empleados. Crear turnos.

5.1.2 Arquitectura de la aplicación Cliente-Servidor

En este apartado se explica la arquitectura que posee la aplicación Cliente-Servidor que es la única que posee una interfaz gráfica. En la Figura 49, se observa la arquitectura de la aplicación así como su navegación entre las distintas pantallas a desplegarse dependiendo del nivel de acceso del usuario del sistema.

Para cada tarea de la Tabla 15 se destina una o dos pantallas, como es el caso de Configuración y Administración de Clientes. Para la tarea de Configuración se destina una pantalla para todo lo referente a productos y dispositivos facturadores y otra para ingreso de empleados y creación de turnos. Por otro lado, para cada acción de la tarea “Hacer reportes” se genera una pantalla distinta con la misma distribución, pero se decidió agrupar todas estas pantallas en una sola denominada Reportes para su mejor comprensión.

La navegación entre pantallas es de forma horizontal, por lo tanto el usuario puede cambiar de pantalla en cualquier momento. Además, se agrega una pantalla de Ayuda que despliega un PDF con el manual de usuario de la aplicación Administración.

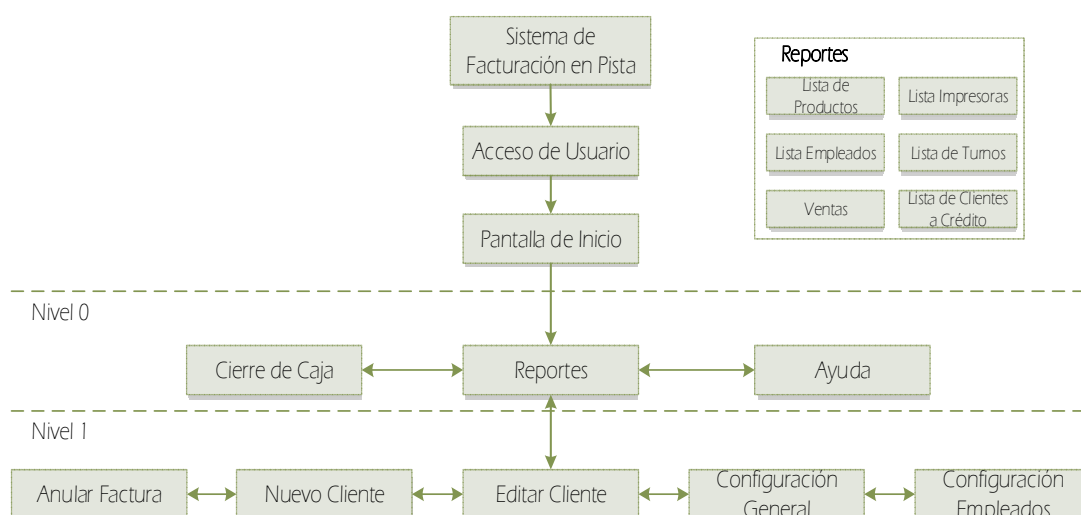


Figura 49. Arquitectura de la aplicación Administración.

5.1.3 Distribución de pantallas de la aplicación Cliente-Servidor

Para el diseño de las distintas pantallas de la aplicación se realizan 4 esquemas que muestran una distribución general de los elementos de cada pantalla. Cada esquema se explica a continuación:

1. Esquema 1

Este esquema (Figura 50) es usado para diseñar las pantallas de: Nuevo Cliente, Editar Cliente, Anular Factura, Configuración de Empleados y Reporte de Ventas. Cuenta con la siguiente distribución:

- Menú de Navegación.- Permite al usuario del sistema elegir que pantalla desea desplegar.
- Título.- Indica que pantalla es.
- Panel de Búsqueda.- Se colocan los campos necesarios para realizar una búsqueda dependiendo de la pantalla que sea.
- Despliegue o Ingreso de Datos.- En esta parte se muestran los datos que se buscaron en el panel de búsqueda y en algunos casos permite la modificación de los datos. En el caso de Ingreso de Datos se cuenta con botones de control para confirmar si se desea realizar los cambios.

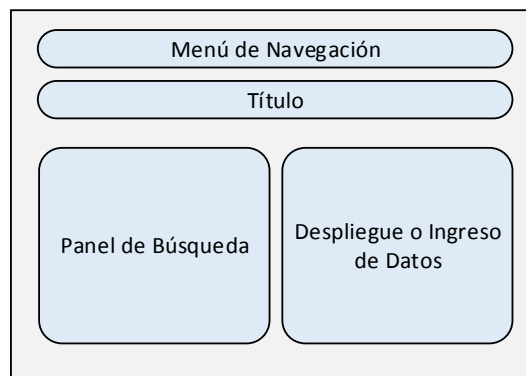


Figura 50. Esquema 1

2. Esquema 2

Este esquema (Figura 51) es usado para el diseño de la pantalla Configuración General, los dos primeros campos Menú de Navegación y

Título son los mismos que en el esquema 1, los tres restantes poseen los campos necesarios para la configuración de dispositivos facturadores, productos e impresoras, cada uno de ellos posee botones de control para confirmar o no los cambios a realizarse.

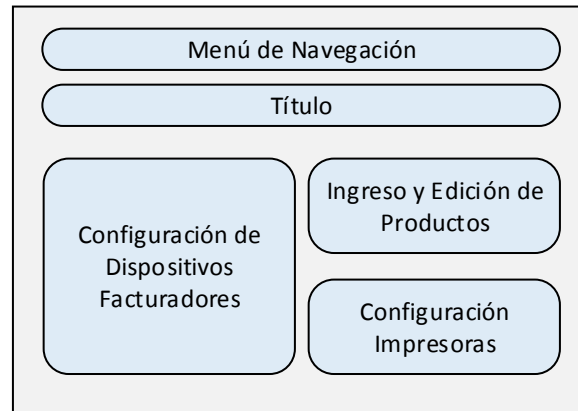


Figura 51. Esquema 2

3. Esquema 3

Este esquema (Figura 52) es usado para el diseño de la pantalla Cierre de Caja. Los dos primeros campos son los mismos que en el esquema 1, los restantes se explican a continuación:

- Datos Empleado.- Muestra el nombre, ID y el turno del usuario que ingreso. Si el usuario es nivel 1 se permite modificar, a través de una lista, el cierre de caja de los usuarios de nivel 0.
- Ingreso Efectivo.- Posee los campos necesarios para detallar el dinero recaudado en efectivo.
- Resumen de Caja.- Presenta un resumen automático de todas las ventas realizadas con el dispositivo facturador y el dinero ingresado en el cuadro Ingreso Efectivo.
- Tablas de Transacciones.- Detalla todas las transacciones en cheque o tarjeta de crédito.

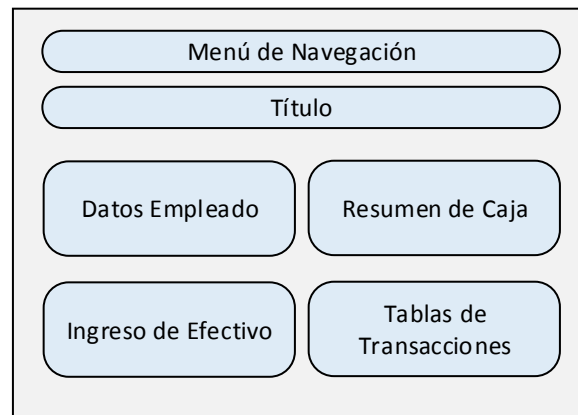


Figura 52. Esquema 3

4. Esquema 4

Esta plantilla (Figura 53) es usada para el diseño de las pantallas de Reportes. Cuenta con el Menú de Navegación, en el cual se selecciona que reporte se ha de desplegar en el cuadro inferior Reporte en objeto PDF, en el mismo se da una vista previa del reporte solicitado y se presentan opciones para guardar o imprimir.

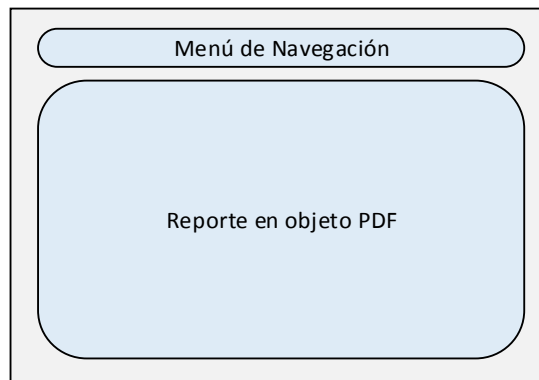


Figura 53. Esquema 4

5.1.4 Diagramas de Flujo

5.1.4.1 Diagrama de Flujo de la aplicación Administrador de Datos

En la Figura 54 se muestra el diagrama de flujo de la aplicación Administrador de Datos, la cual no posee una interfaz gráfica y se encarga del flujo de comunicación entre el dispositivo facturador y la base de datos. El programa comienza con la declaración de variables y estableciendo la

conexión con la base de datos, seguido se inicializa el servidor utilizando el protocolo de comunicación TCP/IP y se ingresa al hilo de la aplicación. La aplicación servidor termina cuando el usuario la cierra a través del icono, que se encuentra en el área de notificación en Windows 7.

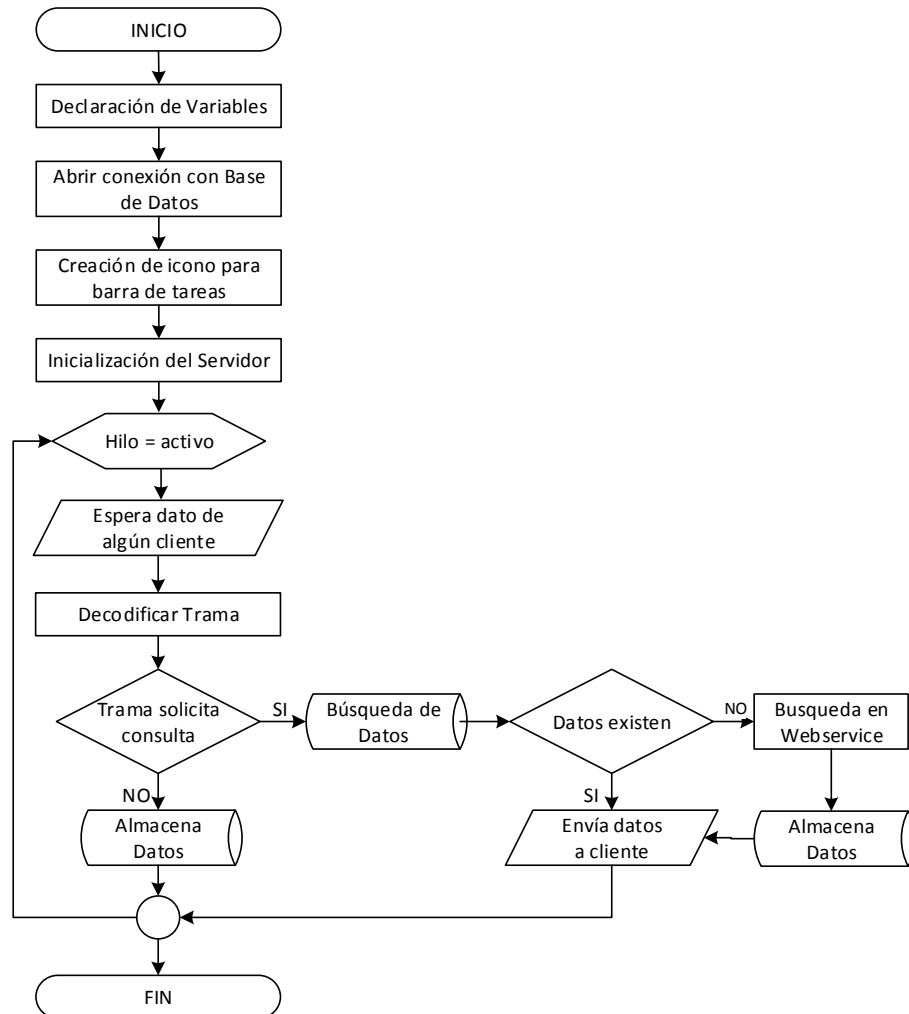


Figura 54. Diagrama de Flujo de la aplicación Administrador de Datos

Cuando la aplicación entra al hilo se coloca en un constante estado de espera de que algún cliente (dispositivos facturadores) envíe una trama, una vez recibida, la decodifica y determina si la trama es una solicitud que espera una respuesta o solamente datos a ser almacenados en la base de datos. En caso de que la trama solicite la información de algún cliente, la aplicación primero realiza una búsqueda en la base de datos, en caso de no encontrar ahí hace uso del web service para solicitar los datos del nuevo cliente y finalmente enviar la trama de respuesta. Cada vez que se hace uso del web

service se almacenan los datos devueltos en la base de datos; garantizando así, búsquedas más rápidas en futuras ventas.

El consumo del web service Merlyna se detalla a continuación:

Como ya se ha mencionado el web service utilizado en la aplicación Administrador de Datos se denomina Merlyna y la aplicación hace uso de él de forma indirecta, tal como se observa en la Figura 55. Para utilizar Merlyna se deben enviar el número de cédula a consultar seguido por una serie de parámetros que permiten la autenticación del usuario que realiza la consulta.

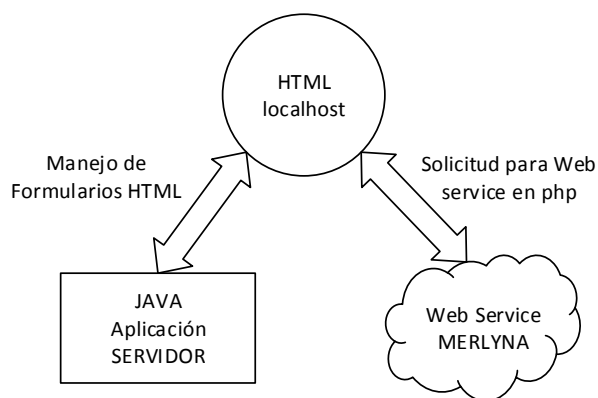


Figura 55. Consumo de Web Service

Para su implementación se crea un proyecto PHP en Netbeans. En código HTML se genera un cuadro de texto y un botón, que es donde se incrustará el código PHP para llamar al web service. Esta pequeña aplicación (Figura 56) se ejecuta en un navegador web y en el localhost, por tal motivo se debe configurar un servidor HTTP, en este caso Apache Tomcat, si bien su configuración es complicada existe un programa llamado XAMPP que lo configura en su totalidad y se lo obtiene gratuitamente de la página <https://www.apachefriends.org/es/>.

cedula

Figura 56. Aplicación PHP

De esta forma, la aplicación PHP ya permite usar el web service cada vez que se pulsa el botón “Enviar” y en la aplicación Servidor se crea un método llamado “consumir_webservice” que permite el manejo de páginas HTML. Finalmente, para realizar una consulta la aplicación Servidor llena automáticamente los campos en la aplicación PHP y obtiene la información que retorna la página, haciendo uso así del web service de forma indirecta. El código va adjunto en el *Anexo C.1*.

5.1.4.2 Diagrama de Flujo de la aplicación Cliente-Servidor

En la Figura 57 se muestra el diagrama de flujo del programa de la aplicación Cliente-Servidor, el programa empieza declarando variables del programa y abriendo la conexión con la base de datos, a continuación se muestra en qué orden aparecen las distintas pantallas de la aplicación, que se las ha enumerado, para facilitar la comprensión del diagrama de flujo, de la siguiente forma:

1. Cierre de Caja.
2. Reportes.
3. Ayuda.
4. Anular Factura.
5. Nuevo Cliente.
6. Editar Cliente.
7. Configuración General.
8. Configuración Empleados.

Dentro del bucle se da a entender que el usuario del sistema puede desplegar cualquier pantalla siempre y cuando su nivel de acceso lo permita. Además, se muestra que acción realiza la aplicación en la base de datos dependiendo de la pantalla que se encuentre. El programa termina cuando se selecciona el botón cerrar de la aplicación.

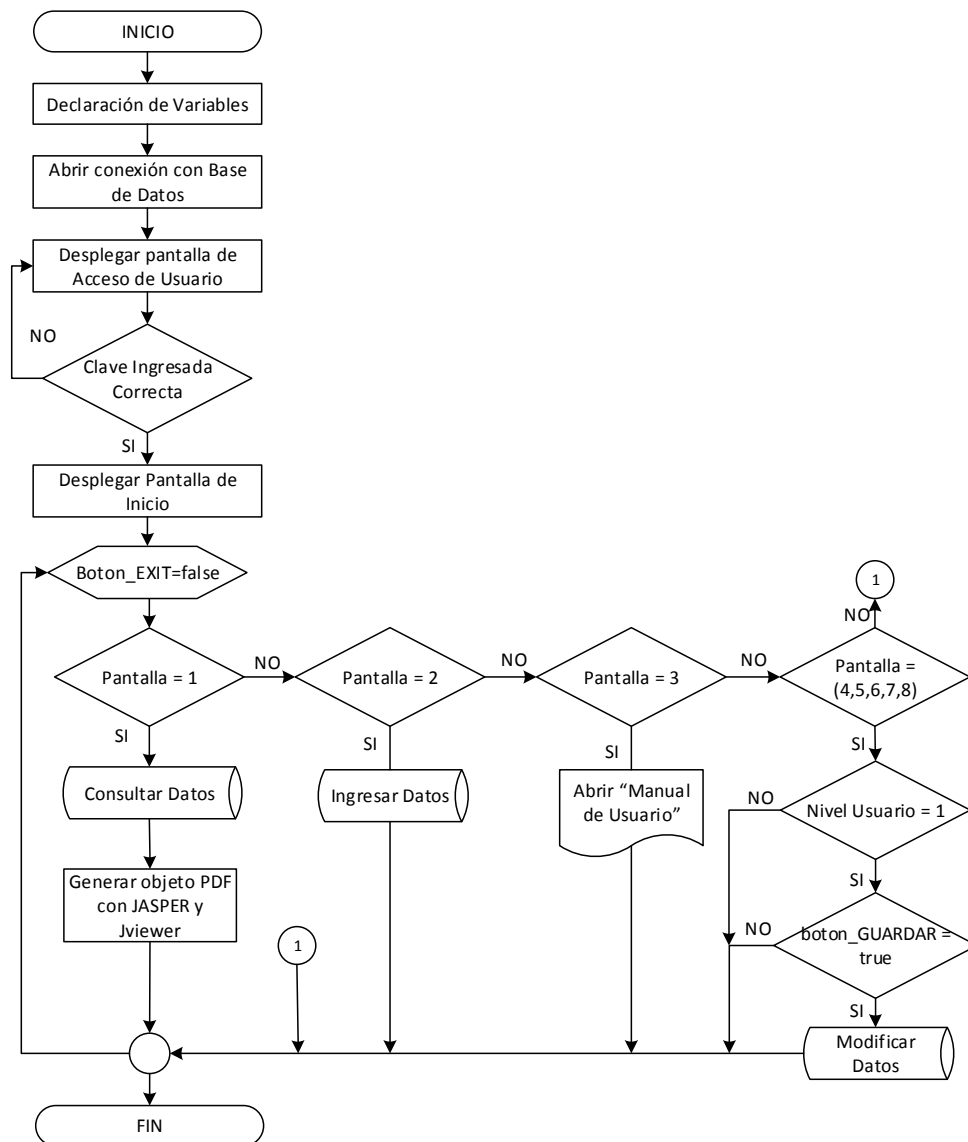


Figura 57. Diagrama de Flujo de la aplicación Cliente-Servidor

Para la elaboración de reportes se utilizó el programa iReport que permite hacer macros para consultas de base de datos; de esta forma, se genera un macro para cada tipo de reporte que se necesita desplegar. En la aplicación Administración se utiliza la librería JasperReport para generar objetos del mismo nombre y que conectan los macros ya realizados con la aplicación, para su visualización se utiliza objetos del tipo JRViewer, que se encuentran en la misma librería JasperReport, para generar una vista previa en formato pdf del reporte requerido; garantizando así, que el usuario guarde solo los reportes que él quiera. El código va adjunto en el *Anexo C.2*.

5.1.5 Diagramas de Clases

5.1.5.1 Diagrama de Clases de la aplicación Administrador de Datos

En la Figura 58 se observa el diagrama de clases de la aplicación Administrador de Datos constituido por 5 clases que se explican a continuación:

- **Clase Comprobante.** Esta clase genera los modelos de impresión tanto de facturas como de pagarés. Posee 2 métodos, el primero “buscar” determina si la impresora seleccionada para la impresión está disponible, el segundo “print” es el método de la interfaz “Printable” que debe implementarse.
- **Clase Base_Datos.** Esta clase gestiona toda la comunicación con la base de datos desde su conexión hasta las búsquedas, ingresos y modificaciones de datos; cada una de estas acciones tiene su método asociado. Además, en esta clase se declara e implementa el método consumir_webservice.
- **Clase Servidor.** En esta clase se implementa la comunicación TCP/IP del sistema de facturación, se extiende de la clase Thread y por lo tanto posee el método “run” necesario para la implementación del hilo de la aplicación. Los atributos de esta clase, permiten decodificar las tramas de datos que le llegan, así como llamar a los métodos de los objetos de las clases Base_Datos y Comprobante.
- **Clase HMI_Servidor.** Aunque esta clase no posee una interfaz gráfica, es necesaria para implementar la aplicación Servidor en segundo plano, por lo tanto hereda las propiedades de un JFrame y posee objetos de las clases PopUpMenu, TrayIcon y MenuItem. Además, se declara un objeto tipo Thread pero con el constructor de la clase Servidor. Posee dos métodos que le permiten seleccionar la imagen que va en el icono y desplegar un pequeño menú para cerrar la aplicación.
- **Clase Principal.** No posee ningún atributo y solo posee el método estático “main” que es el que ejecuta la aplicación Servidor cuando arranca, en este método se declara un objeto de tipo HMI_Servidor.

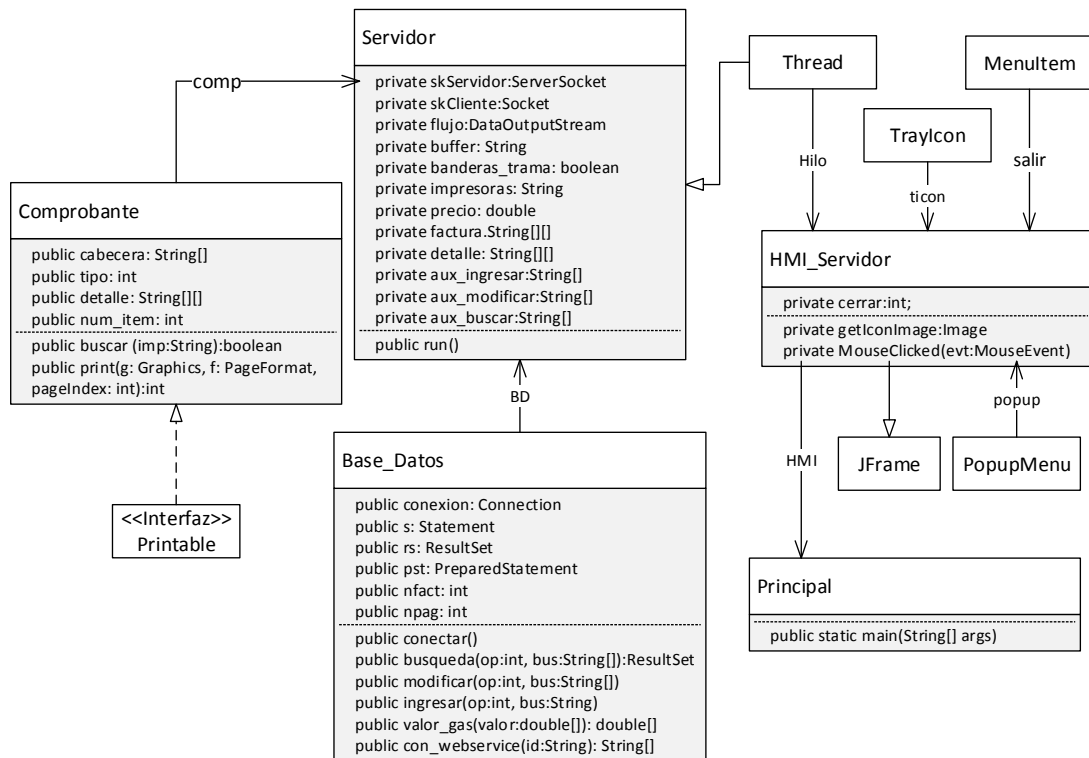


Figura 58. Diagrama de clases de la aplicación Administrador de Datos.

5.1.5.2 Diagrama de Clases de la aplicación Cliente-Servidor

Por otro lado, en el diagrama de clases de la aplicación Cliente-Servidor, *Figura 59*, no se detalla la clase `Base_Datos` debido a que es la misma que en la aplicación Servidor. A continuación se explican las 3 restantes:

- **Clase Acceso.** Esta clase se extiende de un `JFrame` y despliega la pantalla Acceso de Usuario, posee un método asociado a su objeto de tipo `JButton` que permite la validación del usuario del sistema.
- **Clase HMI_Administración.** Esta clase implementa la interfaz gráfica de la aplicación Administración y se compone de muchos tipos de objetos, tal como se observa en el diagrama, se ha preferido por no colocar los nombres de dichos objetos debido a su gran número y porque entorpecería la lectura del diagrama. Cada pantalla que se despliega es un objeto de tipo `JPanel` que contiene una combinación de los otros tipos de objetos con el fin de cumplir el esquema de la pantalla. El resto de atributos son utilizados para implementar los métodos del objeto tipo `Base_Datos` que se declara en esta clase. Los métodos de

la clase pertenecen a los objetos gráficos y tampoco se han colocado todos, en vez de esto, sólo se ha especificado que tipo de métodos ocupan estos objetos gráficos.

- **Clase Principal.** Funciona igual que en la aplicación Servidor, con la diferencia de que en el método “main” se declara un objeto del tipo HMI_Administracion.

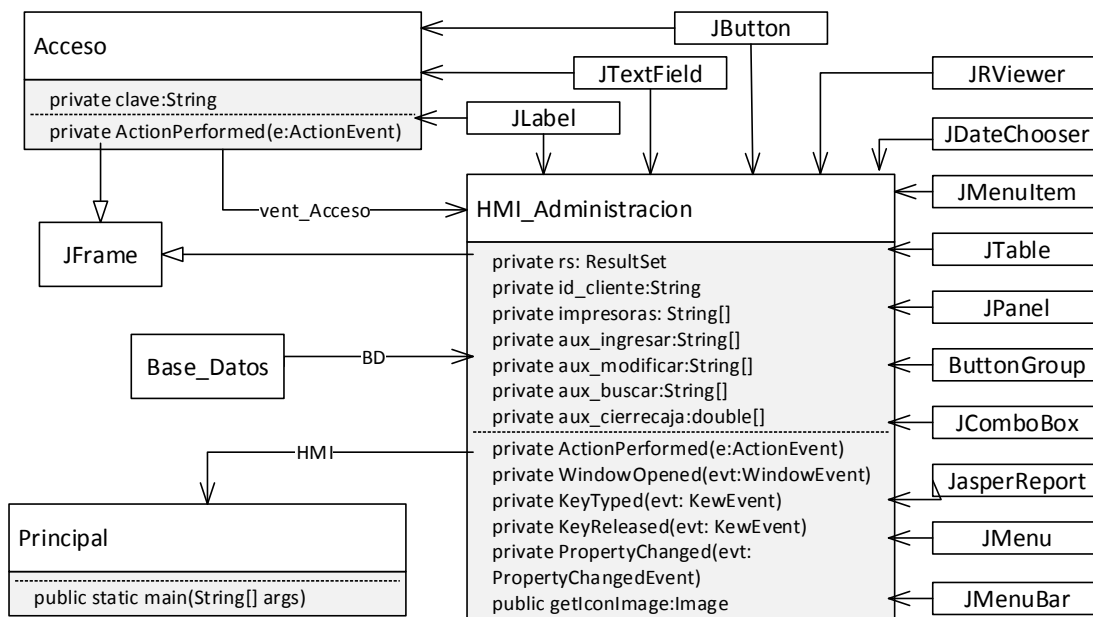


Figura 59. Diagrama de clases de la aplicación Administración.

5.2 DISEÑO DE LA BASE DE DATOS

El diseño de la base de datos del Sistema de Facturación se basa en el modelo entidad-relación que permite optimizar las búsquedas y el espacio de la misma en el disco duro. Como gestor de la base de datos se elige PostgreSQL en su versión 9.2 por ser open-source y para el diseño de la misma se elige el software Power Designer en su versión 16, aunque no es gratuito facilita el diseño de la base de datos y permite pasarlo a PostgreSQL.

5.2.1 Diagrama de Casos de Uso.

El diagrama de casos de uso del Sistema de Facturación, Figura 60, permite visualizar el comportamiento del sistema, así como identificar que actores se ven involucrados en el mismo. Se observan dos actores: un

operador y un administrador, el operador solo puede realizar las acciones de generar comprobante y cierre de caja y para cualquiera de ellas primero debe ingresar una clave de autorización; por otro lado, el administrador puede realizar todas las acciones involucradas en acción Gestionar Base de Datos y al igual que el operador debe primero ingresar su clave para autenticarse. Todas las acciones que se observan en el diagrama deben plasmarse en la base de datos del sistema.

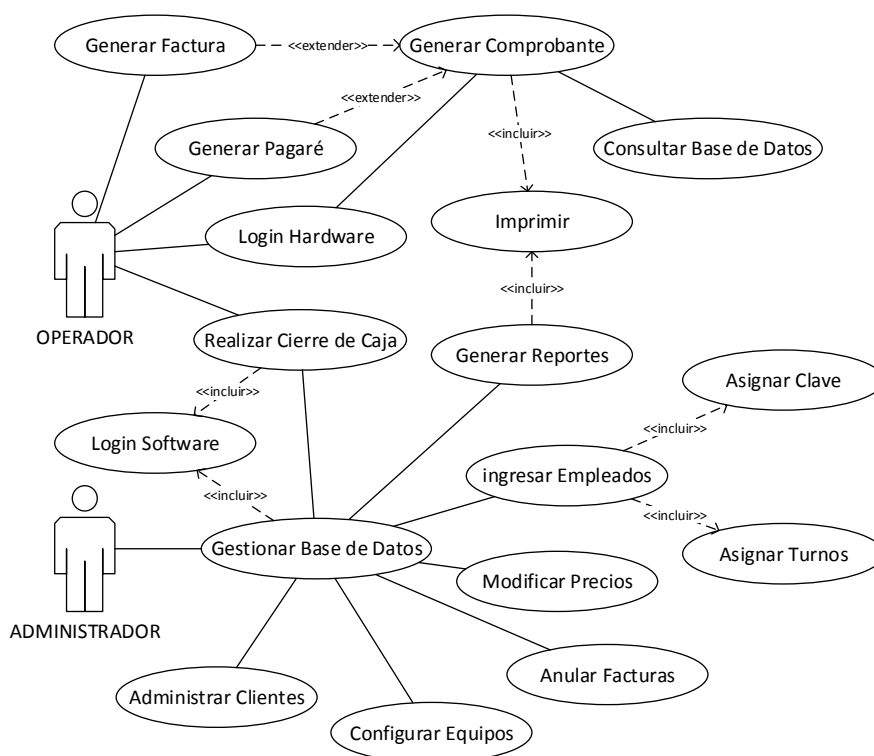


Figura 60. Diagrama de Casos de Uso.

5.2.2 Diagrama de Clases

El diagrama de clases permite describir la estructura del sistema a través de una metodología orientada a objetos. Todas las actividades y relaciones mostradas en el diagrama de casos de uso sirven de guía para estructurar el diagrama de clases de la base de datos (*Figura 61*), el cual ya muestra un modelo entidad-relación porque al ser un sistema pequeño cada clase se convierte en una tabla para la base de datos. En el diagrama se señala la

multiplicidad entre clases, así como su asociación por composición o agregación, lo cual resulta útil para el modelo lógico de la base de datos.

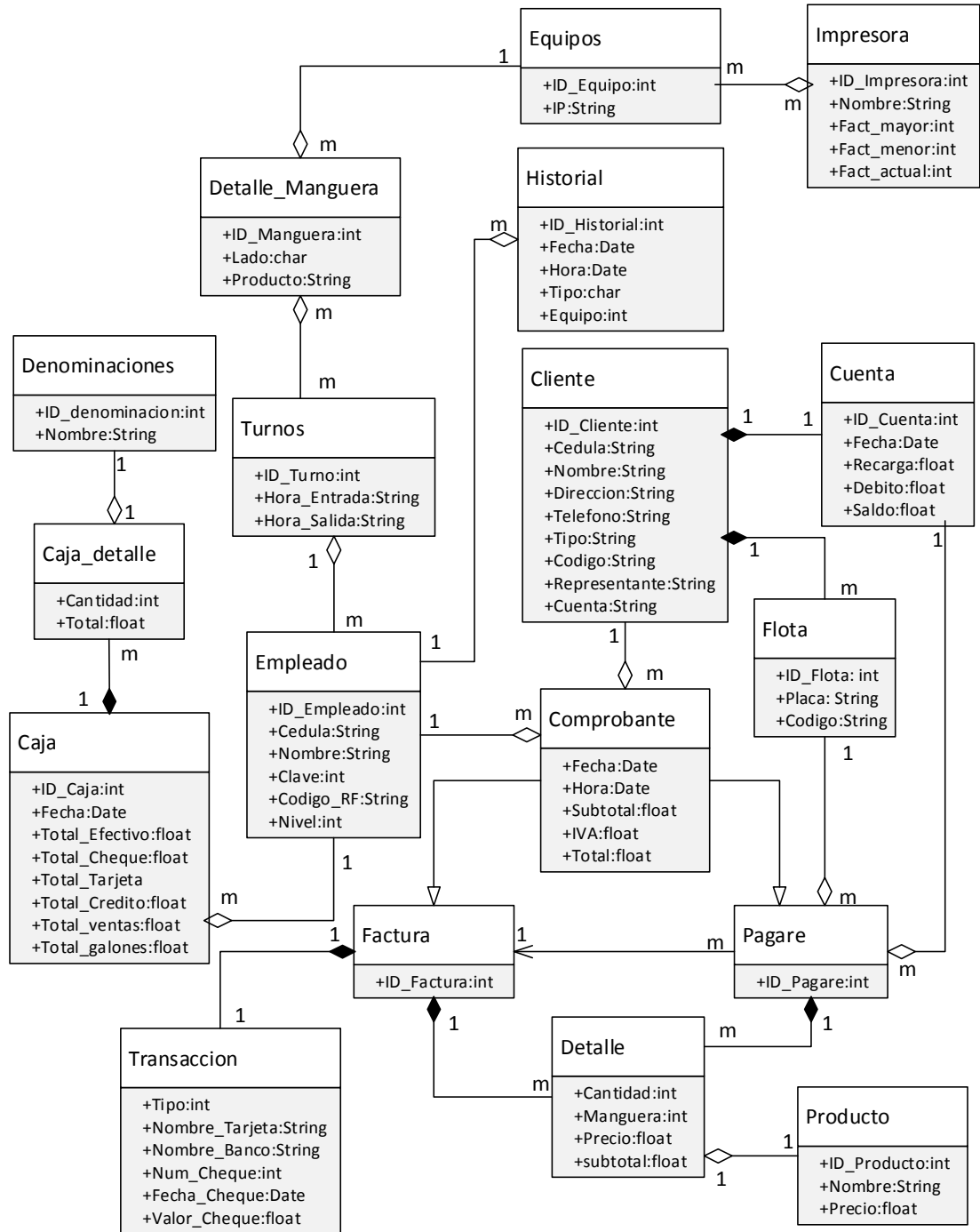


Figura 61. Diagrama de Clases de la Base de Datos

5.2.3 Modelo Lógico y Físico de la Base de Datos

El modelo lógico de la base de datos (Figura 62) parte esencialmente del diagrama de clases. En él se definen las entidades que corresponden a las clases del diagrama de clases, además se definen los atributos de cada entidad, su clave primaria y la relación entre cada entidad.

En el diagrama de clases se ven relaciones de muchos a muchos, esto solo es posible en un modelo conceptual mas no en un modelo lógico; para lograr este tipo de relación se añade una entidad intermedia que posee como atributos las claves primarias de las dos entidades que se desea relacionar. Además se agregan tres entidades más que se relacionan con la entidad Transacción; esto permite reemplazar atributos como Nombre_Banco, Nombre_Tarjeta y Tipo por claves primarias para ahorrar espacio en el disco duro.

El software Power Designer, donde se diseña el modelo lógico de la base de datos, posee una herramienta que permite pasar del modelo lógico al modelo físico automáticamente, y una vez en el modelo físico se puede generar el script para la creación de la base de datos en PostgreSQL.

En el modelo físico de la base de datos (Figura 63), las entidades pasan a ser tablas y tanto los atributos como las relaciones se ajustan de acuerdo al gestor de base de datos que se ocupa, en este caso PostgreSQL. El modelo luce más ordenado y ya no se muestran el tipo de dato de cada atributo, ni las terminaciones de las relaciones que permiten su diferenciación como en el modelo lógico.

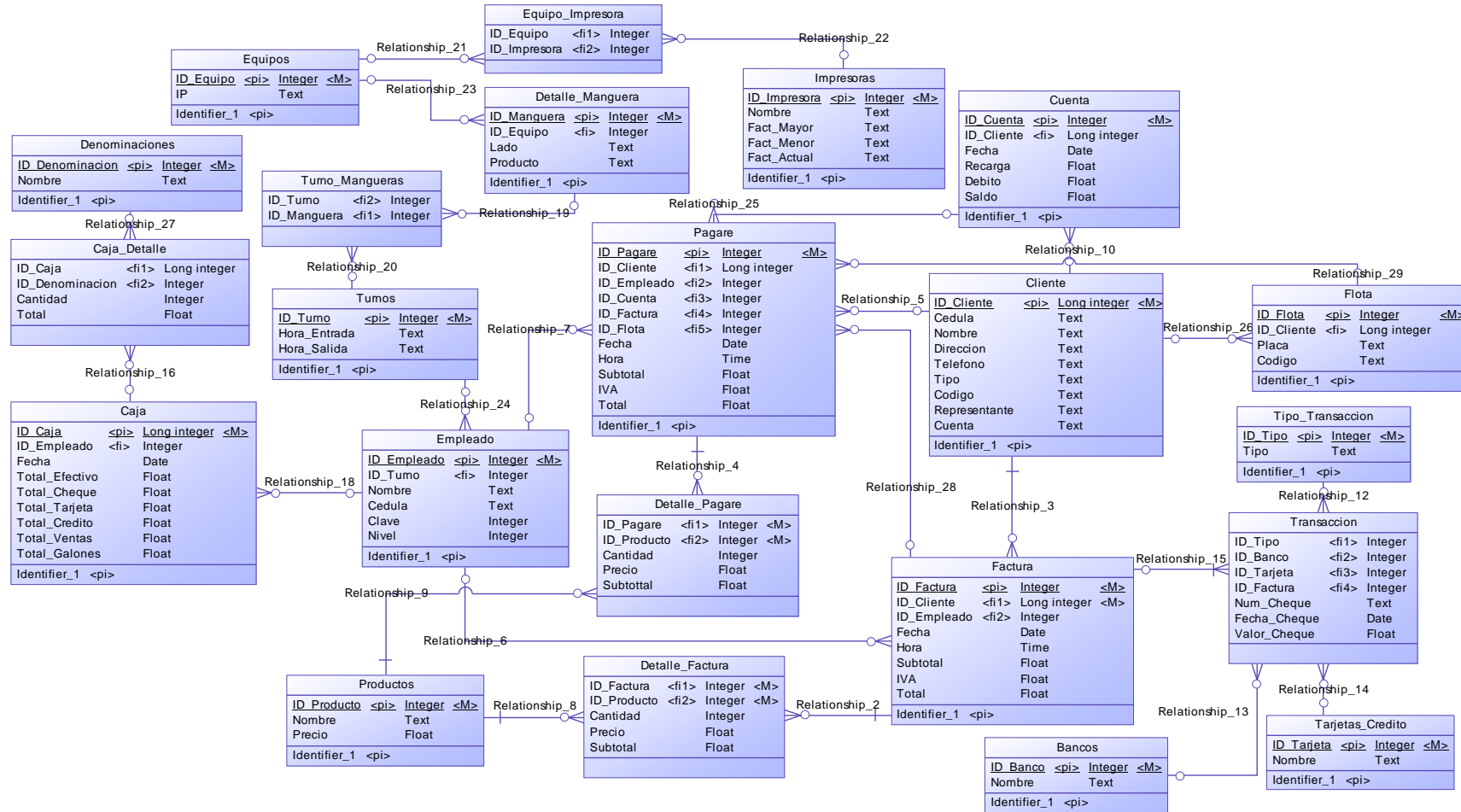


Figura 62. Modelo Lógico de la Base de Datos

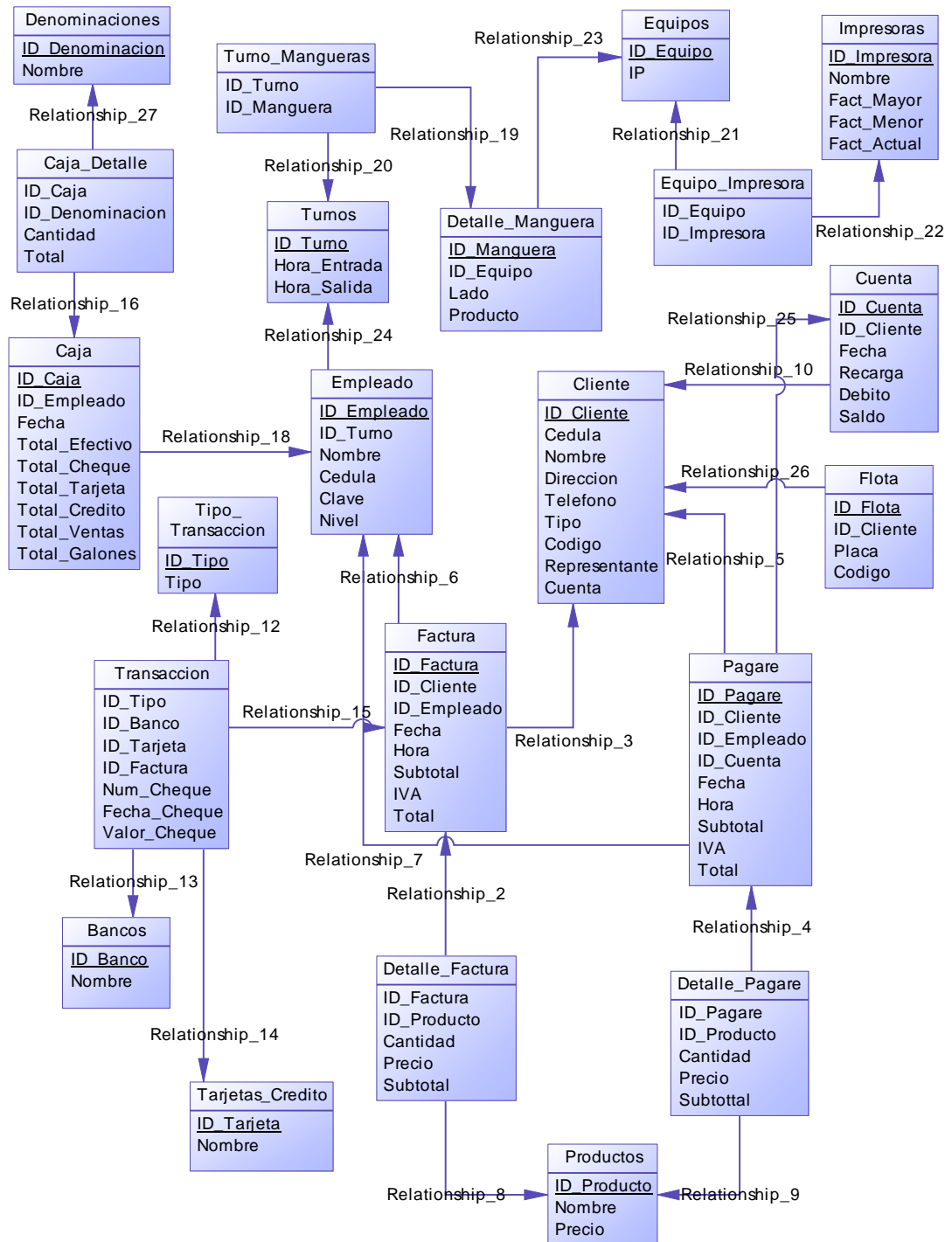


Figura 63. Modelo Físico de la Base de Datos

5.2.4 Diccionario de la Base de Datos

Este diccionario es creado automáticamente con Power Designer y se encuentra adjunto en el *Anexo H*.

5.3 REQUISITOS PARA LA INSTALACIÓN DE LAS APLICACIONES

Como se observa en el desarrollo del presente capítulo el software del sistema de facturación se compone de 2 aplicaciones: Administrador de Datos y Cliente-Servidor. Para ambas aplicaciones se generan instaladores con el software InnoSetup versión 5.0, estos instaladores contienen tanto el ejecutable .jar de cada aplicación, transformado en .exe con el programa launch4j, así como sus librerías. En el *Anexo G* se indica el manual de instalación para ambas aplicaciones; a continuación se muestra los requerimientos de cada una de ellas.

5.3.1 Aplicación Administrador de Datos

Los requisitos mínimos, tanto en software como en hardware, necesarios para la instalación de la aplicación Administrador de Datos son:

1. Software

- Sistema Operativo Windows 7
- Software XAMPP, instala y configura el servidor HTTP Apache Tomcat.
- Gestor de Base de Datos PostgreSQL versión 9.2
- Java Runtime Environment versión 1.7.0_40
- Instalador Aplicación Administrador de Datos
- Aplicación PHP para consumir web service Merlyna. Se encuentra en la carpeta llamada "Aplicación_PHP".
- Archivo Base de datos llamado demo.backup.

2. Hardware

Una computadora con las siguientes características mínimas:

- Disco duro de 1 TB
- Memoria RAM 4 GB
- Lector de CD/DVD
- 2 lectoras USB

- Tarjeta de Red de 10/100 Mb

5.3.2 Aplicación Cliente-Servidor

Los requisitos mínimos, tanto en software como en hardware, necesarios para la instalación de la aplicación Cliente-Servidor son:

1. Software

- Sistema Operativo Windows 7
- Java Runtime Environment versión 1.7.0_40
- Microsoft Excel 2010
- Instalador de la aplicación Cliente-Servidor.

2. Requisitos Hardware

Una computadora con las siguientes características mínimas:

- Disco duro de 200 GB
- Memoria RAM 2 GB
- Lector de CD/DVD
- 2 lectoras USB
- Tarjeta de Red de 10/100 Mb

El manual de Usuario de la interfaz gráfica va adjunto en el *Anexo E*.

5.4 COMPARACIÓN DE FUNCIONES DEL SISTEMA DE FACTURACIÓN DESARROLLADO CON SISTEMAS COMERCIALES

Para concluir el presente capítulo, se realiza una comparación de funciones del sistema desarrollado con sistemas comerciales que se encuentran en nuestro país perteneciente a las empresas SOLINTECE Y PYS, como se muestra en la *Tabla 18*. Se elige estas empresas debido a que

lideran el mercado nacional en el tema de sistemas de facturación para estaciones de servicio.

Considerando las características citadas en la Tabla 16, se concluye que el sistema posee cualidades que lo ponen a competir dentro del mercado nacional como una solución básica para el tema de facturación para estaciones de servicio.

Tabla 16. Comparación sistema desarrollado con otras marcas comerciales.

Característica	Sistema Desarrollado	Solución SOLINTECE	Solución PYS
Cantidad máxima de surtidores de combustible controlados	Hasta 4 comprobados.	Hasta 32	Hasta 8
Velocidad de registros e impresión automática de facturas preimpresas.	Si	Si	Si
Registro automático en el sistema, de carga de combustible	No	Si	Si
Control y programación de turnos	Si	Si	Si
Desglose de ventas, por turno, empelado y surtidor.	Si	Si	Si
Comunicación directa con el protocolo del servidor	No	Si	Si
Permite realizar la venta a la puerta del vehículo	Si	No	No
Registro de clientes, ventas y crédito en una base de datos	Si	Si	Si
Ingreso manual alfanumérico de información	Si	Si limitado	Si
Selección de 2 o más productos en la misma factura	Si	No	Si
Selección de forma de pago, efectivo, tarjeta y cheque con datos.	Si	Si	No
Administración de Clientes	Si	Si	Si
Permite exportar datos	Si	Si	Si

CAPÍTULO 6

PRUEBAS Y ANÁLISIS DE RESULTADOS

Una vez terminado el desarrollo del sistema de facturación en pista para estaciones de servicio, se realizan las pruebas de funcionamiento y el análisis de resultados, los cuales se encuentran detallados en este capítulo. Estas pruebas permiten verificar la no existencia de errores de software o hardware, que pueden presentarse durante el proceso de facturación

Las pruebas se realizan en la estación de servicio ESNAL 2 ubicada en Tandapi, km 42 vía Santo Domingo de los Tsachiles, estación que cuenta con: 3 dispensadores de gasolina, un servidor, instalación de red y demás trabajos de obra civil necesarios para la instalación del sistema.

En la estación de servicio se realiza la siguiente instalación correspondiente al sistema de facturación en pista:

- 3 dispositivos facturadores en pista
- 3 módulos de comunicación
- 2 impresoras matriciales
- El software del sistema de facturación en el servidor.

Dentro de las pruebas de funcionamiento se determina una evaluación de los siguientes puntos:

- Conectividad de los módulos de comunicación
- Impresión de facturas válidas
- Uso de Tarjetas RFID
- Tiempos de Facturación

6.1 CONECTIVIDAD DE LOS MÓDULOS DE COMUNICACIÓN A LA RED

La estación cuenta con tres módulos de comunicación conectados a un switch, al que también se conecta el servidor del sistema; además, cada módulo de comunicación cuenta con una dirección IP perteneciente a la red local de la estación. La prueba se realiza con un PING desde el servidor a la dirección IP de cada módulo, enviando 4 paquetes de tamaño de 32 Bytes y esperando a que estos sean devueltos.

En la pantalla de símbolo de sistema CMD, se utiliza el comando PING para la comprobación de conectividad, este comando constituye una de las herramientas más sencillas para detectar errores o conflictos en una red.

La dirección IP, así como la prueba que se realiza a cada módulo se muestra a continuación:

- Módulo 1: dirección IP 192.168.1.36, la ejecución del comando PING para este módulo se lo observa en la Figura 64.
- Módulo 2: dirección IP 192.168.1.37, la ejecución del comando PING para este módulo se lo observa en la Figura 65.
- Módulo 3: dirección IP 192.168.1.38, la ejecución del comando PING para este módulo se lo observa en la Figura 66.

```
C:\Documents and Settings\Esnall1>ping 192.168.1.36
Haciendo ping a 192.168.1.36 con 32 bytes de datos:
Respuesta desde 192.168.1.36: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.36: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.36: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.1.36: bytes=32 tiempo=1ms TTL=128

Estadísticas de ping para 192.168.1.36:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

Figura 64. Prueba de estado de comunicación módulo 1

```

C:\Documents and Settings\Esnall1>ping 192.168.1.37
Haciendo ping a 192.168.1.37 con 32 bytes de datos:
Respuesta desde 192.168.1.37: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.37: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.1.37: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.1.37: bytes=32 tiempo=1ms TTL=128

Estadísticas de ping para 192.168.1.37:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 1ms, Media = 0ms

```

Figura 65. Prueba de estado de comunicación módulo 2

```

C:\Documents and Settings\Esnall1>ping 192.168.1.38
Haciendo ping a 192.168.1.38 con 32 bytes de datos:
Respuesta desde 192.168.1.38: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.38: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.1.38: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.1.38: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.1.38:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 1ms, Media = 0ms

```

Figura 66. Prueba de estado de comunicación módulo 3

Análisis:

En cada una de las figuras anteriores que corresponden a las pruebas del comando PING; se observa, que el resultado para los tres módulos es el mismo, el tiempo de vida de los paquetes (TTL) es un valor óptimo igual a 128 y los 4 paquetes de datos son devueltos en un tiempo inferior a 1 nano segundo, debido a esto se concluye que no existen conflictos ni errores de red y se verifica la correcta conectividad de los equipos.

6.2 IMPRESIÓN DE FACTURAS VÁLIDAS.

El proceso de facturación termina cuando son impresas dos facturas: una para ser entregada al cliente y una para la estación de servicio. Estos documentos son necesarios para la declaración del impuesto a la renta y en el caso de la estación de servicio también para evitar sanciones por parte del SRI; por tal motivo deben ser llenados correctamente.

Para la prueba se considera una muestra de clientes equivalente a un mes de trabajo después de instalado el sistema, y se analiza para cada cliente que la factura cumpla con las siguientes características: sea emitida, que no tenga

errores, y que los datos mostrados correspondan a la información generada por el cliente.

En la estación donde se realiza las pruebas el promedio mensual de facturas emitidas es de 3900 y se obtiene los resultados mostrados en la Figura 67.

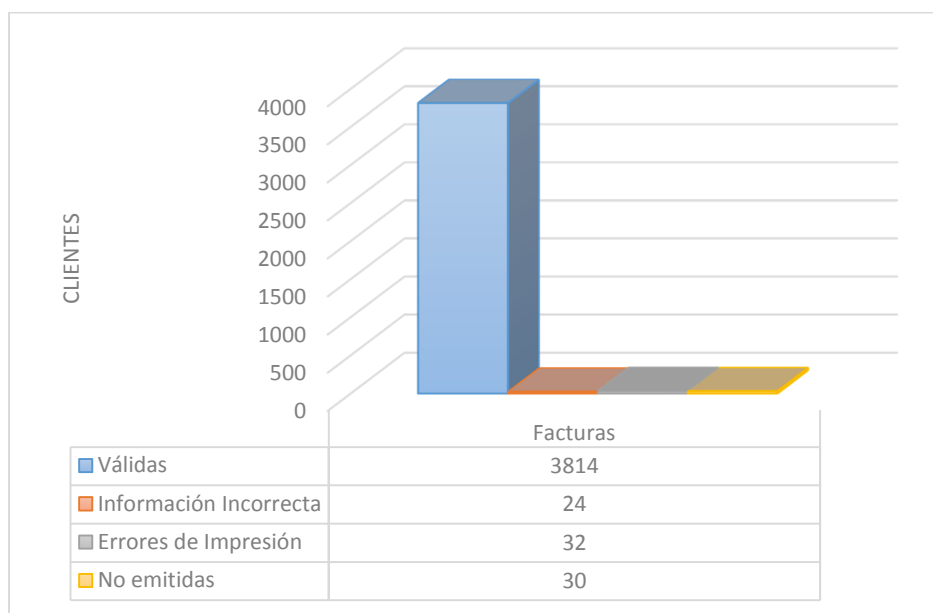


Figura 67. Facturación de un día promedio de trabajo

Análisis:

- Existe un total de 3814 (97.8%) facturas que fueron emitidas correctamente, la muestra incluye clientes que piden más de un producto y su forma de pago varía entre las tres opciones (efectivo, tarjeta, cheque).
- Existe un total de 24 (0.61%) facturas con información incorrecta, debido a un mal ingreso por parte del usuario del sistema, al no verificar la información digitada o dejarla incompleta registrados en su mayoría el primer día de uso del sistema.
- Existe un total de 32 (0.82%) facturas con errores de impresión, presentando el inconveniente de que ocupan más del espacio configurado al momento de imprimirse. El error se da debido a una desconfiguración de los print server conectados a las impresoras para darles acceso a la red después de un reinicio del sistema.

- Existe un total de 30 (0.77%) facturas que no fueron emitidas, debido a una falla de la red eléctrica de la estación se informó que el inconveniente se repitió en 4 oportunidades, teniendo que durante el lapso sin energía facturar a mano.

Finalmente, la prueba da como resultado que no existen errores por mal funcionamiento del hardware o software diseñado, y que los errores encontrados son debido a las siguientes razones: problemas de conexión debido a los equipos de conectividad de red, mala ejecución por parte del usuario del sistema, y por problemas en el sistema eléctrico de la estación.

6.3 MÓDULO RFID

El uso de tarjetas o llaveros RFID (etiquetas pasivas) al presentarse como una alternativa más eficiente con respecto al ingreso manual de información, debe permitir al usuario del sistema realizar el proceso de facturación de una forma ágil y empleando una menor cantidad de tiempo. Para verificar que lo anterior se cumpla se realizan las siguientes pruebas:

- Tiempo de respuesta para usuarios del sistema y clientes comparado con el tiempo de ingreso manual. Se toma en cuenta desde que la tarjeta es acercada al dispositivo facturador, hasta que se muestra el mensaje en pantalla de confirmación de lectura. En la Tabla 17 se observa los resultados.
- Cantidad de lecturas válidas versus no válidas al primer intento y cantidad de tarjetas no registradas validadas por el sistema, para lo cual se registran 20 códigos en el servidor; correspondientes a 10 tarjetas de usuario del sistema, y a 10 tarjetas de clientes; y se mantienen 10 tarjetas sin registrar. La *Figura 68* muestra los resultados de la prueba.

Tabla 17. Resultados prueba de funcionamiento módulo RFID

	Tiempo de respuesta (segundos)	Cantidad de ingresos al mes	Tiempo total mensual (segundos)
Tarjetas RFID			
Usuarios del sistema	4	60	240
Clientes	6	30	180
Ingreso Manual			
Usuarios del sistema	13	60	780
Clientes	16	30	480

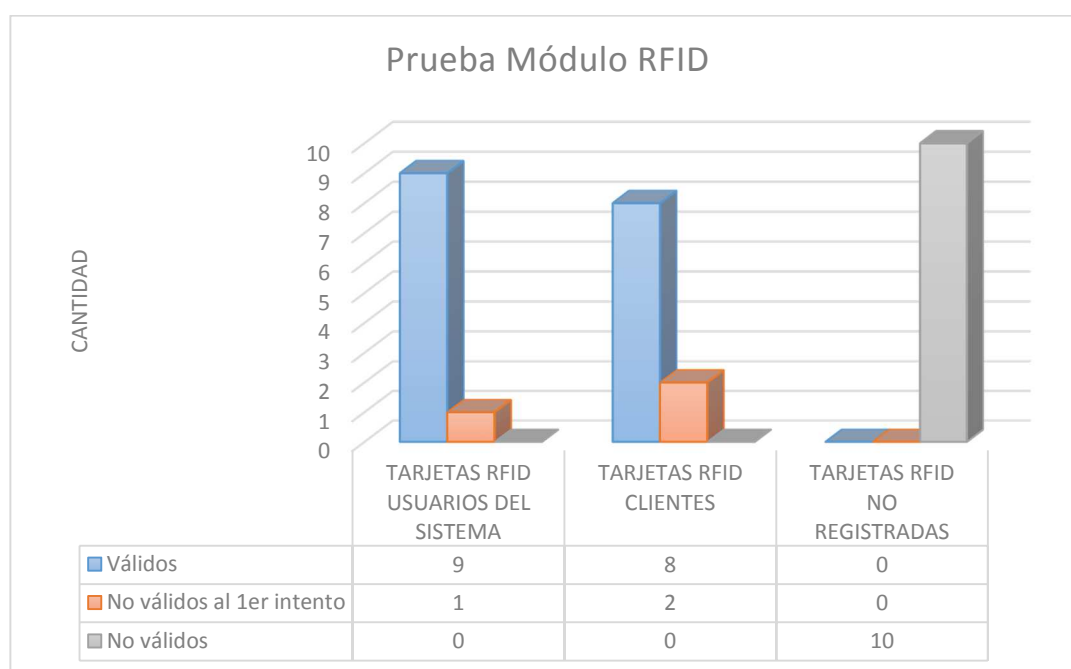


Figura 68. Resultado prueba de reconocimiento de los dispositivos.

Análisis:

Para el primer caso se considera el tiempo de respuesta del servidor multiplicado por la cantidad de uso al mes de las tarjetas RFID, se considera para usuarios del sistema 2 turnos durante 30 días y para clientes 1 compra diaria durante todo el mes; la misma cantidad de usos se multiplica por el tiempo que toma realizar el ingreso manual obteniendo los siguientes resultados:

- Tiempo total empleado al mes con tarjetas RFID = 420 seg. (7 minutos)
- Tiempo total empleado al mes con ingreso manual= 1260 seg. (21 minutos)

Lo que demuestra que el uso de tarjetas RFID es 3 veces más eficiente que el método de ingreso manual.

El segundo caso permite comprobar la seguridad del sistema, en la *Figura 68*, se muestra que si bien existen tarjetas no validadas a la primera lectura la mayoría, que representa el 85% de casos, paso la prueba sin problemas y el 100% de tarjetas no registradas no fueron validadas; por consiguiente, el sistema no da paso a tarjetas que no hayan sido registradas previamente evitando conflictos por uso de tarjetas ajenas al sistema.

6.4 TIEMPO DE FACTURACIÓN

La prueba final constituye el tiempo que lleva realizar el proceso de facturación con el sistema comparado con el ingreso manual tradicional y se la realiza en dos días para una muestra de 10 clientes. En el primer día se toma el tiempo que tarda realizar cada factura de la forma tradicional y en el segundo día se lo realiza con el sistema instalado. Al concluir la prueba se obtiene los resultados mostrados en la *Figura 69*.

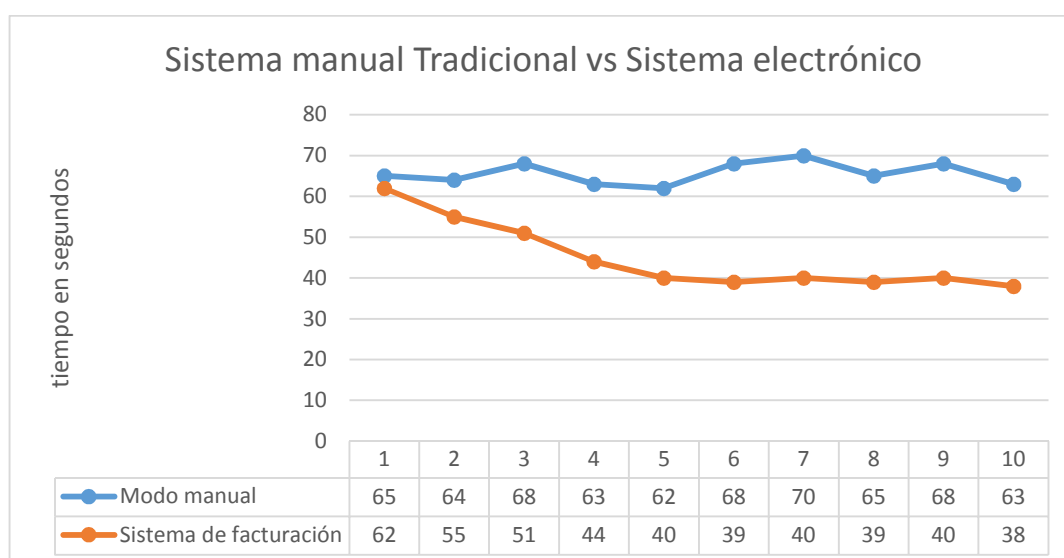


Figura 69. Comparación de tiempo de facturación Ingreso manual vs. Sistema de facturación instalado.

Análisis:

Con los resultados obtenidos en la *Figura 69* se concluye lo siguiente:

- El tiempo promedio que lleva hacer una factura de la forma manual tradicional es superior al tiempo promedio que lleva hacerlo con el sistema de facturación desarrollado, lo que provoca un mayor tiempo en el proceso de despacho de combustible y al final del día un menor número de clientes atendidos.
- La facturación en un principio con el sistema desarrollado puede tomar el mismo tiempo que si se lo realiza manualmente, pero conforme el usuario del sistema se vaya familiarizando con el dispositivo el tiempo que le lleva realizar una factura mejora, a diferencia de la forma tradicional donde el tiempo depende de la velocidad de escritura del operador, de los datos del cliente y de la capacidad de calcular el monto final tomando en cuenta el IVA.

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

- El sistema de facturación desarrollado cumple con los objetivos planteados de diseño e implementación de hardware y software; además, el uso de la plataforma Arduino, tecnología RFID y base de datos lo convierten en un sistema que puede ser instalado en estaciones de servicio como una solución a sus necesidades de facturación en pista.
- La utilización de la norma IPC-2221 para el diseño de PCBs, permite determinar el ancho de las pistas, en base a la corriente máxima que las atraviesa, y las condiciones ambientales existentes en el lugar donde la tarjeta vaya a ser instalada.
- El LCD gráfico con resolución de 128 x 64 pixeles proporciona gran flexibilidad para mostrar texto, gráficos o ambos, y gracias a su tamaño y retroiluminación permite la visualización de los mensajes e imágenes de forma clara y legible; siendo de gran ayuda para los usuarios del sistema tanto en el día como en la noche.
- Para la fabricación de la carcasa se contrata un servicio de impresión 3D, solución eficiente para el desarrollo de modelos y prototipos, porque, permite realizar correcciones y pruebas sin una inversión económica muy elevada, hasta obtener el diseño final

- El diseño del módulo de comunicación se facilita gracias al uso de la plataforma Arduino, útil para la implementación de prototipos debido a su flexibilidad en software y hardware, ya que cuenta con placas que pueden ser montadas extendiendo su funcionalidad; como es el caso de la Ethernet Shield utilizada para la comunicación Ethernet; además posee librerías dedicadas para comunicación Ethernet y serial RS232 indispensables para el desarrollo del sistema.
- El uso de Java como lenguaje de programación para el desarrollo de la aplicación del servidor, permite crear una aplicación multitareas que cumpla todas las funciones necesarias para un proceso de facturación, gracias a la cantidad de librerías disponibles, nativas e independientes que permiten la implementación del protocolo TCP/IP, conexión con bases de datos, manejo de archivos pdf y manejo de páginas web.
- La implementación de tecnología RFID, gracias a su velocidad de respuesta y facilidad de manipulación de los Tags o tarjetas, reduce el tiempo dedicado por el usuario del sistema para el ingreso de información del cliente, siendo tres veces más rápido que el método de ingreso manual de acuerdo a las pruebas y el análisis realizado.
- Se optó por la comunicación serial TTL a través de un cable multipar para comunicar el dispositivo facturador y el módulo de comunicación, debido a que en estaciones de servicio está prohibido cualquier comunicación inalámbrica; además, debido a que el cable es enrollado y su longitud es de 3 metros, facilita el desplazamiento del dispositivo facturador a la cercanía del cliente y evita tropiezos o enredos con la mangueras del dispensador de combustible.
- La implementación de la base de datos con el modelo de entidad relación, permite obtener un mejor registro de las ventas realizadas en la estación de servicio, y facilita la generación de reportes, al asegurar las relaciones con entidades válidas y al evitar la duplicidad en los registros.

7.2 RECOMENDACIONES

- Para el desarrollo de cualquier proyecto se recomienda mantener un orden de diseño e implementación con la finalidad de optimizar el tiempo de ejecución; empezando, por determinar cuáles son los puntos críticos, de los cuales dependen otros subprocesos.
- Se recomienda a los usuarios del sistema, siempre que realicen el proceso de facturación, verificar los valores ingresados antes de continuar con el siguiente paso, debido a que ingresos erróneos producen fallos al momento de realizar el levantamiento de información mensual y reportes de ventas.
- Si bien, la impresión 3D es una buena solución para el desarrollo de modelos experimentales o prototipos, para la producción en serie de un producto se recomienda la fabricación de un molde y el uso de técnicas de inyección de plástico, debido a que para una producción que supere la centena de unidades resulta más económico, debido a que: se reducen costos de mano de obra, permite el uso de una gran variedad de materiales y existe un mínimo desperdicio de materia prima.
- Se recomienda dar capacitación a los usuarios del sistema, con la finalidad de que estén familiarizados con el manejo del equipo y se eviten errores de en la facturación por el mal manejo de los dispositivos.
- Se recomienda antes de proceder a la instalación del sistema de facturación en pista, primero realizar el levantamiento de información de la estación de servicio, confirmando: cantidad de surtidores en los que se va a instalar los dispositivos facturadores, cantidad de impresoras; y distancias entre equipos y hasta el cuarto del servidor.
- Para la instalación, manipulación y energización de los dispositivos facturadores, y el módulo de comunicación, se recomienda siempre contar con manuales, hojas técnicas y la colaboración de personas capacitadas en la instalación, mantenimiento y manipulación de estos equipos, porque se

trabaja cerca de combustible y su mala manipulación puede ser el detonante de algún tipo de incendio.

- Es recomendable la implementación y desarrollo de un módulo adicional, el cual permita, la obtención de datos en tiempo real de cantidad y tipo de combustible despachado directamente de los surtidores, con la finalidad de mejorar la autonomía del sistema y el tiempo en el proceso de despacho evitando el ingreso manual de los datos mencionados anteriormente.
- Para los clientes a crédito que poseen más de un vehículo registrado en el sistema, se recomienda el desarrollo de una aplicación de mensajería web, mediante la cual el cliente, dueño de la cuenta sea informado constantemente del consumo que sus vehículos van teniendo, con el fin de otorgarle un beneficio adicional al servicio prestado normalmente.
- Se recomienda el diseño y desarrollo de un módulo de comunicación dedicado solo para servir de puente entre el dispositivo facturador y la aplicación instalada en el servidor, con el fin de reducir costos, debido a que el modulo actual basado en la plataforma Arduino, posee varios otras funciones no utilizadas y por las cuales el costo del módulo se encarece.

BIBLIOGRAFÍA

- Arduino. (2 de mayo de 2014). *Arduino*. Obtenido de <http://www.arduino.cc/es/>
- Blásquez del Toro, L. M. (06 de 05 de 2014). *Sistemas de Identificación por Radiofrecuencia*. Obtenido de Universidad Carlos III de Madrid: <http://www.it.uc3m.es/jmb/RFID/rfid.pdf>
- El Protocolo Wiegand*. (06 de 05 de 2014). Obtenido de PicManía by RedRaven: http://picmania.garcia-cuervo.net/conceptos_wiegand.php
- Gallego Cano, J. (2005). *PCPI - Montaje de componentes informáticos*. EDITEX.
- GUEMISA. (05 de 05 de 2014). *Teclados de Membrana*. Obtenido de GUEMISA: <http://www.guemisa.com/tecla/docus/tec-teclados.pdf>
- Ibrahim, D. (2008). *PIC Microcontroller Projects in C: Basic to Advanced*. UK: Elsevier.
- id-innovations. (2013). *ID-2/12/20 LA Series Datasheet X1*.
- Importec. (2013). Cargador Aoc Universal 5 Voltios 2a. Quito, Pichincha, Ecuador.
- INAMHI. (2008). *ANALISIS CLIMATOLOGICO*.
- IPC. (Febrero de 1998). IPC-2221. *Generic Standard on Printed Board Design*.
- Maldonado, P. (16 de Septiembre de 2013). El SRI ahora exige número de cédula al comprar combustible. *EL COMERCIO*.
- Mandado Pérez, E., Menéndez Fuertes, L., Fernández Ferreira, L., & López Matos, E. (2007). *Microcontroladores PIC. Sistema Integrado para el autoaprendizaje*. Barcelona, España: MARCOMBO, S.A.
- Microchip. (2009). *PIC18F87J11 Family Data sheet*.
- Saffirio, M. (4 de mayo de 2014). *¿Qué son los Web Services?* Obtenido de <http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>
- Snell, J., & Tidwell, D. (2002). *Programming Web Services with SOAP*. Estados Unidos: O'Reilly&Associates.
- T. Eggeling, H. F. (2003). *Ampliar, Reparar y Configurar su PC*. Marcombo.
- Valdés Pérez, F., & Pallàs Areny, R. (2007). *Microcontroladores: Fundamentos y Aplicaciones con PIC*. España: MARCOMBO, S.A.
- Vishay. (2002). *LCD-128G064I*.

GLOSARIO

- Aba Track2** Es un protocolo de comunicación Síncrono desarrollado por la American Banking Association, utilizado para la decodificación de tarjetas magnéticas.
- Autotex
PET** Es una película de poliéster con textura de alta calidad, que consiste de una base de poliéster con un recubrimiento flexible, químicamente unido y UV-curado.
- Apache
Tomcat** Es un servidor web de código abierto con soporte de servlets y JSPs.
- Arduino** Es una plataforma electrónica para la creación de prototipos basada en software y hardware open-source.
- Ares** Herramienta para la elaboración de placas de circuito impreso que permite un posicionamiento automático de elementos y la generación de pistas.
- Ascii** (Código Estándar Estadounidense para el Intercambio de Información), es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno y en otras lenguas occidentales
- CLDH** Siglas correspondientes a Combustibles Líquidos Derivados de Hidrocarburos
- CMD** (Command Prompt - símbolo de sistema), es un intérprete de comandos en sistemas basados en Windows.
- Eclipse** Eclipse es una plataforma de desarrollo de código abierto basada en Java, es un marco de trabajo y un conjunto de servicios para la construcción del entorno de desarrollo de los componentes de entrada.

Ethernet	Es un estándar de redes de área local para computadores con acceso al medio por detección de la onda portadora y con detección de colisiones
GLCD	Constituye una pantalla gráfica de cristal líquido, formada por una matriz de píxeles monocromos colocados delante de una fuente de luz o reflectora
GND	(Ground - tierra), es el común del circuito, donde se supone que existe 0 voltios.
Hardware	Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático
HTML	Siglas de HyperText Markup Language, hace referencia al lenguaje de marcado para la elaboración de páginas web.
HTTP	Abreviatura de la forma inglesa Hypertext Transfer Protocol (protocolo de transferencia de hipertextos), que se utiliza en algunas direcciones de internet
ICSP	Significa "In Circuit Serial Programming", es un conector que disponen algunas placas mediante el cual podemos actualizar o reprogramar el chip sin sacarlo del zocalo en donde se encuentra colocado.
IDE	(Integrated Development Environment - Entorno de Desarrollo Integrado) es un programa informático compuesto por un conjunto de herramientas de programación.
Interrupción	Se generan mientras un programa se está ejecutando, y busca q detener el proceso en curso y ejecutar la función específica de quien produce la interrupción.
IPC-2221	Es una norma standard para el diseño de placas de circuitos impresos.

Jasper Report	Es una herramienta de creación de informes que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML, etc.
JAVA	Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible
JDK	Java Development Kit, es un software que provee herramientas de desarrollo para la creación de programas en Java
JRE	Java Runtime Environment, es un conjunto de utilidades que permite la ejecución de programas Java
JRViewer	Componente de JasperReports utilizado para mostrar reportes generados en la interfaz gráfica de nuestra aplicación de escritorio.
Launch4j	Permite transformar un archivo JAR en un archivo EXE con un enorme grado de flexibilidad
Layout	Es un esquema de distribución de elementos dentro un diseño.
Máquina CNC	(Control Numérico Computarizado), es un sistema de automatización de máquinas y herramienta que son operadas mediante comandos programados en un medio de almacenamiento.
Merlyna	Web service, provisto por una empresa ecuatoriana, permite obtener el nombre de personas a través de su número de cédula.
Micro-controlador	Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria

Mono-cromática	Luz que está formada por componentes de un solo color.
Netbeans	Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java.
Open-Source	Código abierto, es la expresión con la que se conoce al software distribuido y desarrollado libremente
PCB	(Printed Circuit Board – Tarjeta de Circuito Impreso), Es una superficie constituida por pistas de material conductor laminadas sobre una base no conductora
PCWHD	Es un compilador de CCS Inc, que permite generar archivos en formato Intel-hexadecimal, necesario para programas los microcontroladores (PIC).
PgAdminIII	Es una aplicación gráfica de diseño y manejo de bases de datos para su uso con PostgreSQL, siendo la más completa y popular con licencia Open Source.
PHP	Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.
PostgreSQL	Sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.
Processing	Es un lenguaje de programación basado en java usualmente utilizado para proyectos multimedia.
Proteus	Es una compilación de programas de diseño y simulación electrónica, desarrollado por Labcenter Electronics que consta de los dos programas principales: Ares e Isis.
Pull-up	En electrónica se denomina a la acción de elevar la tensión de salida de un circuito lógico, a la tensión que, por lo general mediante un divisor de tensión, se pone a la entrada

de un amplificador con el fin de desplazar su punto de trabajo

- RFID** (Radio Frequency IDentification - Identificación por Radiofrecuencia), es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID
- RS232:** (Recommended Standard 232), es una interfaz que designa una norma para el intercambio de una serie de datos binarios entre un transmisor y un receptor.
- Shields** Son tarjetas que se montan sobre las tarjetas Arduino expandiendo así su funcionalidad.
- SOAP** : (Simple Object Access Protocol), es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML
- Software** Conjunto de programas y rutinas que permiten a la computadora y sistemas electrónicos realizar determinadas tareas
- SolidWorks** Es un programa de diseño asistido por computadora para modelado mecánico desarrollado en la actualidad por SolidWorks Corp.
- SRI** (Servicio de Rentas Internas), es una entidad técnica y autónoma que tiene la responsabilidad de recaudar los tributos internos establecidos por Ley.
- STL** Es un formato de archivo informático de diseño asistido por computadora que define geometría de objetos 3D,

excluyendo información como color, texturas o propiedades físicas que sí incluyen otros formatos CAD

- Switch** O conmutador, es un dispositivo digital lógico de interconexión de redes de computadoras que opera en la capa de enlace de datos del modelo OSI.
- TAG** Son unos dispositivos pequeños, que pueden ser adheridas o incorporadas a un producto, y contienen antenas para permitirles recibir y responder a peticiones por radiofrecuencia desde un emisor-receptor RFID
- Thread** (Hilo), es una característica que permite a una aplicación realizar varias tareas a la vez.
- Trama** Es una unidad de envío de datos, es una serie sucesiva de bits, organizados en forma cíclica, que transportan información y que permiten en la recepción extraer esta información
- TTL** (transistor-transistor logic - lógica transistor a transistor). Es una familia lógica o lo que es lo mismo, una tecnología de construcción de circuitos electrónicos digitales.
- UDDI** (Universal Discovery Description and Integration), es un modelo de directorios para Web Services. Es una especificación para mantener directorios estandarizados de información acerca de los Web Services, sus capacidades, ubicación, y requerimientos en un formato reconocido universalmente.
- USART** (Universal Asynchronous Receiver-Transmitter), controla los puertos y dispositivos serie y se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo

UTP	Se denomina así al par trenzado no apantallado, uno o más pares de cable rodeados por un aislamiento
Watchdog	Es un mecanismo de seguridad que provoca un reset del sistema en caso de que éste se haya bloqueado
Wiring	Es una estructura de programación open-source para Microcontroladores. Se utiliza comúnmente en los AVR Atmega.
Wiegand26	Protocolo de comunicación, que ocupa tres líneas de transmisión: una para los unos lógicos, otra para los ceros lógicos y una última para la masa o referencia.
WSDL:	(Web Services Description Language), es un protocolo basado en XML que describe los accesos al Web Service. Es el manual de operación del web service, porque indica cuales son las interfaces que provee el Servicio web y los tipos de datos necesarios para la utilización del mismo
XAMPP	Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.
XML	(eXtensible Markup Language), es un lenguaje de marcas desarrollado por el World Wide Web Consortium utilizado para almacenar datos en forma legible.

ACTA DE ENTREGA

El proyecto de titulación “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE FACTURACIÓN EN PISTA PARA ESTACIONES DE SERVICIO” fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Universidad de las Fuerzas Armadas – ESPE, desde:

Sangolquí, _____ de 2014

ELABORADO POR:

JOSÉ JAVIER GUARNIZO POMA
110288431-5

BERNARDO JAVIER VALLEJO MANCERO
172366866-9

AUTORIDAD

Ing. Luis Orozco MSc.
DIRECTOR DE LA CARRERA DE ELÉCTRICA Y ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL