



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO

**FACULTAD DE CIENCIAS FISICAS Y
MATEMATICAS**



ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE
CONTROL PARA EL REGISTRO DE PRÉSTAMOS DE
EQUIPOS A LOS ESTUDIANTES, PARA EL
LABORATORIO DE INGENIERÍA ELECTRÓNICA DE
LA UNPRG.”**

Tesis para optar el Título Profesional de Ingeniero Electrónico

PRESENTADA POR:

Bach. CUSTODIO PISFIL JESÚS FRANCISCO

Bach. VIDARTE COTRINA ABNER MICHEL

LAMBAYEQUE – PERU 2016

**“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE
CONTROL PARA EL REGISTRO DE PRÉSTAMOS DE
EQUIPOS A LOS ESTUDIANTES, PARA EL
LABORATORIO DE INGENIERÍA ELECTRÓNICA DE
LA UNPRG.”**

Bach. CUSTODIO PISFIL JESÚS FRANCISCO

Bach. VIDARTE COTRINA ABNER MICHEL

Tesis para optar el Título Profesional de Ingeniero Electrónico



LAMBAYEQUE – PERÚ 2016

Tesis presentada por:

Bach. CUSTODIO PISFIL JESÚS FRANCISCO

Bach. VIDARTE COTRINA ABNER MICHEL

Tesis para optar el Título Profesional de Ingeniero Electrónico

Aceptado por la Escuela Profesional de Ingeniería Electrónica:

Ing. Manuel Javier Ramírez Castro

PRESIDENTE

Ing. Carlos Leonardo Oblitas Vera

VOCAL

Ing. Hugo Javier Chiclayo Padilla

SECRETARIO

Dr. Julio Ernesto Quispe Rojas

ASESOR

Bach. Jesús Francisco Custodio Pisfil

AUTOR

Bach. Abner Michel Vidarte Cotrina

AUTOR

LAMBAYEQUE – PERÚ 2016

DEDICATORIA

Dedico con amor este proyecto de tesis a mis queridos padres; mi padre Francisco por los consejos de superación que en todos momentos recibo de él, a mi madre María Eugenia por su incondicional apoyo para que pueda superarme tanto como persona, y como profesional; para ustedes va este logro en mi largo camino que trae esta vida, me siento orgulloso de los dos y espero que con mis triunfos ustedes sientan lo mismo por mí.

JESUS FRANCISCO

DEDICATORIA

A mis padres, Roberto Aureliano Vidarte Vega y María Susana Cotrina Lezama, por su indismayable esfuerzo en mi formación personal y profesional.

A mi esposa Katuska, e hija Alyson, por su apoyo incondicional y ser la motivación, para alcanzar mis objetivos.

ABNER MICHEL

AGRADECIMIENTO

Agradecemos en primer lugar Dios todo poderoso por darnos la sabiduría necesaria para poder enfrentar nuestras metas, por darnos la fuerza espiritual de continuar por el largo camino de la vida.

En segundo lugar agradecemos al Ing. Julio Ernesto Quispe Rojas por la paciencia y confianza que tuvo hacia nosotros de asesorarnos con nuestro proyecto de tesis.

Agradecemos de manera general a todos los ingenieros docentes de la Escuela Profesional de Ingeniería Electrónica por su constante orientación, motivación y apoyo recibido, a todas las personas, amigos que con una u otra forma estuvieron dándonos los ánimos de poder sacar adelante este proyecto.

A todos ellos, muchas gracias

Jesús Francisco y Abner Michel

RESUMEN

En la presente tesis se diseñó un sistema de control para el registro de préstamos de equipos a los estudiantes, para el laboratorio de Ingeniería Electrónica de la UNPRG, y dar solución a una problemática real que tiene el laboratorio de Ingeniería Electrónica con respecto al proceso de pedido y devolución de equipos, basándose en el uso de tecnología RFID para el registro de equipos y huella dactilares de los alumnos.

El uso de la plataforma Arduino que es de código abierto y que actualmente tiene gran aceptación por profesionales y estudiantes es uno de los principales elementos que contiene este sistema y que servirá para controlar la parte electrónica de este mismo.

La comunicación entre la parte electrónica y la base de datos se realiza mediante el estándar de comunicación Ethernet para garantizar una permanente transmisión de datos.

La demostración del funcionamiento de este sistema se realizó mediante un prototipo con todas las características que cuenta el sistema propuesto, y por lo tanto se demuestra que utilizando la plataforma Arduino como controlador principal se puede realizar un sistema de interconexión de adquisición de datos y las bases de datos que gestionan el proceso de pedidos y devolución de equipos.

ABSTRACT

In this thesis a control system for recording equipment loans to students Laboratory Electronic Engineering UNPRG was designed, and solve a real problem that has the laboratory of Electronic Engineering regarding the process of order and return of equipment based on the use of RFID technology for recording equipment and fingerprint imprint of students.

Using the Arduino platform is open source and now is widely accepted by professionals and students is one of the main elements contained in this system and that serve to control the electronic part of the same.

Communication between the electronic part and the database is done using standard Ethernet communication to ensure continuous data transmission.

The demonstration of this system was performed using a prototype with all the features available to the proposed, and therefore demonstrated that using the Arduino platform main controller can perform an interconnection system data acquisition and bases system data that manage the ordering process and return of equipment.

Tabla de Contenidos

INTRODUCCION	1
CAPITULO I	3
1. PLANTEAMIENTO DEL PROBLEMA.....	3
1.1. DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA	3
1.2. FORMULACIÓN DEL PROBLEMA.....	4
1.3. OBJETIVOS DE LA INVESTIGACIÓN	4
1.3.1. <i>Objetivo General</i>	4
1.3.2. <i>Objetivos Específicos</i>	5
1.4. JUSTIFICACIÓN DE LA INVESTIGACIÓN	5
1.5. LIMITACIONES DEL ESTUDIO	6
1.6. VIABILIDAD DEL ESTUDIO	6
CAPITULO II	7
2. MARCO TEORICO	7
2.1. ANTECEDENTES DE LA INVESTIGACIÓN	7
2.1.1. <i>Investigaciones hechas en la Universidad Nacional Pedro Ruiz Gallo.....</i>	7
2.1.2. <i>Investigaciones hechas en otro país.....</i>	9
2.1.3. <i>Investigaciones hechas en el Perú.....</i>	12
2.2. FUNDAMENTOS TEÓRICOS	14
2.2.1. <i>Sistemas de Identificación Automática</i>	14
2.2.1.1. Las Bandas Magnéticas.	15
2.2.1.2. El Reconocimiento Magnético de Caracteres.	15
2.2.1.3. El Infrarrojo.....	16
2.2.1.4. Los Códigos de Barras.	16

2.2.1.5. Reconocimiento Óptico de Caracteres (OCR)	19
2.2.1.6. Sistemas Biométricos	20
2.2.1.7. Identificación por Radiofrecuencia (RFID)	20
2.2.1.8. Comparación entre los Sistemas de Identificación Automática	21
2.2.2. Sistemas Arduino	24
2.2.2.1. Origen del Arduino	24
2.2.2.2. Plataforma Arduino	25
2.2.2.3. Ventajas del Arduino	26
2.2.3. Los Sistemas Biométricos	27
2.2.3.1. La Biometría	27
2.2.3.2. Rasgos Biométricos	32
2.2.3.3. La Huella Dactilar	33
2.2.4. Sistemas de Identificación por Radiofrecuencia (RFID)	37
2.2.4.1. Funcionamiento de un sistema RFID	40
2.2.4.2. Near Field communication NFC	40
2.2.5. Sistemas de Programación	42
2.2.5.1. Visual Basic	42
2.2.5.2. Base de Datos	43
2.2.5.3. Sistema de Gestión de Base de Datos	44
2.2.5.4. Microsoft SQL Server	44
2.2.5.5. Lenguaje SQL	44
CAPITULO III	45
3. DISEÑO DEL SISTEMA	45
3.1. DISEÑO DEL HARDWARE	53
3.1.1. Descripción de los elementos que intervienen en el sistema	55
3.1.1.1. Arduino Uno	55
3.1.1.2. Arduino Mega 2560	57

3.1.1.3. Arduino Ethernet Shield.....	59
3.1.1.4. Modulo Pantalla TFT LCD.....	63
3.1.1.5. CTE LCD TFT/SD shield.	65
3.1.1.6. Sensor de huellas GT-511C5.	65
3.1.1.7. Lector USB / RFID.....	67
3.1.1.8. Servidor HP Proliant.....	69
3.1.1.9. Disco duro HP.	70
3.1.1.10. Monitor HP.	70
3.1.1.11. Switch Cisco Catalyst 2960C.....	72
3.1.1.12. Etiquetas RFID.....	74
3.2. DISEÑO DEL SOFTWARE.....	74
3.2.1. Programación del Arduino	75
3.2.1.1. Entorno de desarrollo integrado (IDE) Arduino.	76
3.2.1.2. Librerías del Arduino usadas en el sistema.....	78
3.2.1.3. Diagrama de flujo para el proceso de registro de usuarios.	79
3.2.1.4. Diagrama de flujo para el Pedido de Equipos	80
3.2.1.5. Diagrama de Flujo para la Devolución de Equipos.....	82
3.2.1.6. Diagrama de flujo para la transferencia de equipos.	84
3.2.1.7. Diagrama de flujo de la programación del Arduino Uno	85
3.2.2. Programación del software de aplicación.....	87
3.2.2.1. Entorno de desarrollo integrado IDE de Microsoft Visual Studio.	87
3.2.2.2. Librerías del Visual Basic usadas en el sistema.	88
3.2.2.3. Diagrama de Flujo de la Programación del Visual Basic y el SQL Server.	89
3.2.2.4. Resumen de los comandos principales para la aplicación en Visual Basic.....	93
3.2.2.5. Comando: "CODIGO USUARIO".	94
3.2.2.6. Comando: "IDENTIFICAR SENSOR DE HUELLA".....	95
3.2.2.7. Comando: "BORRAR HUELLAS".....	96
3.2.2.8. Comando: "GRABAR HUELLA".	97
3.2.2.9. Comando: "REGISTRO".	98

3.2.2.10. Comando: "IDENTIFICAR R F I D".	99
3.2.2.11. Comando: "LEER ID TARJETA".	100
3.2.2.12. Comando: "CLAVE R F I D".	101
3.2.2.13. Comando: "LISTA MATERIALES".	102
3.2.2.14. Comando: "LISTA PENDIENTES".	103
3.2.2.15. Comando: "OBJETO RECONOCIDO".	104
3.2.2.16. Comando: "ELIMINAR OBJETO DE LA LISTA".	105
3.2.2.17. Comando: "AGREGAR OBJETO A LA LISTA".	106
3.2.2.18. Comando: "OTRO EQUIPO".	107
3.3. COSTOS DEL SISTEMA.	108
3.3.1. Costos de equipamiento.	108
3.3.2. Costos de componentes electrónicos.	109
3.3.3. Costos del diseño e implementación del sistema.	110
3.3.4. Costos del sistema general.	110
CAPITULO IV	111
4. RESULTADOS	111
4.1. DISEÑO DEL PROTOTIPO	111
4.1.1. Modulo electrónico	111
4.1.1.1. Elementos de encendido del módulo electrónico	113
4.1.1.2. Elementos de lectura de datos	115
4.1.1.3. Elementos de control y procesamiento de datos.	121
4.1.1.4. Elementos de comunicación de datos	124
4.1.1.5. Elementos Acopladores de pines.	129
4.1.1.6. Elementos de visualización de datos	130
4.1.1.7. Diagrama de bloques del hardware del prototipo.	133
4.1.1.8. Circuito de Conexión del Arduino Mega con el Sensor de Huella GT511C3.	135
4.1.1.9. Circuito de Conexión del Arduino Mega con el Shield NFC/RFID.	137

4.1.1.10. Circuito de Conexión del Arduino Mega, Arduino Uno y el Shield Ethernet.	139
4.1.1.11. Circuito de conexión para el acople de pines para los dispositivos del prototipo.	142
4.1.1. <i>PC de Control</i>	143
4.2. IMPLEMENTACIÓN DEL PROTOTIPO	144
4.2.1. <i>Verificación del estado de elementos principales</i>	145
4.2.1.1. Verificación del estado del sensor de huella GT-511C3.	145
4.2.1.2. Verificación del estado del lector NFC/RFID.	146
4.2.1.3. Calibración de la pantalla táctil.	147
4.2.2. <i>Implementación del circuito adaptador de pines</i>	150
4.2.3. <i>Compilación y Carga del código fuente del Arduino Mega y Arduino Uno.</i>	151
4.2.3.1. Compilación del código fuente del Arduino Mega.	151
4.2.3.2. Cargar el código fuente al Arduino Mega	152
4.2.4. <i>Ensamblaje del Módulo Electrónico</i>	153
4.3. FUNCIONAMIENTO DEL PROTOTIPO.....	158
4.3.1. <i>Interfaz Gráfico de la Pantalla LCD/TFT (Modulo Electrónico)</i>	158
4.3.1.1. Verificación de la conexión entre la PC de control y el Modulo Electrónico.....	158
4.3.1.2. Verificación del proceso de registro de usuarios.	159
4.3.1.3. Verificación del proceso de logueo.	163
4.3.1.4. Verificación del proceso de pedido de equipos.	164
4.3.1.5. Verificación del proceso de Devolución de equipos.	166
4.3.1.6. Verificación del proceso de Transferencia de equipos.	167
4.3.2. <i>Funcionamiento del Interfaz Gráfica de Usuario (PC de Control)</i>	168
4.3.2.1. Inicio	170
4.3.2.2. Login	171
4.3.2.3. Menú.....	172
4.4. PRUEBAS DE DEMOSTRACIÓN DEL FUNCIONAMIENTO DEL PROTOTIPO.....	192
4.5. COSTOS DEL PROTOTIPO	193
4.5.1. <i>Costos de Componentes Electrónicos</i>	193

4.5.2. Costo de Materiales para Ensamblar el prototipo.	194
4.5.3. Costos de Investigación.....	194
4.5.4. Costo de Diseño y Construcción del prototipo.....	195
4.5.5. Costos Administrativo.	195
4.5.6. Costos del prototipo general.....	196
CAPITULO V	197
5. CONCLUSIONES Y RECOMENDACIONES.....	197
5.1. CONCLUSIONES	197
5.2. RECOMENDACIONES.....	198
REFERENCIAS BIBLIOGRÁFICAS.....	199
ANEXOS.....	203

Índice de Figuras

FIGURA 2.1. ANATOMÍA DE UN CÓDIGO DE BARRAS EAN 13.	19
FIGURA 2.2. PRIMER PROTOTIPO DE ARDUINO.	25
FIGURA 2.3. ETAPAS O PROCESOS QUE COMPONEN LOS SISTEMAS BIOMÉTRICOS.	31
FIGURA 2.4. PRINCIPALES RASGOS BIOMÉTRICOS.	33
FIGURA 2.5. REPRESENTACIÓN DE MINUCIAS EN TÉRMINOS DE SUPOSICIÓN Y DIRECCIÓN.	34
FIGURA 2.6. TIPOS DE MINUCIA.	35
FIGURA 2.7. CLASIFICACIÓN DE LAS HUELLAS DACTILARES.	36
FIGURA 2.8. USOS DE LA AUTENTICACIÓN BIOMÉTRICA.	36
FIGURA 3.1. DIAGRAMA DE BLOQUES DE LA GESTIÓN DE USO DE EQUIPOS ACTUAL.	47
FIGURA 3.2. DIAGRAMA DE BLOQUES DEL CASO ESPECIAL DE TRANSFERENCIA DE EQUIPOS ACTUAL.	47
FIGURA 3.3. DIAGRAMA DE FLUJO DE LA GESTIÓN DE USO DE EQUIPOS ACTUAL.	48
FIGURA 3.4. DIAGRAMA DE BLOQUES DEL SISTEMA GENERAL PROPUESTO.	49
FIGURA 3.5. DIAGRAMA DE FLUJO DEL SISTEMA GENERAL PROPUESTO.	50
FIGURA 3.6. ESQUEMA GENERAL DEL SISTEMA PROPUESTO.	54
FIGURA 3.7. ARDUINO UNO R3.	57
FIGURA 3.8. ARDUINO MEGA.	58
FIGURA 3.9. SHIELD ARDUINO ETHERNET.	62
FIGURA 3.10. PANTALLA TFT LCD.	64
FIGURA 3.11. SHIELD LCD TFT/SD PARA LA PANTALLA TFT LCD.	65
FIGURA 3.12. SENSOR DE HUELLA GT-511C5.	67
FIGURA 3.13. LECTOR RFID.	68
FIGURA 3.14. MONITOR HP.	71
FIGURA 3.15. ETIQUETA RFID.	74
FIGURA 3.16. SKETCH DEL ARDUINO.	77

FIGURA 3.17. DIAGRAMA DE FLUJO PARA EL REGISTRO DE ALUMNOS.....	79
FIGURA 3.18. DIAGRAMA DE FLUJO PARA EL PEDIDO DE EQUIPOS.	80
FIGURA 3.19. DIAGRAMA DE FLUJO PARA DEVOLUCIÓN DE EQUIPOS.	82
FIGURA 3.20. DIAGRAMA DE FLUJO PARA TRANSFERENCIA DE EQUIPOS.....	84
FIGURA 3.21. DIAGRAMA DE FLUJO DEL ARDUINO UNO	85
FIGURA 3.22. IDE VISUAL STUDIO	87
FIGURA 3.23. DIAGRAMA DE FLUJO DE LA PROGRAMACIÓN DEL VISUAL BASIC Y SQL SERVER	91
FIGURA 3.24. COMANDO CÓDIGO DE USUARIO.	94
FIGURA 3.25. COMANDO IDENTIFICAR SENSOR DE HUELLA.	95
FIGURA 3.26. COMANDO IDENTIFICAR BORRAR HUELLA.	96
FIGURA 3.27. COMANDO GRABAR HUELLA.	97
FIGURA 3.28. COMANDO REGISTRO.	98
FIGURA 3.29. COMANDO IDENTIFICAR RFID.	99
FIGURA 3.30. COMANDO LEER ID TARGETA.....	100
FIGURA 3.31. COMANDO CLAVE RFID.....	101
FIGURA 3.32. COMANDO LISTA MATERIALES.	102
FIGURA 3.33. COMANDO LISTA PENDIENTES.	103
FIGURA 3.34. COMANDO OBJETO RECONOCIDO.	104
FIGURA 3.35. COMANDO ELIMINAR OBJETO DE LA LISTA.	105
FIGURA 3.36. COMANDO AGREGAR OBJETO A LA LISTA.....	106
FIGURA 3.37. COMANDO OTRO EQUIPO.	107
FIGURA 4.1. ESQUEMA GENERAL DEL PROTOTIPO	112
FIGURA 4.2. INTERRUPTOR CON LÁMPARA DE NEÓN.....	113
FIGURA 4.3. BOTON RESET	114
FIGURA 4.4. ADAPTADOR AC/DC	115
FIGURA 4.5. SENSOR DE HUELLA GT-511C3	116

FIGURA 4.6. SHIELD NFC/RFID 13.56 MHZ.....	118
FIGURA 4.7. ETIQUETA RFID PASIVA	121
FIGURA 4.8. ARDUINO UNO R3	122
FIGURA 4.9. ARDUINO MEGA R3	123
FIGURA 4.10. SHIELD ARDUINO ETHERNET	125
FIGURA 4.11. CONECTOR ETHERNET.....	127
FIGURA 4.12. CONEXIÓN DE LOS CONECTORES USB	127
FIGURA 4.13. CABLE UTP	128
FIGURA 4.14. CABLE USB PARA ARDUINO.....	128
FIGURA 4.15. SHIELD LCD TFT/SD PARA LA PANTALLA TFT LCD.....	129
FIGURA 4.16. CIRCUITO ADAPTADOR DE PINES.....	130
FIGURA 4.17. PANTALLA TFT LCD.	131
FIGURA 4.18. DIODOS LEDS	132
FIGURA 4.19. DIAGRAMA DE BLOQUES DEL HARDWARE DEL PROTOTIPO.	133
FIGURA 4.20. CIRCUITO DE CONEXIÓN DEL ARDUINO MEGA CON EL SENSOR GT-511C3.	136
FIGURA 4.21. CIRCUITO DE CONEXIÓN DEL ARDUINO MEGA CON EL LECTOR NFC/RFID.	137
FIGURA 4.22. INTERCAMBIO DE DATOS ENTRE EL HOST CONTROLADOR Y EL PN532.	138
FIGURA 4.23. CIRCUITO DE CONEXIÓN DEL ARDUINO MEGA, ARDUINO UNO Y SHIELD ARDUINO ETHERNET.....	141
FIGURA 4.24. CIRCUITO ESQUEMÁTICO DEL ACOPLAMIENTO DE PINES.	142
FIGURA 4.25. PC DE CONTROL	143
FIGURA 4.26. DIAGRAMA DE BLOQUES DEL PROCESO DE CONSTRUCCIÓN DEL PROTOTIPO.....	144
FIGURA 4.27. VERIFICACIÓN DEL ESTADO DEL SENSOR DE HUELLA.	145
FIGURA 4.28. LECTURA DE MEMORIA COMPLETA SOBRE UNA TARJETA MIFARE.	146
FIGURA 4.29. ESCRITURA DE DATOS EN UN BLOQUE DE MEMORIA MIFARE.	147
FIGURA 4.30. MENÚ DE INSTRUCCIONES PARA CALIBRAR LA PANTALLA TÁCTIL.	148
FIGURA 4.31. CUADROS PARA PRESIONAR EN PANTALLA TÁCTIL PARA SU CALIBRACIÓN.	148

FIGURA 4.32. CALIBRACIÓN TERMINADA Y VALORES RESULTANTES DESPUÉS DE LA CALIBRACIÓN.	149
FIGURA 4.33. VALORES DE CALIBRACIÓN COPIADOS EN EL ARCHIVO UTOUCHCD.H.	149
FIGURA 4.34. PCB DEL CIRCUITO ACOPLADOR DE PINES.	150
FIGURA 4.35. CIRCUITO DE ACOPLAMIENTO DE PINES IMPLEMENTADO.	151
FIGURA 4.36. COMPILACIÓN DEL CÓDIGO FUENTE DEL ARDUINO MEGA.	152
FIGURA 4.37. SUBIDA DEL CÓDIGO FUENTE AL MICROCONTROLADOR DEL ARDUINO MEGA.	153
FIGURA 4.38. ARDUINO MEGA Y ARDUINO UNO MONTADOS EN EL MÓDULO ELECTRÓNICO.	155
FIGURA 4.39. MÓDULO SHIELD NFC/RFID Y ADAPTADOR DE PINES MONTADO EN EL MÓDULO ELECTRÓNICO.	155
FIGURA 4.40. MÓDULO ETHERNET Y ADAPTADOR DE PINES MONTADOS EN EL MÓDULO ELECTRÓNICO.	156
FIGURA 4.41. SENSOR DE HUELLA, PANTALLA TÁCTIL Y SHIEL PARA LA PANTALLA MONTADOS EN EL MÓDULO ELECTRÓNICO.	156
FIGURA 4.42. MÓDULO ELECTRÓNICO TERMINADO.	157
FIGURA 4.43. ESPERANDO CONEXIÓN CON LA PC DE CONTROL.	158
FIGURA 4.44. MENSAJE DE BIENVENIDA.	159
FIGURA 4.45. MENÚ CON OPCIONES DE REGISTRO Y LOGUEO.	160
FIGURA 4.46. FORMULARIO PARA PONER EL CÓDIGO DE USUARIO.	160
FIGURA 4.47. FORMULARIO PARA PONER LOS NOMBRES DE USUARIO.	161
FIGURA 4.48. FORMULARIO PARA PONER LOS APELLIDOS DE USUARIO.	161
FIGURA 4.49. PETICIÓN DE LA HUELLA DACTILAR DEL USUARIO.	162
FIGURA 4.50. PETICIÓN DE LA HUELLA DACTILAR DEL USUARIO POR 3 VECES.	162
FIGURA 4.51. MENSAJE DE BIENVENIDA AL USUARIO.	163
FIGURA 4.52. MENÚ DE PEDIDO, DEVOLUCIONES Y TRANSFERENCIA DE EQUIPOS.	164
FIGURA 4.53. VENTANA DE SELECCIÓN DE EQUIPOS.	165
FIGURA 4.54. PETICIÓN DE LOS EQUIPOS PEDIDOS.	165
FIGURA 4.55. VENTANA CON LA LISTA DE EQUIPOS PARA DEVOLVER.	166
FIGURA 4.56. VENTANA DE EQUIPO DEVUELTO.	166
FIGURA 4.57. LECTOR NFC/RFID EN FUNCIONAMIENTO.	167

FIGURA 4.58. PREGUNTA SI EL USUARIO 1 DESEA TRANSFERIR LOS EQUIPOS AL USUARIO2.	168
FIGURA 4.59. VENTANA TRANSFERENCIA EXITOSA.	168
FIGURA 4.60. PANTALLA DE INICIO.	170
FIGURA 4.61. PANTALLA DE LOGIN.	171
FIGURA 4.62. PANTALLA DE MENÚ PRINCIPAL.	172
FIGURA 4.63. MENÚ USUARIOS.	173
FIGURA 4.64. VENTANA AGREGAR USUARIOS.	174
FIGURA 4.65. MENSAJE PARA GRABAR HUELLA.	174
FIGURA 4.66. VENTANA MODIFICAR USUARIO.	175
FIGURA 4.67. VENTANA ELIMINAR USUARIO.	176
FIGURA 4.68. VENTANA BUSCAR USUARIOS.	177
FIGURA 4.69. MENÚ EQUIPOS.	178
FIGURA 4.70. VENTANA AGREGAR ÁREAS.	179
FIGURA 4.71. VENTANA AGREGAR NOMBRE DE OBJETO.	180
FIGURA 4.72. VENTANA BUSCAR USUARIOS.	181
FIGURA 4.73. MENSAJE PARA PASAR EL TAG EN LA LECTORA NFC/RFID.	182
FIGURA 4.74. VENTANA MODIFICAR EQUIPOS.	183
FIGURA 4.75. VENTANA MOSTRAR LISTA DE EQUIPOS.	184
FIGURA 4.76. VENTANA MOSTRAR EQUIPOS DISPONIBLES.	185
FIGURA 4.77. MENÚ PEDIDOS.	185
FIGURA 4.78. VENTANA MOSTRAR PEDIDOS.	186
FIGURA 4.79. VENTANA TRANSFERENCIA DE EQUIPOS.	187
FIGURA 4.80. MENÚ DEVOLUCIONES.	187
FIGURA 4.81. VENTANA MOSTRAR DEVOLUCIONES.	188
FIGURA 4.82. MENÚ ADMINISTRADORES.	188
FIGURA 4.83. VENTANA REGISTRO DE ADMINISTRADORES.	189

FIGURA 4.84. MENÚ HERRAMIENTAS.	190
FIGURA 4.85. VENTANA PARA VER LA CONEXIÓN DE LOS MÓDULOS.	191
FIGURA 4.86. HABILITAR REGISTRO DE USUARIOS.	191
FIGURA 4.87. PRUEBA DE FUNCIONAMIENTO DEL PROTOTIPO.	192

Índice de Tablas

TABLA 2.1. CUADRO COMPARATIVO DE LAS TECNOLOGÍAS DE IDENTIFICACIÓN AUTOMÁTICA.....	23
TABLA 3.1. CARACTERÍSTICAS TÉCNICAS DEL ARDUINO UNO.	56
TABLA 3.2. CARACTERÍSTICAS TÉCNICAS DEL ARDUINO MEGA 2560.	59
TABLA 3.3. CARACTERÍSTICAS TÉCNICAS DEL SHIELD ARDUINO ETHERNET.....	62
TABLA 3.4. CARACTERÍSTICAS TÉCNICAS DEL MÓDULO PANTALLA TFT LCD.....	64
TABLA 3.5. CARACTERÍSTICAS TÉCNICAS DEL SENSOR GT-511C5.	66
TABLA 3.6. CARACTERÍSTICAS DEL SERVIDOR HP PROLIANT.	69
TABLA 3.7. CARACTERÍSTICAS TÉCNICAS DEL DISCO DURO HP.....	70
TABLA 3.8. CARACTERÍSTICAS TÉCNICAS DEL MONITOR HP.	71
TABLA 3.9. CARACTERÍSTICAS TÉCNICAS DEL SWITCH CISCO CATALYST 2960C.....	73
TABLA 3.10. LIBRERÍAS USADAS EN EL ARDUINO.	78
TABLA 3.11. LIBRERÍAS PARA EL VISUAL BASIC.....	88
TABLA 3.12. COMANDOS PARA EL PROGRAMA EN VISUAL BASIC.	93
TABLA 3.13. COSTOS DE EQUIPAMIENTO.	108
TABLA 3.14. COSTOS DE COMPONENTES ELECTRÓNICOS.	109
TABLA 3.15. COSTOS DEL DISEÑO E IMPLEMENTACIÓN DEL SISTEMA.....	110
TABLA 3.16. COSTOS DEL SISTEMA GENERAL.	110
TABLA 4.1. CARACTERÍSTICAS TÉCNICAS DEL SENSOR GT-511C3.	117
TABLA 4.2. CARACTERÍSTICAS TÉCNICAS DEL SHIELD NFC/RFID.	120
TABLA 4.3. CARACTERÍSTICAS TÉCNICAS DEL SHIELD ARDUINO ETHERNET.	126
TABLA 4.4. CONEXIÓN DE LOS PINES DEL SENSOR GT-511C3.	135
TABLA 4.5. CARACTERÍSTICAS TÉCNICAS DE LA PC DE CONTROL.....	143
TABLA 4.6. COSTOS DE COMPONENTES ELECTRÓNICOS.	193
TABLA 4.7. COSTOS DE MATERIALES PARA ENSAMBLAR EL PROTOTIPO.....	194

TABLA 4.8. COSTOS DE INVESTIGACIÓN.	194
TABLA 4.9. COSTOS DE DISEÑO Y CONSTRUCCIÓN DEL PROYECTO.....	195
TABLA 4.10. COSTOS ADMINISTRATIVOS.	195
TABLA 4.11. COSTOS DEL PROTOTIPO.	196

INTRODUCCION

La presente tesis describe el diseño de un sistema de control para el registros de préstamos de equipos a los estudiantes, para el laboratorio de ingeniería electrónica, con la finalidad de mejorar el orden y control de estos mismos. Estudiamos una problemática real de la Escuela Profesional de Ingeniería Electrónica seleccionando de entre dos alternativas de solución la que más adecuada esté a nuestro ámbito profesional, la primera adquirir un sistema ya fabricado de alguna empresa reconocida y la segunda diseñando un sistema de control propio teniendo en cuenta diferentes opciones actuales de Identificación Automática de Captura de Datos, la captura de información a través de características biométricas (huellas dactilares) para el registro de alumnos, y la captura de información a través de la tecnología RFID, para el control de los equipos del laboratorio de la EPIE. De las dos alternativas de solución propuestas se optó por realizar la segunda, puesto que como carrera profesional de Ingeniería Electrónica nuestro énfasis se basa en el diseño. Por lo cual se detallará todos los elementos físicos de control involucrados, así como la técnica de control empleada y el software que permitirá visualizar y registrar la información de acuerdo al registro de equipos y de alumnos.

El objetivo de este proyecto es diseñar y construir un sistema de control de registros que permita agilizar los préstamos de equipos, utilizando tecnología RFID que es una tecnología de identificación remota e inalámbrica en la cual un dispositivo lector vinculado a un equipo de cómputo, se comunica a través de una antena con un transponder o tag (etiqueta) mediante ondas de radio; además a esto que permita tener un registro de alumnos utilizando un rasgo biométrico que sea único en cada individuo (huellas dactilares).

Recogiendo la información de estas dos tecnologías de identificación automática (shield NFC/RFID y sensor de huella GT-511C3) y procesarlas con controladores (Arduino) para permitir al alumno poder manipular de manera interactiva los diferentes dispositivos que intervienen en este proyecto (pantalla táctil) y luego visualizarlas en un software de aplicación (Visual Basic) y a todo ello almacenarlos en una base de datos (SQL Server) hace que este proyecto cumpla con las expectativas propuestas para la mejora de la situación problemática que tiene el laboratorio de Ingeniería Electrónica.

Esta tesis está dividida en cinco capítulos. En el Capítulo I se hace un análisis de la situación problemática en la que se encuentra el laboratorio de la EPIE en cuanto a la dificultad del registro de préstamos de equipos, además se describen los objetivos de este proyecto y se da una justificación de la investigación realizada. En el Capítulo II se hace un recorrido de los diversos sistemas de identificación que se presentan en el mundo, y se hace un estudio de los temas empleados para este proyecto. En el Capítulo III muestra el diseño del sistema describiendo sus elementos empleados, y el funcionamiento a través de diagrama de bloques; y muestra el diseño del software que se empleará para el interfaz gráfico del usuario, junto con los diagramas de flujo para la programación de los Arduinos y la programación del software de aplicación y la base de datos del sistema. En el Capítulo IV muestra y analiza el diseño e implementación del prototipo que demostrará el funcionamiento de las características del sistema de control de este proyecto de tesis, además de los resultados obtenidos de las pruebas realizadas en la construcción del hardware y la implementación del software; se detalla también la evaluación económica del prototipo. Y por último en el capítulo V se concluye los resultados obtenidos y se hace recomendaciones para posibles mejoras de este proyecto.

CAPITULO I

1. PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción de la Realidad Problemática

En la actualidad, el laboratorio de Ingeniería Electrónica es utilizado tanto por alumnos de la misma escuela profesional como por alumnos de la escuela profesional de Ingeniería en Computación e Informática, como complemento a la educación obtenida en los salones de clases, para afianzar los conocimientos recibidos y además poner en práctica los mismos.

Por tal motivo, los elementos y equipos pertenecientes al laboratorio se encuentran en constante uso, incluso en horarios fuera de clase además que muchos de estos equipos son utilizados en otro lugar sea este como por ejemplo un aula de clase, oficinas, etc.; teniendo así muchas veces desconocimiento de la persona quien trasladó el equipo de su lugar de origen.

Los equipos que pertenecen al laboratorio de la EPIE alcanzan precios muy elevados, por lo cual se ha visto la necesidad de brindar seguridad a los mismos; a ello se le agrega la dificultad de los estudiantes para pedir los equipos y el complicado manejo del encargado para poder entregar los equipos.

En función de esto se han desarrollado diversos proyectos como control de acceso electrónico y demás, dichos proyectos han sido siempre enfocados al acceso de personas a los laboratorios más no al control de los equipos.

En otras universidades los procesos son automatizados y emplean como medida de solución, el control de acceso biométrico, el control de acceso por RFID, bandas magnéticas, código de barras, etc. o simplemente utilizan una sola tecnología para acaparar los problemas de acceso de personas o de registro de materiales, pero no dos a la vez. Es decir se centran en una sola tecnología o en un solo problema para dar solución.

1.2. Formulación del Problema

Una manera de contribuir a la automatización del laboratorio es agilizando el proceso de préstamos de equipos de manera que sea fácil e interactiva de manejar por los estudiantes de la EPIE, por ello el problema de la presente tesis es:

¿Cómo automatizar el proceso de registro y control de entrega y devolución de los equipos del laboratorio de Ingeniería Electrónica?

1.3. Objetivos de la Investigación

1.3.1. Objetivo General

Diseñar y Construir un Sistema de Control para el Registro de Préstamos de Equipos a los Estudiantes, para el Laboratorio de Ingeniería Electrónica de la UNPRG.

1.3.2. Objetivos Específicos

- Diseño del hardware para el registro dactilar de los estudiantes y el control del pedido de los equipos.
- Diseño del software de interfaz para el control de equipos que entran y salen del laboratorio.
- Implementar la base de datos con información de los alumnos y de los materiales del laboratorio.

1.4. Justificación De La Investigación

Dada la importancia de los equipos del Laboratorio y la constante renovación de los mismos, se tiene la necesidad de tener un registro claro, real y conciso sobre todos los equipos de dicho laboratorio, razón por la cual presentamos e incluimos la solución de este registro sobre ellos en un sistema electrónico fácil de usar y que este interactuará tanto con el encargado del laboratorio como con los alumnos.

En este caso, el alcance del presente proyecto se define, para los laboratorios de la EPIE, por la cantidad de equipos y su costo, de manera que se pueda tener un mejor control sobre los mismos, ya que es fundamental conocer que persona retiró cierto equipo y en qué momento, y así satisfacer lo más rápido posible una necesidad urgente de disponibilidad del mismo.

Se plantea el diseño del presente proyecto, utilizando como medio de comunicación la red de datos (LAN), en este caso en particular, la red interna disponible en los laboratorios de la EPIE, para lo cual se propone la utilización de un Shiel Arduino ETHERNET este módulo tiene

la particularidad de poseer un puerto Ethernet compatible con el estándar 802.3af y sobre todo que es compatible con la plataforma del Arduino por comunicación SPI.

1.5. Limitaciones del Estudio

En el desarrollo de este proyecto se presentaron las siguientes limitaciones:

- Demora en la entrega de los componentes electrónicos a utilizar en este proyecto, debido que la mayoría de los componentes electrónicos que se utilizaron se compraron vía online, puesto que en el mercado nacional no se encuentran disponibles y si las hay, están a un elevado costo.
- Falta de conocimientos en la programación de una base datos. Se tuvo que recurrir al aprendizaje autodidacta.

1.6. Viabilidad Del Estudio

A pesar de las limitaciones que se presentaron para el diseño y construcción de este sistema, se pudo realizar de manera satisfactoria las pruebas necesarias para el adecuado funcionamiento de este mismo.

Entonces se puede decir que este sistema de control para el registro de préstamos de equipos para el laboratorio de la EPIE es factible puesto que se tomará como prototipo para nuevas versiones en las podrían participar los mismos alumnos de la EPIE en la construcción de posibles mejoras tanto en software como en hardware de este sistema de control.

Por lo tanto se torna viable ya que los componentes necesarios para realizar este sistema de control se encuentran accesibles en el mercado interno y externo.

CAPITULO II

2. MARCO TEORICO

2.1. Antecedentes de la investigación

2.1.1. Investigaciones hechas en la Universidad Nacional Pedro Ruiz Gallo.

Título 1: Sistema de Control de Acceso con Tecnología Biométrica Informática; Caso: Control de Asistencia y Permanencia de Personal utilizando Huella Dactilar (año 2002).

Autores: Gil Seytuque, Segundo Francisco y Lujan Flores, César Augusto.

Especialidad: Ingeniería de Sistemas.

Resumen: Este proyecto constituye un acercamiento al ámbito de los sistemas biométricos. En concreto, nos centramos en el reconocimiento de huellas dactilares, desarrollando un software piloto que, sobre el soporte de una base de datos, permita una gestión de las entidades involucradas sobre una arquitectura Cliente – Servidor. El sistema hará la autenticación de los usuarios a través de un dispositivo lector de huellas dactilares. En el proceso de autenticación se utiliza el algoritmo de comprobación de plantillas basadas en la obtención de minucias. Tanto la gestión de la base de datos como las comprobaciones de usuarios registrados estarán orientadas a un sistema de control de personal y/o acceso a recintos físicos, pudiéndose realizar cualquiera de las tareas a través del software piloto de control de asistencia de personal.

El contenido de la presente investigación empieza haciendo un estudio de nuestra realidad, evaluamos las diferentes técnicas de control de personal existentes (bandas magnéticas, códigos de barras, tarjetas inteligentes) así como de las diferentes técnicas biométricas centrándonos en la tecnología de la huella dactilar (algoritmos basados en minucias y patrones), concluyendo con estudio de factibilidad de nuestro proyecto.

Título 2: Sistema de Biblioteca Electrónica para la Facultad de Ciencias Físicas y Matemáticas – UNPRG (2002).

Autores: Jorge Martín Rodríguez Castro, Percy Paredes López y Polo Arturo Mogollón Galecio.

Especialidad: Ingeniería en Computación e Informática

Introducción: En los últimos años la tecnología ha contribuido a solucionar una de las necesidades más urgentes de la sociedad, la búsqueda de información de calidad con rapidez y en formatos adecuados. Desde los sistemas locales de información al usuario, hasta los grandes buscadores de información en internet.

Es necesario rescatar la importancia de estos sistemas para garantizar al usuario un apoyo sólido al momento de tomar decisiones.

Uno de los sectores que se ha beneficiado más con los sistemas de almacenamiento y búsqueda de información son las bibliotecas, entidades que por su naturaleza requieren de sistemas eficaces que brinden a los usuarios la información que ellos requieran, haciendo uso de técnicas y herramientas modernas.

La biblioteca especializada de la Facultad de Ciencias Físicas y Matemáticas ha mostrado un crecimiento los últimos meses y los sistemas actuales de información no son eficaces, por lo que el alumno solo debe conformarse con catálogos que no brindan información completa ni actualizada.

El sistema de biblioteca Electrónico desarrollado para la facultad se presenta como una alternativa efectiva y segura para solucionar mencionados, está conformada por 5 módulos con tareas específicas, los cuales en conjunto cubren todas las necesidades de manejo de la biblioteca.

2.1.2. Investigaciones hechas en otro país

Título 1: Diseño e Implementación de un Sistema de Control e Inventario Electrónico a través de la Internet basado en la Tecnología Rfid para los laboratorios del DEEE-ESPE (Sangolquí – Ecuador, año 2011).

Autores: Daniel Alejandro Cadena Morán, Luis Guillermo Romero Sánchez.

Institución: Escuela Politécnica del Ejército

Especialidad: Ingeniería en Electrónica, Telecomunicaciones.

Resumen: El presente trabajo realiza el diseño e implementación de un prototipo de seguridad electrónica para el control de acceso de los usuarios y de los equipos de los laboratorios del DEEE basado en la tecnología RFID.

El proyecto tiene como finalidad realizar un sistema de seguridad para tener un control del acceso de usuarios a los laboratorios y también un control de los dispositivos que se encuentran

en las instalaciones del DEEE, para esto se realiza una revisión de la teoría de la tecnología RFID y microcontroladores con comunicación Ethernet que involucra en el diseño del sistema.

En el desarrollo del trabajo se realiza el diseño del hardware utilizando el microcontrolador de Microchip PIC 18F97J60 con su característica principal de disponer un módulo de comunicación Ethernet embebido, complementariamente se utiliza el lector RFID ID-20 para registrar los tags de los usuarios y se diseña un circuito RFID activo para el control de los dispositivos de los laboratorios.

Se realizó también un software en Visual Basic para almacenar los datos de los registros de los tags en una base de datos (MySQL) así como también se realizó una interfaz web para que el administrador de los laboratorios tenga acceso a la base de datos de mejor manera y pueda autorizar ingresos a los laboratorios sin tener la necesidad de realizarlo en el servidor.

Finalmente, se presenta un análisis de costos y comparación con otros sistemas similares y se realiza las recomendaciones del proyecto para un mejor desempeño.

Título 2: Implementación del Prototipo Para el Control de Acceso Docente mediante la Interoperabilidad de WSN Y RFID (Riobamba - Ecuador año 2014)

Autores: Darío Javier Llanga Herrera, David Antonio Cano Estrella.

Institución: Escuela Superior Politécnica de Chimborazo.

Especialidad: Ingeniería Electrónica en Telecomunicaciones Y Redes.

Resumen: Se realizó la implementación de un prototipo de sistema de control de acceso mediante la interoperabilidad de las tecnologías WSN y RFID para automatizar el registro de acceso docente a los laboratorios de la academia CISCO-ESPOCH. La investigación se efectuó mediante el método científico aplicativo, donde se pudo conocer las variables que intervienen en el registro como son: horarios, materias, aulas y docentes. Se comparó los protocolos de comunicación inalámbrica ZigBee y Bluetooth resultando el protocolo ZigBee que más se adaptó a nuestro sistema. Además se realizó la comparación de los sistema de control de acceso como: WSN-RFID, biométrico, dactilar y facial obteniendo el sistema propuesto un 95% de eficiencia frente a los demás analizados. Para el diseño e implementación del prototipo sistema WSN y RFID, se construyó dos nodos finales y un nodo coordinador basados en Arduino que envía la información hacia una base de datos, desarrollada mediante el software NetBeans, JAVA, PgAdmin III y PostgreSQL. Como resultado se obtuvo que el sistema propuesto brindara un control de registro docente con un 100% de efectividad, frente al 25% del método utilizado actualmente en la academia CISCO-ESPOCH Estas dos tecnologías heterogéneas, pueden ser integradas en los sistemas de control de acceso permitiendo un registro confiable y en tiempo real. Se recomienda implementar este sistema en todas las aulas de la EIE-TR de la ESPOCH debido a su bajo costo y alta efectividad.

Título 3: Automatización del proceso de control de asistencia del personal académico en tiempo real a través de reconocimiento biométrico (México, año 2010).

Autor: Alejandro Olivares Morales.

Institución: Universidad Autónoma de México.

Especialidad: Ingeniería Mecánica Eléctrica.

Alcance: El control de asistencia a través de reconocimiento biométrico supera muchas de las desventajas de identificación automática con dispositivos convencionales como son: tarjetas de registro haciendo uso de reloj de checado, números de identificación personal (NIP), tarjetas de identificación (código de barras, banda magnética, tarjetas inteligentes, RFID, etc.) ya que verifica la identidad de la persona autorizada de manera positiva y definitiva con virtualmente 0% de error.

El registro de control de asistencia se ha vuelto por sí mismo una tarea que requiere de precisión en su captura y en la emisión de reportes fidedignos que permitan automatizar el área de recursos humanos.

El presente proyecto mejorará de manera notable la rapidez con la que se realizan estas actividades, además de que la precisión de la información estará garantizada por tratarse de un sistema en la que la identificación del personal será totalmente automática.

2.1.3. Investigaciones hechas en el Perú

Título 1: Diseño de un Sistema de Control de Activos para el Almacén de Electrónica de la Pontificia Universidad Católica del Perú utilizando RFID (lima, año 2012).

Autores: Cristhian Peter Alejandro Meneses.

Institución: Pontificia Universidad Católica del Perú

Especialidad: Ingeniería Electrónica.

Resumen: La tecnología RFID tiene mucho potencial para el desarrollo de aplicaciones, pero aún no es de uso masivo, a pesar de que esta tecnología no es nueva. El RFID podría en un futuro reemplazar a los códigos de barra, actualmente esto no es posible debido al costo todavía elevados de las etiquetas. Sus principales aplicaciones son: El control de accesos, inventario y trazabilidad de objetos.

El almacén de Electrónica de la Pontificia Universidad Católica del Perú no posee procesos automatizados aplicados al control de activos, todos los procesos son hecho de forma manual, por lo cual se identificó la necesidad de automatizarlos para poder hacerlos más eficientes.

El presente trabajo de tesis tiene como objetivo diseñar un sistema de control de activos para el Almacén de Electrónica de la Pontificia Universidad Católica del

Perú utilizando la tecnología RFID, que sea capaz de gestionar los préstamos, generar reportes de estos y mantener actualizado el inventario.

El diseño propuesto resuelve la necesidad de control del inventario haciendo uso de la tecnología RFID, a través de un software que se ha desarrollado, el SADAEP, Sistema de Administración del Almacén de Electrónica PUCP, el cual permite registrar los préstamos de materiales del almacén por usuario, gestionar el inventario y generar reportes de préstamos.

2.2. Fundamentos teóricos

2.2.1. Sistemas de Identificación Automática

Los sistemas de identificación automática se emplean para el manejo de información relativa a las personas y a los objetos. Para tal efecto se utilizan formas de registro magnético, óptico, sonoro e impreso. Generalmente, estos sistemas requieren de dos componentes fundamentales: un elemento codificado que contiene la información (léase, datos procesados siguiendo alguna norma o patrón preestablecido) y un elemento con capacidad de reconocer la información. Posteriormente, el equipo lector se comunica con una computadora donde se realizan diversos procesos; en primer lugar, los datos son decodificados, esto es, se transforman en información entendible para la computadora. A continuación, la información es verificada, comparada y aceptada para luego realizar alguna decisión lógica. Gracias a que los sistemas modernos son automáticos, los procesos se agilizan, se cometen menos errores y en consecuencia se incrementa la confiabilidad y la eficiencia. Estos sistemas también son empleados para la identificación de objetos sobre todo cuando se destinan a usos comerciales. Cuanto mayor es la diversificación, esto es, cuando el número de artículos rebasa la capacidad de clasificación humana, más necesaria es la identificación exacta del producto. De tal manera que el industrial, el comerciante, distribuidor y cliente, conocidos en el argot mercadológico como los elementos integradores de los canales de distribución puedan reconocer algunas características del producto como su lugar de origen, ubicación y destino, costo y precio de venta, verificación y control, contabilidad y administración, estadísticas e inventarios [1].

A continuación se mencionan algunos de los sistemas de identificación automática, especificando sus características principales.

2.2.1.1. Las Bandas Magnéticas.

Las Bandas Magnéticas es uno de los sistemas de identificación más difundidos en la actualidad, principalmente por el uso de las tarjetas de crédito; tarjetas de identificación, tarjetas para el pago y control de servicios múltiples (autopistas, transportes, etc.). La información se graba sobre segmentos de cinta magnética similar a la empleada en la elaboración de cassettes musicales, pero es segmentada y adherida a alguna superficie que permita su manejo y lectura por los equipos destinados para este fin. Por ejemplo, en las tarjetas de crédito, el recubrimiento magnético es aplicado directamente sobre la tarjeta portadora. Cuando la cinta magnética pasa por el lector, la información es interpretada y procesada esto es lo que ocurre cuando se paga en algún establecimiento con una tarjeta de crédito y de forma inmediata, se imprime el pagaré correspondiente a la transacción comercial realizada [1].

2.2.1.2. El Reconocimiento Magnético de Caracteres.

En el Reconocimiento Magnético de Caracteres, estos equipos aprovechan las características físicas de los caracteres, ya que en su forma, estructura o relieve almacenan la información. Y esta última es leída por medios mecánicos o magnéticos, por lo general los caracteres son números, lo que permite al hombre el lograr su interpretación sin el empleo del equipo lector respectivo. Algunos ejemplos son los cheques y otros instrumentos comerciales como papel moneda o sistemas mecanizados de correspondencia. Los cheques por lo general presentan estos caracteres o números impresos en la parte inferior del mismo para su procesamiento automático mediante medios magnéticos [1].

Otrora en Europa se empleaba un código conocido por sus siglas como CMC7, donde la figura de cada número estaba formada por siete líneas verticales y siete espacios. Mediante esta forma de impresión se integraba un código binario de unos y ceros respectivamente; y según su distribución definían los números del 0 al 9, y adicionando algunos símbolos, se podía identificar magnéticamente al banco, sucursal, número de cuenta y número de cheque. Este sistema podría ser catalogado como el antecesor del código de barras. En los Estados Unidos y México se utiliza con el mismo objeto, el código E13B que recibe su nombre del proceso efectuado por el lector magnético, consistente en leer en sentido vertical líneas de trece milésimas de pulgada que condensan información similar a la citada para el caso europeo [1].

2.2.1.3. El Infrarrojo.

El Infrarrojo es un tipo de sistemas que son de uso cotidiano en casi todos los hogares, al encender el televisor con el control remoto, al abrir la puerta de la cochera, al activar la alarma del automóvil y al utilizar algunos juguetes. En el caso de las cerraduras electrónicas, la información tiene trampas y vericuetos que hacen difícil su alteración. Estos sistemas son de transmisión e identificación simultánea, en su interior la información se traduce e interpreta de diversas formas que después de ser reconocidas permiten el acceso, pasivo o activo, al banco de comandos o a la memoria del computador. La forma común más empleada en estos equipos es el sistema infrarrojo, que vino a substituir al viejo sistema del ultrasonido [1].

2.2.1.4. Los Códigos de Barras.

En 1932, un ambicioso proyecto fue realizado por un pequeño grupo de estudiantes encabezados por Wallace Flint en la Escuela de la Universidad de Harvard en Administración de

Empresas. El proyecto propone que los clientes seleccionen la mercancía deseada de un catálogo mediante la eliminación de las tarjetas perforadas correspondientes del catálogo. Estas tarjetas perforadas fueron entregadas luego a un inspector que colocó las tarjetas en un lector. El sistema luego sacó la mercancía de forma automática desde el almacén y lo entregó a la caja. Un proyecto de ley completa de los clientes ha sido producido y los registros de inventario se han actualizado. El Código de barras moderno comenzó en 1948. Bernard Silver, un estudiante graduado en el Instituto de Tecnología de Drexel en Filadelfia, escuchó al presidente de una cadena local de alimentos preguntarle a uno de los decanos para llevar a cabo la investigación para desarrollar un sistema para leer automáticamente la información del producto durante la verificación. Silver le comentó a su amigo Norman Joseph Woodland sobre la solicitud del presidente de la cadena alimentaria. Woodland era un estudiante de veintisiete años de edad y profesor de postgrado en Drexel; el problema lo fascinó y comenzó a trabajar en él [2].

El Estándar UPC (Uniform Product Code) desde 1972 es administrado en los Estados Unidos por la Uniform Code Council (UCC) y es en esta fecha cuando entra el uso del código en el comercio. En 1977, representantes de doce países europeos de la industria y el comercio decidieron formar una organización a la cual llamaron European Article Numbering Association (EAN), en la cual desarrollaron un sistema compatible con el sistema de la UCC. Al poco tiempo y tras unirse países no europeos, el nombre fue cambiado por EAN INTERNATIONAL (las siglas EAN fueron conservadas como la identificación del sistema de codificación y simbolización). El objetivo de esta organización era difundir y administrar los estándares de identificación de productos EAN y/o UPC [2].

FUNCIONAMIENTO DEL CÓDIGO DE BARRAS

El procedimiento consiste en que el símbolo de código de barras es iluminado por una fuente de luz visible o infrarrojo, las barras oscuras absorben la luz y los espacios las reflejan nuevamente hacia un escáner. El escáner transforma las fluctuaciones de luz en impulsos eléctricos, los cuales copian las barras y el modelo de espacio en el código de barras. Un decodificador usa algoritmos matemáticos para traducir los impulsos eléctricos en un código binario y transmite el mensaje decodificado a una terminal manual, PC, ó sistema centralizado de computación. El decodificador puede estar integrado al escáner o ser externo al mismo. Los escáneres usan diodos emisores de luz visible (generalmente de color rojo) o Infrarroja (LED), láser de Helio-Neón o diodos láser de estado sólido (visibles o infrarrojos) para leer el símbolo. Algunos de ellos necesitan estar en contacto con el símbolo, otros leen desde distancias de hasta varios metros, algunos son estacionarios y otros son portátiles como los escáneres manuales. La calidad de lectura de un código radica en las características del lector de código de barras (corto o largo alcance) y del tipo de papel que refleje el código [3].

Los códigos de barras se miden en proporción a la barra más delgada y en mils, o $1/1000$ de pulgada. Un código de barras de 15 mils, por ejemplo, tiene una barra delgada de $15/1000$ de pulgada de ancho. Añadido a esto, se incluyen las zonas silenciosas o espacios en blanco, a ambos lados del símbolo, para garantizar la lectura de código de barras. Un código de barras numérico puede ser de 6, 8, 12 o 13 caracteres. Los códigos de barras son asignados localmente pero son únicos a nivel mundial, se usa en más de 100 países. Para mercancías en venta. Por ejemplo tiene información sobre la del país, el fabricante, el producto y un dígito verificador [2].

Figura 2.1. Anatomía de un Código de Barras



Fuente. Propia de los Autores

En octubre de 2015 el código de barras cumplió 63 años de su aplicación en el comercio mundial. Erasmo Wong lo aplicó en nuestro país en la década de los 90 para su cadena de supermercados [4].

Según uno de los organismos internacionales que regula la adjudicación de estas "huellas dactilares" del comercio, GS1, hay más de cinco millones de códigos de barras únicos e individuales en uso en todo el mundo. Y no sólo en los pasillos de los supermercados y las tiendas de ropa; también en los hospitales, donde permiten identificar pacientes y localizar equipos de forma rápida y eficiente, o en la logística detrás de los servicios de Courier o la distribución de mercancías [5].

2.2.1.5. Reconocimiento Óptico de Caracteres (OCR).

El Reconocimiento Óptico de Caracteres es un sistema que reconoce y procesa tipos especiales de caracteres legibles para luego comparar esos patrones con aquellos que están almacenados en la memoria del computador. Mediante esta tecnología el dispositivo lector

identifica la serie de detalles de líneas, curvas y bordes que definen a cada carácter en un conjunto de caracteres definidos. Tiene presencia por lo general en el mundo de las finanzas; por ejemplo, en el procesamiento de documentos, cheques bancarios, giros, etc. Su aplicación está asociada al empleo de tinta magnética y a la tecnología conocida como MICR (por sus siglas en inglés; Magnetic Ink Character Recognition) o reconocimiento de caracteres por tinta magnética. Esta técnica mantiene la legibilidad de los caracteres aun cuando han sido cubiertos por algún sello, manchas, o similares [6].

2.2.1.6. Sistemas Biométricos.

Las tecnologías de identificación biométricas no son nuevas, basta con mencionar como ejemplo a la identificación de huellas dactilares, invento que se desarrolló y patentó en la Argentina por Juan Vucetich. Pero en cuanto a la sofisticación tecnológica estamos hablando de un campo muy amplio para explorar. Este tipo de tecnologías se utiliza generalmente para aplicaciones de control de acceso y seguridad. Un sistema biométrico utiliza información sobre alguna característica fisiológica, la cual es digitalizada y almacenada en el computador. Esta información se emplea más tarde como un medio de identificación personal. La biometría aprovecha que existen ciertas características biológicas o conductuales singulares e inalterables, por lo que pueden ser analizados y medidos para crear una huella biométrica. Estas características son difíciles de perder, transferir u olvidar y son perdurables en el tiempo [6].

2.2.1.7. Identificación por Radiofrecuencia (RFID).

RFID (Radio Frequency Identification - Identificación por radiofrecuencia) es una tecnología de identificación automática similar, en cuanto a su aplicación, a la tecnología de

código de barras. La diferencia es que RFID utiliza una señal de radiofrecuencia en lugar de una señal óptica.

Los sistemas de código de barras utilizan un lector y etiquetas impresas. En cambio RFID utiliza un lector y una tarjeta especial. En lugar de utilizar el reflejo de un rayo láser sobre la etiqueta de código de barras, RFID utiliza una señal de radiofrecuencia de baja potencia. Esta señal de radio transmitida no requiere que la tarjeta esté dentro de la línea visual del lector, ya que las señales de radio pueden propagarse fácilmente a través de materiales no metálicos. Por esto, la tarjeta de RFID (Transponder) no tiene por qué estar en contacto directo con el lector. [7].

2.2.1.8. Comparación entre los Sistemas de Identificación Automática

A continuación se muestra un cuadro comparativo entre los principales Sistemas de Identificación más usados en el mercado.

Esta comparación se hizo mediante unos parámetros de evaluación, que son las siguientes [8]:

- **Modificación de datos.** Capacidad de cambiar los datos registrados o de grabar datos nuevos, reutilización.
- **Seguridad de datos.** Capacidad de encriptar los datos que contiene la información.
- **Cantidad típica de datos (byte).** Es la cantidad de información que puede almacenar.
- **Coste, cuánto cuesta.** Además del costo individual debe considerarse la inversión en conectividad y sistemas de seguridad, costos de mantenimiento y renovación de equipos.
- **Estándares.** Escoger estándares abiertos que utilicen la mayoría de fabricantes y usuarios finales, no tecnologías propietarias de un solo fabricante.

- **Desgaste.** Cuánto tiempo permanece legible, algunas tecnologías permiten que los datos puedan leerse indefinidamente, ello implica el ciclo de vida.
- **Distancia de lectura.** La necesidad de tener visibilidad directa para leer y la distancia máxima para poder identificar al objeto (entre el lector y el receptor).
- **Interfaz.** La forma de comunicación entre el lector y el receptor.
- **Susceptible a la suciedad/líquidos.** Como afectan dichos elementos en la lectura de los datos.
- **Influencia en la dirección y posición.** Es decir la interferencia que pueda ocasionar agentes externos, pudiendo ocasionar falla al sistema.

Tabla 2.1. Cuadro comparativo de las Tecnologías de Identificación Automática.

	Código de Barras	Memorias de Contacto	Biometría	OCR	Tarjetas Magnéticas	Tarjetas Inteligentes	RFID	
							RFID Pasivo	RFID Activo
Modificación de Datos	No Modificable	Modificable	No Modificable	No Modificable	Parcialmente Modificable	Modificable	Modificable	Modificable
Seguridad de Datos	Seguridad mínima	Altamente seguro	Altamente seguro	Seguridad media	Seguridad media	Alta Seguridad	Rango de baja a alta seguridad	Alta Seguridad
Cantidad típica de datos (Byte)	1 a 100	De 8Mb en adelante	Ninguno	Ninguno	16 a 64k	1 MB	Alrededor de 64 Kbytes	Alrededor de 8 Mb
Coste	Muy Bajo	Alto (más de 1 \$ por tag)	Ninguno	Medio	Medio	Medio	Medio (unos 0.25\$ por tag)	Muy alto (más de 10\$ por tag)
Estándares	Estable e implantado	Propietario, sin estándar	Ninguno	Estable	Estable e implantado	Estable e implantado	Con estándares en fase de implantación	Propietarios y estándares abiertos
Desgaste	Limitado	Limitado	Indefinido	Limitado	Limitado	Limitado	Ninguno	Ninguno
Distancia de lectura	Pocos centímetros	Contacto necesario	Contacto directo	Contacto directo	Contacto directo	Contacto directo	Del orden de 1metro	Del orden de 100 metros
Interfaz	Lectura óptica directa	Contacto	Contacto	Contacto	Contacto	Contacto	Sin barreras aunque puede haber interferencias	Sin barreras aunque puede haber interferencias
Susceptible a la suciedad/liquidos	Alto	Alto	Ninguno	Medio	Posible	Posible	Ninguno	Ninguno
Influencia en la dirección y posición	Ligero	Ligero	Muy alto	Alto	Muy alto	Ligero	Ninguno	Ninguno

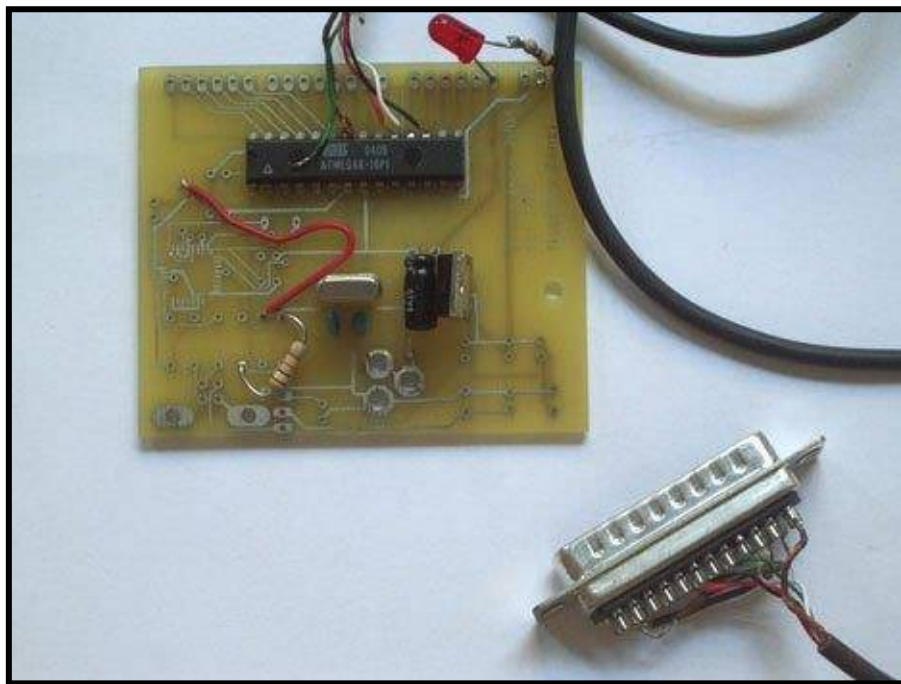
Fuente. Radiofrecuencia de Identificación (RFID), microtecnología de gran impacto [8]

2.2.2. Sistemas Arduino

2.2.2.1. Origen del Arduino

Arduino nació como un proyecto educativo allá por el año 2005. Su nombre viene del nombre del bar Bar di Re Arduino donde Massimo Banzi pasaba algunas horas, el cual a su vez viene del nombre de un antiguo rey europeo allá por el año 1002. Nació por una necesidad de subsistir ante el eminente cierre del Instituto de diseño Interactivo IVREA en Italia. Es decir, al crear unos productos open hardware (de uso público) no podría ser embargado. Es aquí donde se crea el primer prototipo de Arduino. En este proyecto Arduino participaron alumnos que desarrollaban sus tesis como Hernando Barragan (Colombia) quien desarrollo la plataforma de programación Wiring con la cual se programa el microcontrolador. La primera producción de Arduinos fue de 300 unidades y se las dieron a los alumnos del Instituto IVRAE, (las ganancias fueron de sólo 1 dólar por placa), con el fin de que las probaran y empezaran a diseñar sus primeros prototipos. Uno de los primeros proyectos fue un reloj alarma, el cual no se apagaría hasta que no te pararas de la cama. El profesor y padre de la computación física Tom Igoe, se unió al proyecto luego que se enterara del mismo a través de la web. El ofreció su apoyo para desarrollar el proyecto a grandes escalas [9].

Figura 2.2. Primer prototipo de Arduino.



Fuente. <http://spectrum.ieee.org> [9].

2.2.2.2. Plataforma Arduino

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.

Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el *Arduino Programming Language* (basado en Wiring) y el *Arduino Development Environment* (basado en Processing).

Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con *Flash*, *Processing*, *MaxMSP*, etc.) [10].

2.2.2.3. Ventajas del Arduino

- ***Las placas de Arduino son Baratas:*** Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras. Las placas de Arduino preensamblados cuestan menos de 50 soles. E incluso uno mismo podría construirla, con lo que el precio total de la placa sería menor.
- ***Multiplataforma:*** El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.
- ***Entorno de programación simple y clara:*** El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también.
- ***Arduino es libre y extensible:*** Esto quiere decir que cualquiera que desee mejorar y ampliar tanto el diseño del hardware de las placas como el entorno de desarrollo software y el propio lenguaje de programación puede realizarlo sin problemas; la cual permite que exista una amplia gama de extensiones, tanto de placas no oficiales como de librerías software de tercero, que pueden adaptarse mejor a nuestras necesidades concretas.

2.2.3. Los Sistemas Biométricos

2.2.3.1. La Biometría

El concepto biometría proviene de las palabras *bio* (vida) y *metría* (medida), por lo tanto con ello se infiere que todo equipo biométrico mide e identifica alguna característica propia de la persona. La biometría es una tecnología de seguridad basada en el reconocimiento de una característica de seguridad y en el reconocimiento de una característica física e intransferible de las personas, como por ejemplo la huella digital [11].

En [12] define que existen tres sistemas biométricos en función del contexto de la aplicación biométrica:

- **Los sistemas de verificación:** (también llamados de autenticación) autentifican la identificación de la persona mediante la comparación del rasgo biométrico acabado de capturar con el rasgo biométrico que el sistema ha capturado antes en el proceso de inscripción al sistema.

El usuario tiene que presentar su identificación mediante un carné o clave secreta. El sistema realiza una única comparación entre el rasgo biométrico que el usuario acaba de presentar con el rasgo biométrico que hay en la base de datos con la misma identificación presentada. La salida de un sistema de verificación es normalmente binaria: es la misma persona si los rasgos biométricos coinciden (son muy similares) o son dos personas diferentes en otro caso. En algunos casos, los rasgos biométricos y la identificación de la base de datos están en la propia tarjeta del usuario en forma encriptada. En este caso, se

dice que disponemos de una base de datos distribuida entre todas las tarjetas de los usuarios.

- **Los sistemas de identificación:** reconocen a la persona a través de la busca del rasgo biométrico que más se asemeja al usado para identificarlo en toda una base de datos. El usuario no aporta ninguna información de su identificación, como era el caso del sistema de verificación. El sistema lleva a cabo una comparación uno a muchos. Esto quiere decir que el rasgo biométrico del usuario desconocido se compara con muchos rasgos biométricos de una base de datos. Hay varias salidas a este sistema. La más sencilla es devolver el nombre de la persona (identificador) cuyo rasgo biométrico se asemeja al introducido. Otra posibilidad es deducir que este rasgo biométrico no pertenece a ninguna persona de la base de datos (este caso se da cuando las distancias del rasgo biométrico con todos los rasgos biométricos de la base de datos es superior a un umbral). Finalmente, y es el caso más usual en las aplicaciones forenses, el sistema no devuelve una sola persona sino una lista de candidatos. Es decir, devuelve las personas cuya distancia de sus rasgos biométricos con el rasgo biométrico introducido es inferior a un umbral.
- **Sistema de matriculación:** Tanto los sistemas de verificación como de identificación necesitan un proceso previo. Este proceso se encarga de recoger el rasgo biométrico (o los rasgos biométricos) junto con la identificación de la persona. Este proceso es muy importante puesto que se encarga de relacionar la identificación de la persona con el rasgo biométrico. Normalmente, este proceso tiene lugar ante una persona autorizada que vela por la veracidad de los datos que aporta el usuario (carné de identidad, pasaporte) y controla que realmente sea este usuario el que presenta el rasgo

biométrico al sistema. Además, durante el proceso de matriculado, esta persona verifica la calidad de los datos biométricos obtenidos. Si considera que los datos no tienen suficiente calidad, pide al usuario que vuelva a presentar el rasgo biométrico (huella dactilar, cara, iris) al sistema.

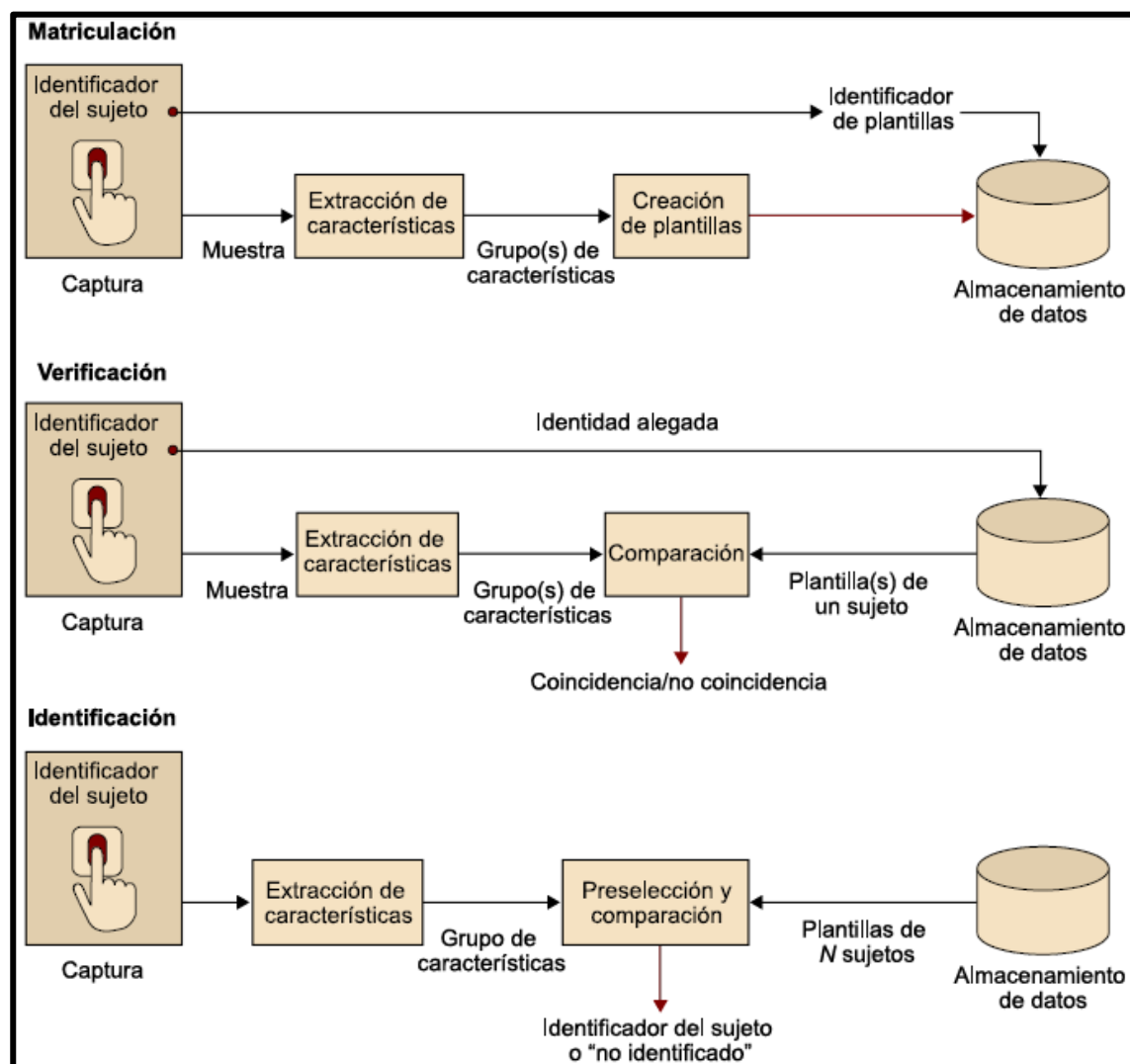
Los tres sistemas antes mencionados usan los procesos siguientes [12]:

- **Captura:** La representación digital del rasgo biométrico tiene que ser capturada. El sensor biométrico es usualmente un sistema para capturar una imagen (excepto la identificación del hablador, que es una máquina de grabar voz). Normalmente, la información capturada se denomina muestra. A veces, el sistema de captura también incorpora otros periféricos para introducir información no biométrica o mostrar información.
- **Extracción de las características:** Con el objetivo de facilitar la comparación, aumentar la información y reducir el ruido, la representación original digital (imagen digital) se procesa normalmente con un extractor de características para generar una representación compacta y más identificadora llamada registro de identificación o conjunto de características.
- **Creación de la plantilla:** La plantilla es una forma compacta de representar un conjunto de muestras de una sola característica biométrica (por ejemplo, se puede crear una plantilla de dieciséis muestras diferentes de la imagen de la cara de una misma persona). El proceso de creación de la plantilla recibe como entrada los registros de identificación y crea una información más compacta donde se intenta extraer la información que persiste en todas las muestras puesto que se consideran los rasgos característicos. En algunos

casos, esta plantilla está formada por una única muestra y por lo tanto se puede representar como un registro de identificación.

- **Comparación:** El proceso de comparación recibe como entrada un registro de identificación y una plantilla y calcula una distancia entre los dos. A veces, en lugar de una distancia obtiene una probabilidad de que representen un mismo individuo. En el proceso de verificación, hay un umbral interno al sistema que solo lo puede modificar el administrador del sistema. Si la distancia es inferior al umbral (o la probabilidad es superior al umbral), el sistema considera que los dos datos provienen de la misma persona, de lo contrario, se considera que provienen de dos personas diferentes.
- **Selección o filtrado:** En los sistemas de identificación con muchos datos (podemos hablar de 50 millones de huellas dactilares), el filtrado es un método para aumentar el tiempo de respuesta del sistema. Con técnicas típicas de las bases de datos, logran no tener que explorar la base de datos entera y así ganar tiempo.
- **Almacenamiento de los datos:** Es el proceso para almacenar la información del usuario. Esta información está compuesta por un identificador único (por ejemplo, el número del DNI o del pasaporte), la plantilla biométrica y otros datos (por ejemplo, la dirección o la profesión). En función de la aplicación, los datos se almacenan en sistemas de almacenamiento centralizados (para poder llevar a cabo la identificación) o en tarjetas inteligentes (para poder llevar a cabo la verificación). Además, se aplican técnicas de encriptación con todos los datos para que así el registro formado por el número del DNI más los rasgos biométricos sea indivisible.

Figura 2.3. Etapas o procesos que componen los Sistemas Biométricos.



Fuente. La Biometría para la Identificación de las Personas [12]

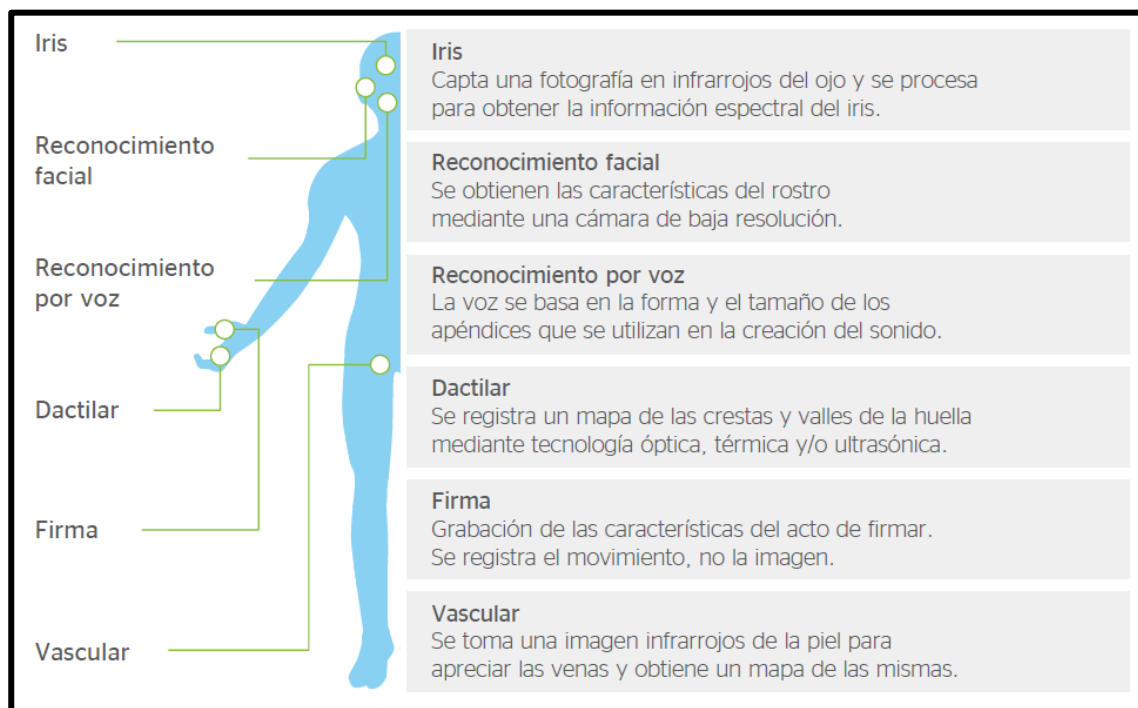
2.2.3.2. Rasgos Biométricos

Se puede usar cualquier rasgo anatómico o de comportamiento humano como identificador biométrico para identificar o verificar a las personas si satisface los requerimientos siguientes [12]:

- **Universalidad:** Cada persona debe poseer ese rasgo biométrico.
- **Particularidad:** Todas las personas tienen que ser suficientemente diferentes en términos del rasgo biométrico.
- **Permanencia:** El rasgo biométrico debe ser invariante en el tiempo y a cualquier otro factor desde el punto de vista de la comparación entre rasgos biométricos.
- **Medible:** El rasgo biométrico se tiene que poder medir cuantitativamente.
- **Rendimiento:** El rasgo biométrico debe garantizar precisión y robustez en diferentes factores ambientales.
- **Aceptabilidad:** Los usuarios del sistema deben aceptar el uso de ese rasgo biométrico para su identificación.
- **No falsificable:** El rasgo biométrico tiene que garantizar que su falsificación sea dificultosa.

En la figura 2.16 muestra algunos rasgos biométricos utilizados para diferentes aplicaciones biométricas.

Figura 2.4. Principales Rasgos Biométricos.



Fuente. Tecnología Biométrica BBVA [13]

2.2.3.3. La Huella Dactilar

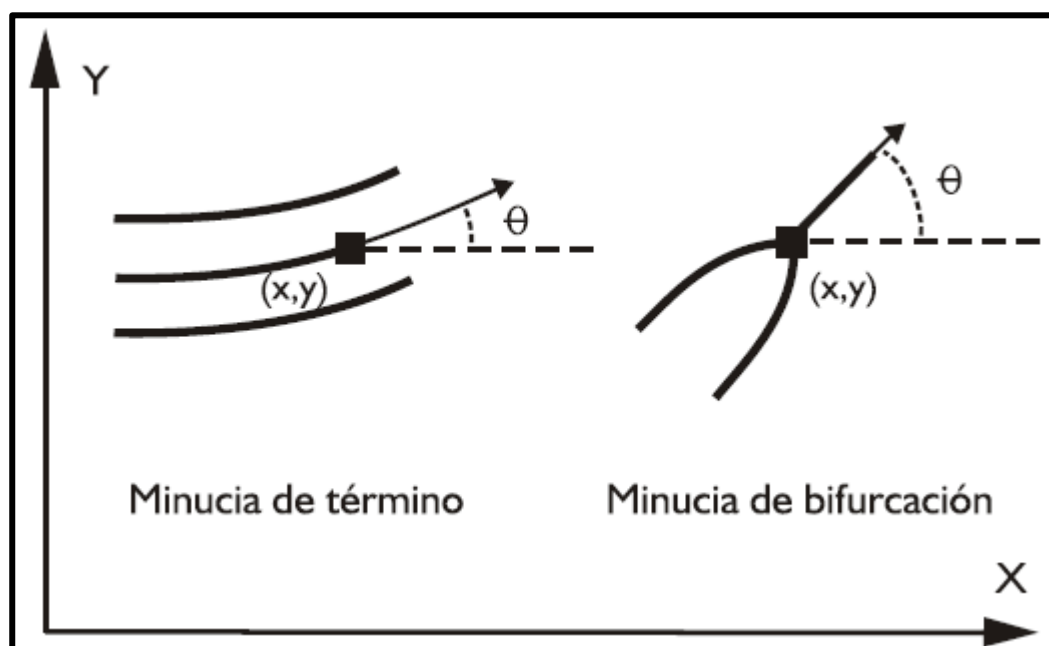
La huella dactilar es un rasgo biométrico de cada persona y la ciencia encargada de su estudio es la dactiloscopia (de las voces griegas “daktilos” que significa dedos y “skopein” que significa examen o estudio). Una huella dactilar es la representación de la morfología superficial de la epidermis de un dedo.

Características De Las Huellas Digitales

Las huellas digitales tiene características visibles que son [14]: las crestas, los surcos, las minucias, el núcleo y el delta.

- **Surcos:** Son la áreas de la huella digitales que se encuentran a cierto nivel de profundidad.
- **Crestas:** Son la áreas de la huella digitales que se encuentran al nivel de la misma.
- **Minucias:** Es el punto en el que termina una cresta o se bifurca en dos o más crestas. Se define en términos de sus coordenadas (x, y) y el ángulo de orientación de la cresta.

Figura 2.5. Representación de minucias en términos de suposición y dirección.



Fuente. Reconocimiento y validación de huellas dactilares utilizando una Red Neuronal [14].

La característica más interesante que presentan tanto las minucias como los puntos singulares núcleo y delta es que son únicos para cada individuo y permanecen inalterados a través de su vida. A pesar de la variedad de minucias, las más importantes son las terminaciones y bifurcaciones de crestas. Esto último se debe a que las terminaciones de crestas representan aproximadamente el 60.6% de todas las minucias en una huella y las bifurcaciones el 17.9%. Además varias de las minucias menos típicas pueden expresarse en función de las dos señaladas.

Tipo de minucias

- ✓ Una isla se presenta cuando se unen las dos ramas de dos bifurcaciones diferentes, ambas bifurcaciones deben estar frente a frente.
- ✓ En una bifurcación la línea de la cresta se divide en varios segmentos de línea.
- ✓ La terminación es donde la línea de la cresta inicia o acaba.
- ✓ El punto es una línea de la cresta bastante corta que no está conectada a ninguna otra línea de cresta.
- ✓ Lago es una cresta que se divide en dos ramas y se unifica otra vez luego de recorrer una distancia corta creando un área cerrada.

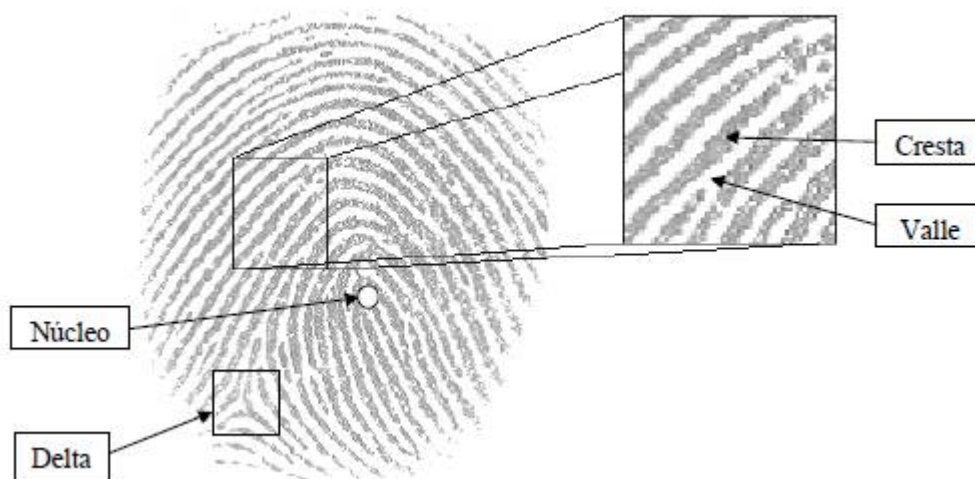
Figura 2.6. Tipos de Minucia.



Fuente. Reconocimiento y validación de huellas dactilares utilizando una Red Neuronal [15].

- **Núcleo:** Es el punto más alto de la cresta curvada más interior.
- **Delta:** Es la zona donde se unen tres flujos de crestas.

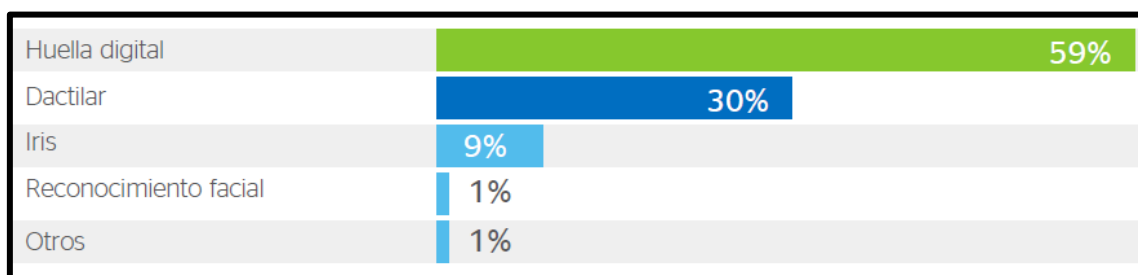
Figura 2.7. Clasificación de las Huellas Dactilares.



Fuente. Vulnerabilidades en Sistemas de Reconocimiento basados en huella Dactilar [15].

El sistema de identificación biométrica con huellas digitales es uno de los sistemas actualmente más rápidos y su seguridad depende del grado de aceptación que se le dé al mismo. Así mismo podemos decir que en la actualidad el reconocimiento automático de huellas dactilares se ha consolidado como el rasgo biométrico más utilizado en el mercado.

Figura 2.8. Usos de la Autenticación Biométrica.



Fuente. Tecnología Biométrica BBVA [13]

2.2.4. Sistemas de Identificación por Radiofrecuencia (RFID)

RFID (Identificación por Radiofrecuencia) es un método de almacenamiento y recuperación remota de datos, basado en el empleo de etiquetas o “tags” en las que reside la información. RFID se basa en un concepto similar al del sistema de código de barras; la principal diferencia entre ambos reside en que el segundo utiliza señales ópticas para transmitir los datos entre la etiqueta y el lector, y RFID, en cambio, emplea señales de radiofrecuencia (en diferentes bandas dependiendo del tipo de sistema, típicamente 125 KHz, 13,56 MHz, 433-860-960 MHz y 2,45 GHz). Todo sistema RFID se compone principalmente de cuatro elementos [16]:

- **Una etiqueta RFID**, también llamada tag o transpondedor (transmisor y receptor). La etiqueta se inserta o adhiere en un objeto, animal o persona, portando información sobre el mismo. En este contexto, la palabra “objeto” se utiliza en su más amplio sentido: puede ser un vehículo, una tarjeta, una llave, un paquete, un producto, una planta, etc. Consta de un microchip que almacena los datos y una pequeña antena que habilita la comunicación por radiofrecuencia con el lector.
- **Un lector o interrogador**, encargado de transmitir la energía suficiente a la etiqueta y de leer los datos que ésta le envíe. Consta de un módulo de radiofrecuencia (transmisor y receptor), una unidad de control y una antena para interrogar los tags vía radiofrecuencia. Los lectores están equipados con interfaces estándar de comunicación que permiten enviar los datos recibidos de la etiqueta a un subsistema de procesamiento de datos, como puede ser un ordenador personal o una base de datos. Algunos lectores llevan integrado un programador que añade a su capacidad de lectura, la habilidad para escribir información en las etiquetas.

- **Un ordenador, host o controlador**, que desarrolla la aplicación RFID. Recibe la información de uno o varios lectores y se la comunica al sistema de información. También es capaz de transmitir órdenes al lector.
- **Un middleware** y en backend un sistema ERP de gestión de sistemas IT son necesarios para recoger, filtrar y manejar los datos.

Clasificación de los distintos sistemas RFID existentes [16]:

- ***Según su capacidad de programación:***
 - ✓ ***De sólo lectura:*** las etiquetas se programan durante su fabricación y no pueden ser reprogramadas.
 - ✓ ***De una escritura y múltiples lecturas:*** las etiquetas permiten una única reprogramación.
 - ✓ ***De lectura/escritura:*** las etiquetas permiten múltiples reprogramaciones.
- ***Según el modo de alimentación:***
 - ✓ ***Activos:*** si las etiquetas requieren de una batería para transmitir la información.
 - ✓ ***Pasivos:*** si las etiquetas no necesitan batería.
- ***Según el rango de frecuencia de trabajo:***
 - ✓ ***Baja Frecuencia (BF):*** se refiere a rangos de frecuencia inferiores a 135 KHz.
 - ✓ ***Alta Frecuencia (AF):*** cuando la frecuencia de funcionamiento es de 13,56 MHz.
 - ✓ ***Ultra Alta Frecuencia (UHF):*** comprende las frecuencias de funcionamiento en las bandas de 433 MHz, 860 MHz, 928 MHz.

- ✓ **Frecuencia de Microondas:** comprende las frecuencias de funcionamiento en las bandas de 2,45 GHz y 5,8 GHz.
- **Según el protocolo de comunicación:**
 - ✓ **Dúplex:** el transpondedor transmite su información en cuanto recibe la señal del lector y mientras dura ésta. A su vez pueden ser:
 - **Half dúplex,** cuando transpondedor y lector transmiten en turnos alternativos.
 - **Full dúplex,** cuando la comunicación es simultánea. En estos casos la transmisión del transpondedor se realiza a una frecuencia distinta que la del lector.
 - ✓ **Secuencial:** el campo del lector se apaga a intervalos regulares, momento que aprovecha el transpondedor para enviar su información. Se utiliza con etiquetas activas, ya que el tag no puede aprovechar toda la potencia que le envía el lector y requiere una batería adicional para transmitir, lo cual incrementaría el coste.
- **Según el principio de propagación:**
 - ✓ **Inductivos:** utilizan el campo magnético creado por la antena del lector para alimentar el tag. Opera en el campo cercano y a frecuencias bajas (BF y AF).
 - ✓ **Propagación de ondas electromagnéticas:** utilizan la propagación de la onda electromagnética para alimentar la etiqueta. Opera en el campo lejano y a muy altas frecuencias (UHF y microondas).

2.2.4.1. Funcionamiento de un sistema RFID.

Los sistemas RFID constan de etiquetas o tags, lectores y software para procesar los datos. El lector envía una señal de radio que es recibida por todos los tags presentes en el campo de radiofrecuencia sintonizado con dicha frecuencia. Los tags reciben la señal a través de sus antenas y responden transmitiendo los datos que almacenan de los sensores. El dispositivo de lectura/escritura recibe la señal del tag a través de su antena, la descodifica y transfiere los datos al sistema informático a través de una conexión de cable o inalámbrica [17].

2.2.4.2. Near Field communication NFC

La tecnología NFC (Comunicaciones en Campo Cercano) [16] ofrece nuevas funcionalidades a la tecnología RFID propiamente dicha, gracias a la combinación de una etiqueta y un lector RFID en un mismo dispositivo. Este hecho facilita la comunicación bidireccional entre dos dispositivos, pudiendo actuar ambos como emisor y como receptor. La tecnología NFC rompe por tanto con la separación funcional descrita en apartados anteriores, entre el lector y la etiqueta RFID. La tecnología NFC resulta especialmente útil aplicada a los dispositivos móviles (teléfonos, PDAs), de modo que el usuario lleva en su terminal móvil además de una etiqueta RFID con sus datos (o la información necesaria para cada aplicación), un lector para poder leer información de otras etiquetas. De este modo se complementa la comunicación a corta, media y larga distancia provista por los dispositivos móviles (Bluetooth, WiFi, GPRS, UMTS) con la comunicación a muy corto alcance (centímetros) provista por NFC.

NFC surgió en el año 2002 como resultado de la cooperación entre Philips, Sony y posteriormente Nokia. Se trata de un estándar ISO, ECMA y ETSI que trabaja en la banda de

frecuencia AF (13,56 MHz) y por tanto con un rango de cobertura pequeño (<10 cm).

Actualmente ofrece velocidades de transmisión de datos de 106 kbps, 212 kbps y 424 kbps no está pensado para transmitir grandes volúmenes de datos, sino más bien para intercambiar información de forma rápida, eficiente y segura. Al igual que el resto de tecnología RFID, el protocolo NFC cubre los modos de operación activo y pasivo.

El NFC Forum ha desarrollado cuatro tipos diferentes de etiquetas que todo dispositivo NFC debe soportar:

- **Tipo 1:** basado en ISO 14443 A. Proporcionado por Innovision Research & Technology (Topaz™). Posee una capacidad de hasta 1 Kb y velocidades de transmisión de 106 Kbps. Son etiquetas de bajo coste.
- **Tipo 2:** basado en ISO 14443 A. Proporcionado por NXP Semiconductors (MIFARE Ultralight). Posee una capacidad de 0,5 Kb y velocidad similares a las tipo 1. También son de bajo coste.
- **Tipo 3:** basado en FeliCa (que deriva de ISO 18092). Proporcionado por Sony, con capacidades de hasta 2 Kb y velocidades de 212 Kbps. El coste es mayor aunque útil para aplicaciones más complejas.
- **Tipo 4:** Basado en ISO 14443 A/B. En este caso son varios fabricantes los que proporcionan este tipo de etiquetas. Posee capacidad de hasta 64 Kb y velocidades comprendidas entre 106 Kbps y 424 Kbps.

2.2.5. Sistemas de Programación

2.2.5.1. *Visual Basic*

Visual Basic es un lenguaje desarrollado por Microsoft a principios de la década de los noventa y que se ha seguido usando, en distintas versiones, hasta hoy día. Tras su lanzamiento adquirió gran popularidad porque Microsoft facilitó un entorno de programación amigable, que era de fácil uso para los programadores y permitía crear aplicaciones para Windows con aspecto profesional y con rapidez. A lo largo de la historia ha tenido distintas versiones y son [18]:

- **Visual Basic 1.0, 2.0 y 3.0:** podemos denominar a estas versiones como antecedentes del Visual Basic que conocemos hoy en día.
- **Visual Basic 4.0 y 5.0:** fueron versiones que incrementaron la popularidad de Visual Basic y se integraron en los sistemas operativos de última generación de Microsoft.
- **Visual Basic 6.0:** fue la consagración de Visual Basic y supuso su expansión a gran escala. Visual Basic 6 se usó para el desarrollo de numerosas aplicaciones profesionales y empresariales por todo el mundo y tuvo un gran éxito. Tanto, que aunque ya no cuenta con soporte por parte de Microsoft, es un lenguaje que sigue siendo usado debido a que numerosos programadores estaban muy especializados en él y a que muchas empresas tenían programas funcionando con este lenguaje y no han querido eliminar estos programas.

A partir de 2002 Visual Basic se integró dentro de una plataforma de programación más amplia que admitía la programación en distintos lenguajes denominada Visual Studio.NET y empezó a usarse el término Visual Basic.NET para hacer referencia a estas nuevas versiones. No

obstante, Visual Basic ha seguido siendo un lenguaje propio que ha seguido usándose. Ya dentro de Visual Studio pueden destacarse estas versiones [18]:

- **Visual Basic 2005:** introdujo cambios importantes en relación a la filosofía de programación en que se basaba Visual Basic (aunque el lenguaje continúa siendo esencialmente el mismo)
- **Visual Basic 2008, 2010, 2012, 2020...** : Visual Basic es un lenguaje vivo que se mantiene integrado dentro de la plataforma Visual Studio y del que continúan apareciendo versiones que van introduciendo progresivas mejoras.

El término “Visual” hace referencia a que nos movemos en un entorno gráfico e intuitivo. “Basic” alude al conocido lenguaje BASIC (Beginners All – Purpose Symbolic Instruction Code) del que se conservan diversas palabras claves e instrucciones. Visual Basic es un lenguaje que se apoya en objetos. En Teoría se podría distinguir entre programación orientada a objetos, programación basada en objetos o programación que utiliza objetos.

2.2.5.2. Base de Datos

Una base de datos (cuya abreviatura es BD) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y diferentes usuarios deben poder utilizar estos datos. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red ya que se debe poder compartir esta información. De allí el término *base*. "Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado [19].

2.2.5.3. Sistema de Gestión de Base de Datos

SGBD (Sistema de Gestión de Base de Datos) o en ingles DBMS (Database management system) es un conjunto de servicios (aplicaciones de software) para administrar bases de datos, que permite [19]:

- Un fácil acceso a los datos.
- El acceso a la información por parte de múltiples usuarios.
- La manipulación de los datos encontrados en la base de datos (insertar, eliminar, editar).

Algunos de los productos comerciales más difundidos son: Borland Paradox, Filemaker, IBM DB2, Ingres, Interbase, Microsoft SQL server, Microsoft Access, Microsoft FoxPro, Oracle, Sybase, MySQL, PostgreSQL, mSQL, SQL Server 11.

2.2.5.4. Microsoft SQL Server

Microsoft® SQL Server™ es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos [20].

2.2.5.5. Lenguaje SQL

S.Q.L. significa lenguaje estructurado de consulta (*Structured Query Language*). En [21] el SQL es un lenguaje estándar de cuarta generación que se utiliza para definir, gestionar y manipular la información contenida en una Base de Datos Relacional.

CAPITULO III

3. DISEÑO DEL SISTEMA

Este sistema de control es diseñado, para dar solución a la problemática en el que hoy vive la escuela profesional de Ingeniería Electrónica, cuyas características son:

- El desorden al momento en que el usuario gestiona el pedido, devolución o la transferencia de los equipos solicitados;
- La demanda de tiempo que conlleva a realizar estos procesos.
- Gestionar el pedido de un equipo sin saber su disponibilidad.

En la actualidad el alumno (usuario) para realizar la gestión de pedidos, devoluciones o transferencia de equipos, al encargado o responsable del Laboratorio de Ingeniería Electrónica, debe cumplir con los siguientes procesos.

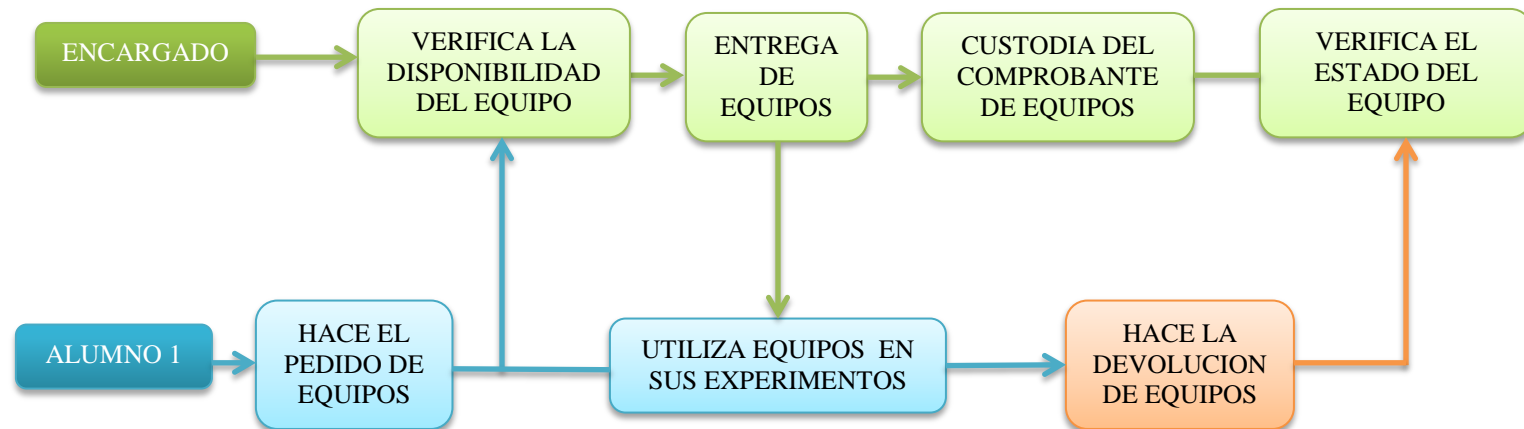
- **PEDIDOS.** El Alumno al momento de solicitar el pedido de los equipos que necesita para realizar sus prácticas de laboratorio, se acerca al encargado y le solicita una ficha de papel en la que debe de escribir sus nombres y apellidos , código universitario, N° DNI, Esc. Profesional, y marcar los equipos a solicitar, que se encuentran impresos en dicha ficha sin saber su disponibilidad; para luego ser entregado junto con su DNI al encargado del Laboratorio y este realizar la verificación de los datos registrados en dicha ficha; el encargado se cerciora si los datos registrados por el alumno son correctos, comparándolo con su DNI o Carnet Universitario y ve la disponibilidad del equipo solicitado, en el caso

de no ser encontrado en el laboratorio, dicha petición del equipo será anulada siendo tachado y se procede con la petición de los equipos disponibles solicitados. Al momento en que el encargado le hace entrega de los equipos, el alumno debe firmar la ficha y dejar su DNI hasta la devolución del equipo o realizar la transferencia con otro alumno.

- **DEVOLUCION.** El alumno se acerca al encargado del laboratorio con los equipos a devolver, y le hace entrega de los mismos; el encargado revisa la conformidad de los equipos, devuelve el DNI y desecha la ficha. En el caso de no existir conformidad el encargado establecerá responsabilidad.
- **TRANSFERENCIA.** El alumno hace entrega de los equipos a su compañero que está solicitándolo. El encargado actualiza la nueva gestión, devolviendo el DNI al usuario inicial (alumno) y pidiendo el DNI al usuario final (compañero).

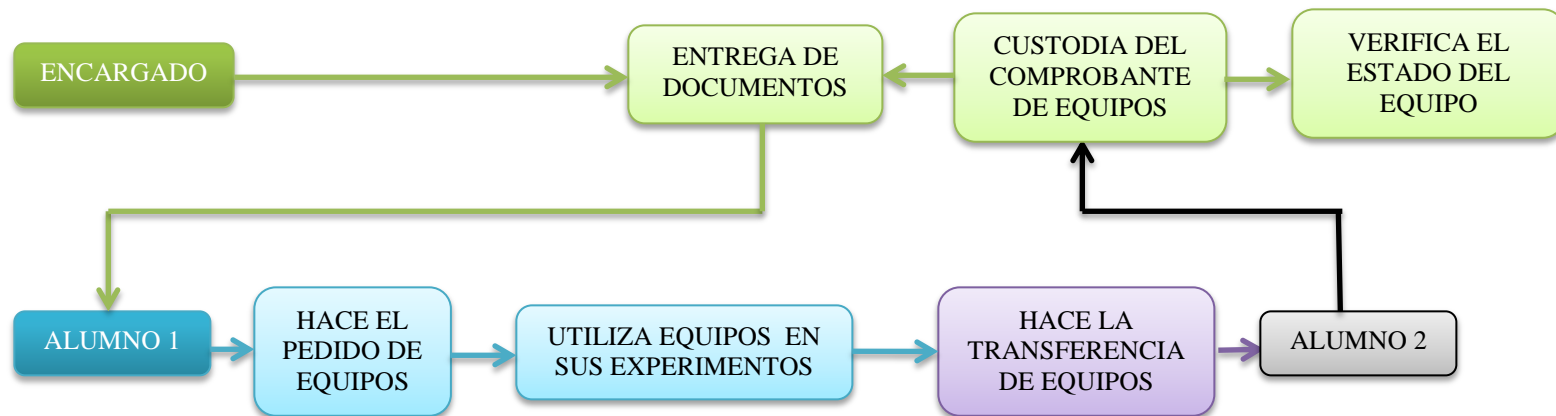
A continuación se presente un diagrama de bloques que describe el proceso actual de pedidos de equipos en el laboratorio de Ingeniería Electrónica.

Figura 3.1. Diagrama de bloques de la gestión de uso de equipos actual.



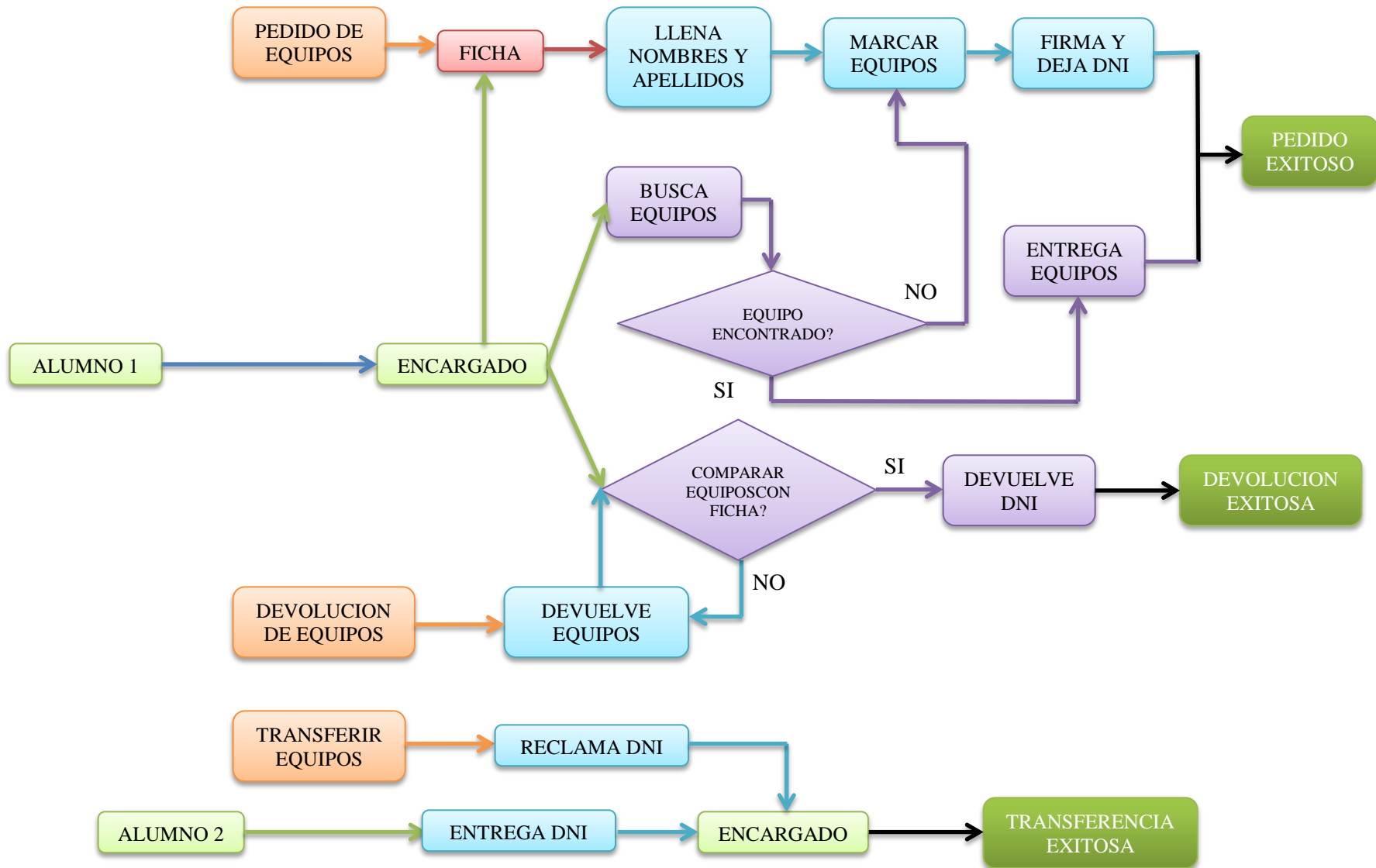
Fuente. Propia de los autores

Figura 3.2. Diagrama de bloques del caso especial de transferencia de equipos actual.



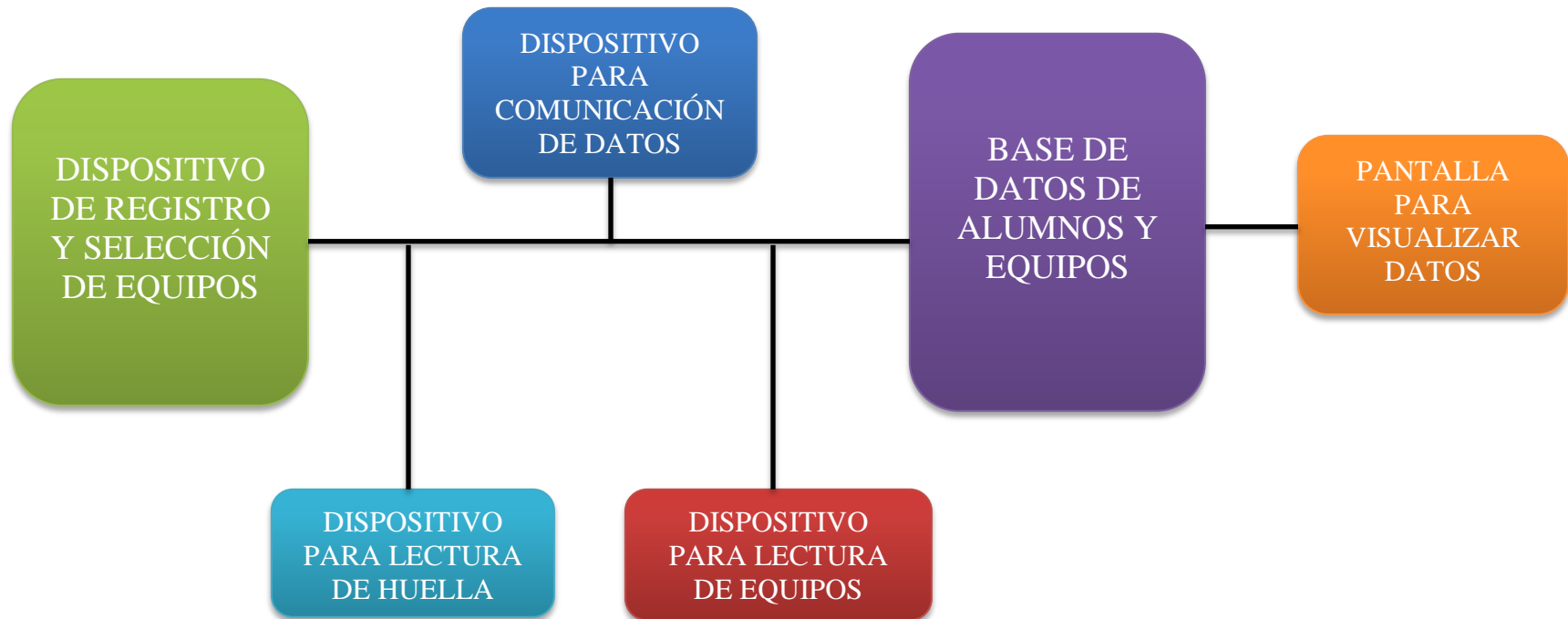
Fuente. Propia de los autores

Figura 3.3. Diagrama de flujo de la gestión de uso de equipos actual.



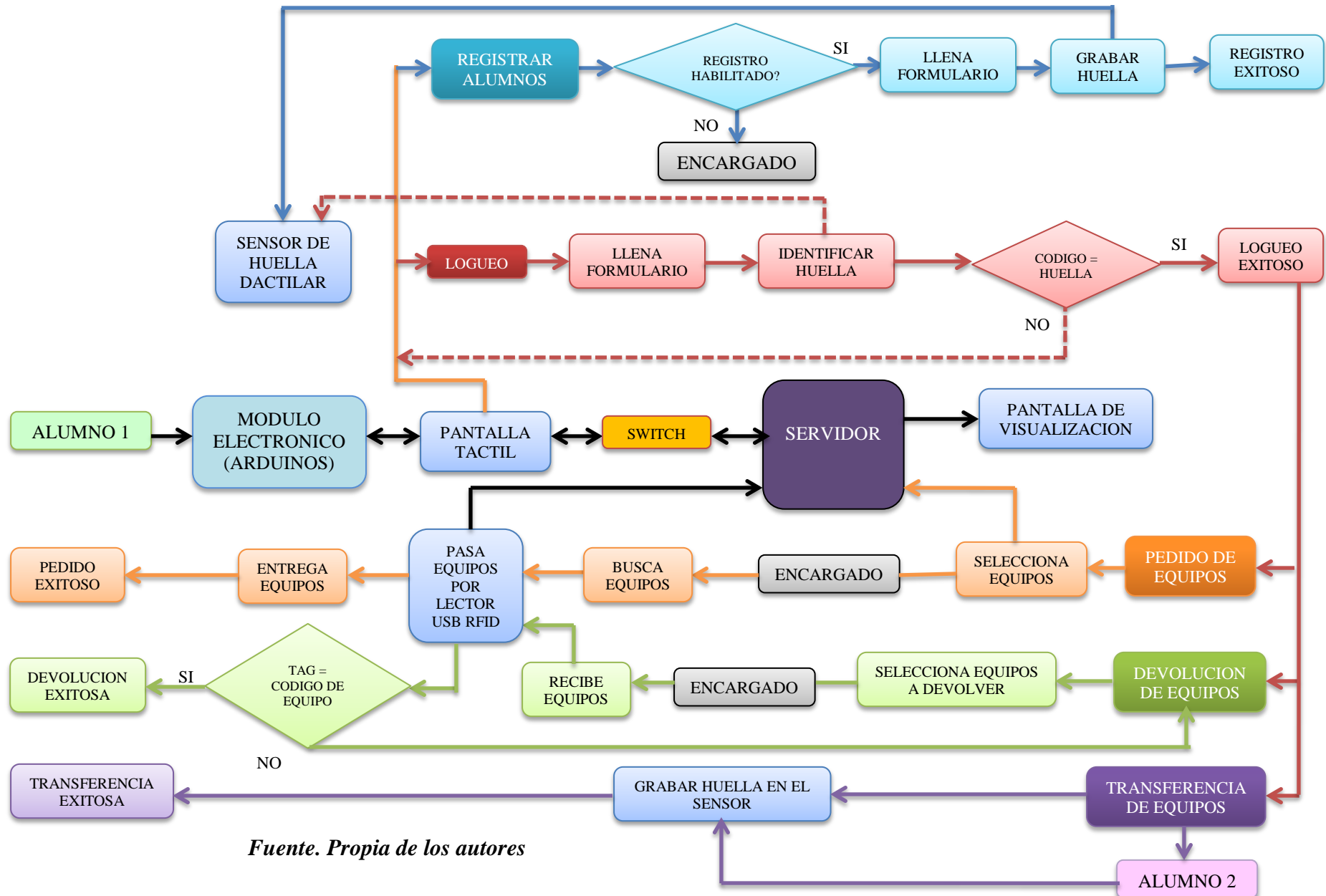
Fuente. Propia de los autores

Figura 3.4. Diagrama de bloques del sistema general propuesto.



Fuente. Propia de los autores

Figura 3.5. Diagrama de flujo del sistema general propuesto.



Fuente. Propia de los autores

Este sistema de control permite el registro en su base de datos, tanto de los alumnos como de los equipos, en la opción de registro de alumnos se puede realizar de dos formas:

- **Registro de alumnos.**

- ✓ **Registro por el encargado.** Se realiza el registro acercándose directamente al encargado y por medio de este registrarse,

- ✓ **Registro por el usuario.** El alumno se acerca directamente al módulo electrónico y registrarse de allí, con la debida autorización del encargado del laboratorio.

- **Registro de equipos.**

El registro de equipos en la base de datos del sistema de control, será realizado por el encargado del laboratorio

A continuación se describe el diagrama de bloques funcionamiento del nuevo sistema que se plantea ejecutar, para ello se ha dividido en 3 procesos:

- **Proceso de registro de alumnos:**

Lo primero que hará cada usuario para registrarse es acercarse al encargado y pedir la habilitación de registro, el alumno se dirigirá a la pantalla táctil y deberá llenar un formulario de registro (nombres, apellidos, código, etc.) luego se le pedirá que se acerque al sensor de huella para colocar cualquiera de sus dedos en el sensor y así pueda tener un registro completo, posteriormente dicha información se enviará a la PC de Control donde se almacenaran los datos completos en una base de datos.

- **Proceso de pedido de equipos:**

Para este proceso el sistema cuenta con dos lectores RFID, uno directamente en el interior del almacén para la lectura de equipos que se entregan a través del encargado y el segundo en la parte exterior del almacén para la lectura de equipos que se devuelven.

Para este proceso el usuario debe estar registrado en la base de datos del sistema; el usuario deberá acercarse a la pantalla táctil y debe loguearse con su código y su huella dactilar, luego aparecerá una opción de pedidos y un menú con los equipos disponibles en el laboratorio, se seleccionará los equipos que desee y se le enviara a la pc de control para que el encargado lo busque y pase cada equipo por la lectora USB-RFID que tiene el encargado y este pueda ser entregado al usuario solicitante y la cual registrará su pedido en la base de datos del sistema . Si se diera el caso que no se mostrara el equipo que el usuario desee, este puede digitar en la pantalla táctil el nombre del material que requiere y enviarlo a la PC de Control para que el encargado del laboratorio lo busque en el almacén del laboratorio.

- **Proceso de devolución de equipos:**

Para este proceso el usuario debe estar registrado en la base de datos del sistema; el alumno deberá acercarse a la pantalla táctil y debe loguearse con su código y su huella dactilar, luego aparecerá una opción de devoluciones y una lista de materiales que el alumno ha retirado del almacén, luego seleccionará los materiales que se va a devolver, cada material debe pasar por el lector NFC/RFID que se encuentra en el módulo electrónico y luego se enviará a la PC de Control para que registre su devolución en la base de datos del sistema.

Este sistema cuenta con dos opciones, la primera devolver los equipos de manera personal de tal forma que la devolución es hecha por el mismo alumno que retiro los equipos, y la otra

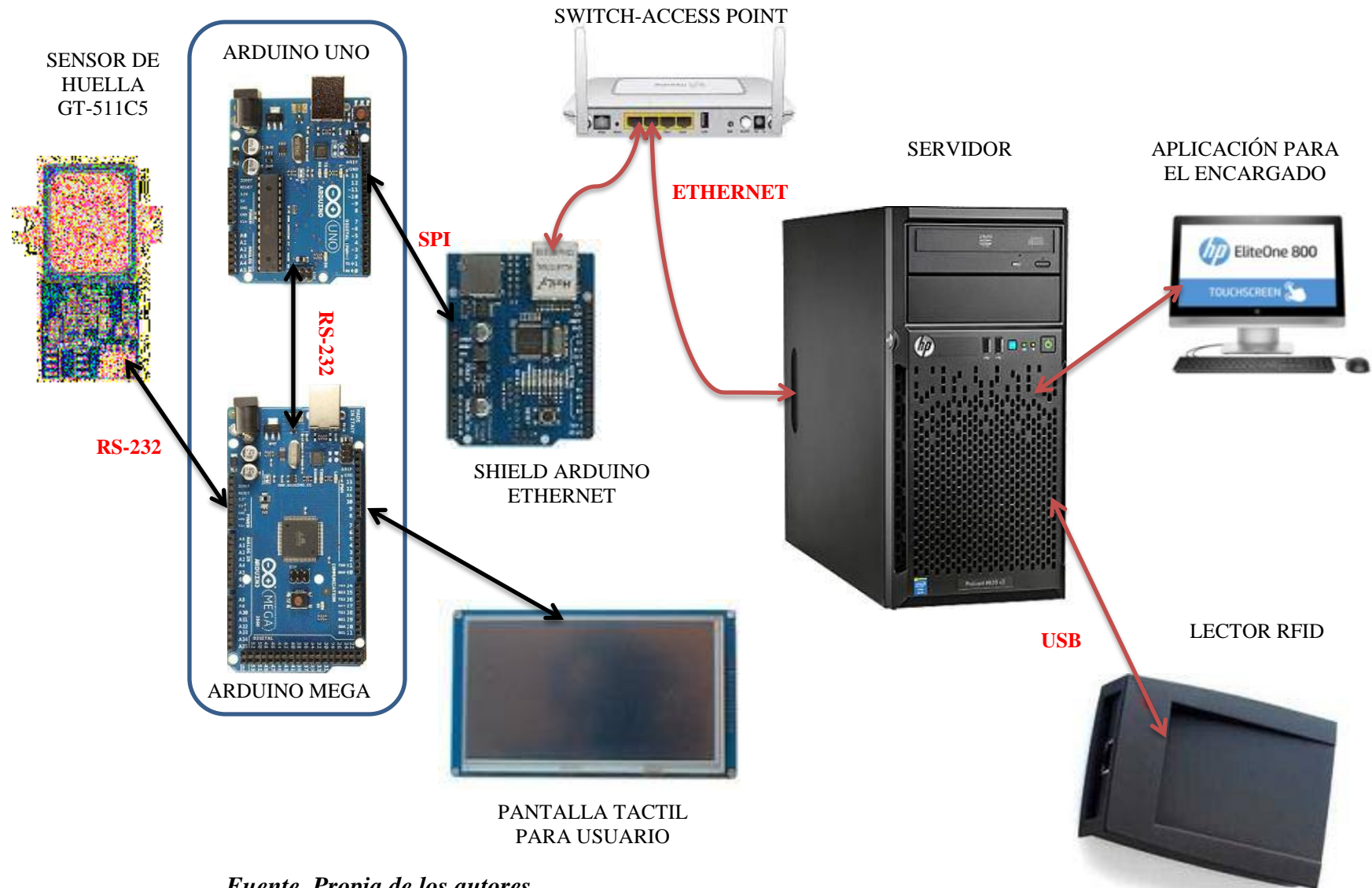
forma es de transferencia, que consiste en transferir el pedido de equipos a otro alumno para que el los haga uso, de tal manera que si hubiera un alumno que desea devolver equipos y otro alumno que desee pedirlos el primero deberá transferirlo por medio de un formulario en la pantalla táctil al otro alumno para que quede libre de toda deuda.

3.1. Diseño del Hardware

Para el diseño del hardware del sistema de control del presente proyecto, se optó por utilizar la plataforma Arduino, puesto que presenta ciertas ventajas tanto en hardware como en software, y además que Arduino es de bajo costo y muy flexible en comparación con otras tecnologías. En esta presente tesis se hace uso de un sensor de huella dactilar que hace el registro de alumnos, y un lector RFID para registrar los equipos del laboratorio de Ingeniería Electrónica.

A continuación se presenta un esquema general del nuevo sistema de control de registro de préstamos de equipos para el laboratorio de Ingeniería Electrónica.

Figura 3.6. Esquema general del sistema propuesto.



Fuente. Propia de los autores

3.1.1. Descripción de los elementos que intervienen en el sistema

3.1.1.1. Arduino Uno.

El Arduino Uno es una placa microcontroladora basada en el ATmega328. Tiene 14 pines de entrada/salida (E/S) digital (de los cuales 6 pueden ser usados como salidas PWM), 6 entradas analógicas, un oscilador de cuarzo a 16MHz, una conexión USB, un conector para alimentación, una cabecera ICSP, y un botón de reset.

El Arduino Uno puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa (no USB) por ejemplo con un adaptador AC/DC o una batería. El adaptador se conecta en la entrada de alimentación externa de la placa. Si es una batería se pueden conectar en los pines GND y VIN del conector de alimentación, La tarjeta puede funcionar con un suministro externo de 6 a 20 voltios. Si se proporcionan menos de 7V, la placa puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

El Arduino Uno incorpora un microcontrolador ATmega328 que tiene una capacidad de memoria 32KB (de los cuales 0,5KB son usados por el bootloader). Además tiene 2KB de SRAM y 1KB EEPROM (que puede ser leída y escrita con la librería EEPROM).

Al Arduino Uno solo tiene un puerto serial 0 RX y 1 TX, también tiene dos interruptores externos el Pin 2 y el Pin 3, y para la comunicación SPI utiliza los pines 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Esto se detallará más adelante.

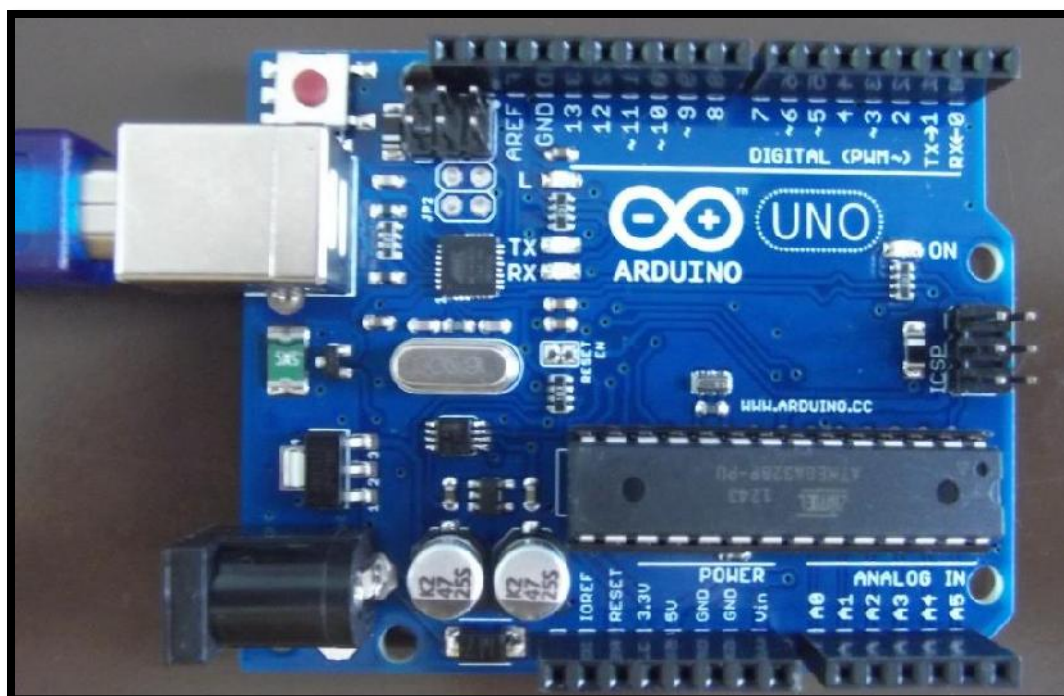
➤ *Resumen de características técnicas del Arduino Uno.*

Tabla 3.1. Características técnicas del Arduino Uno.

CARACTERÍSTICAS	DESCRIPCIÓN
Microcontrolador	Atmega328
Voltaje de Operación	5V
Tensión de Entrada (recomendada)	7-12V
Tensión de Salida (limite)	6-20V
Pines digitales de E/S	14 (de las cuales 6 proveen salida PWM)
Pines de Entradas Analógicos	6
Corriente máxima DC por Pin E/S	40Ma
Corriente máxima DC por Pin 3.3V	50Ma
Memoria Flash	32KB (de los cuales 0.5 es usado por el bootloader)
SRAM	2KB
EEPROM	1KB
Frecuencia de Reloj	16MHz

Fuente. Página web del Arduino [22].

Figura 3.7. Arduino Uno R3



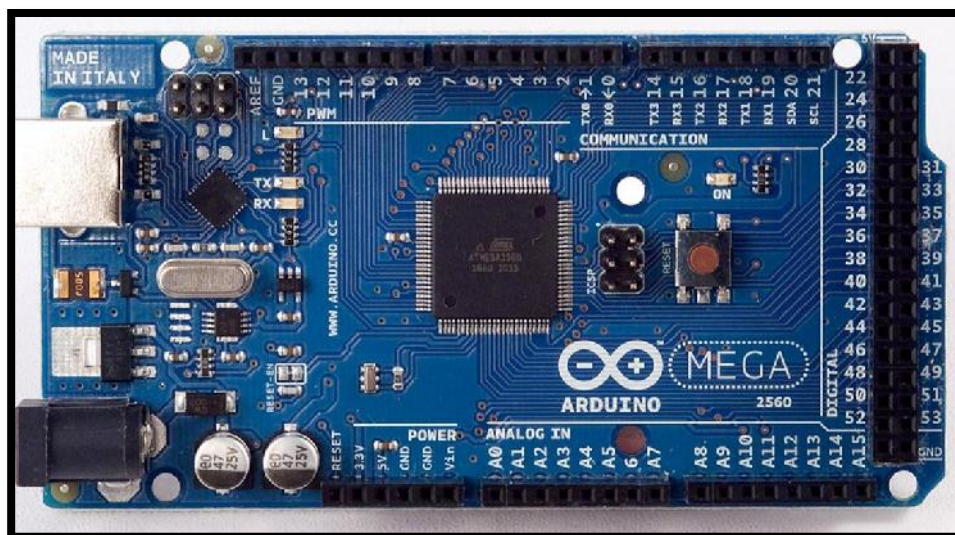
Fuente. Propia de los autores

3.1.1.2. Arduino Mega 2560.

El Arduino Mega 2560 está hecho para proyectos más grandes. Ofrece una mayor cantidad de entradas y salidas, además de otras mejoras. Está basado en el microcontrolador ATmega2560. Tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serial por hardware), un oscilador de cuarzo 16 MHz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Incorpora todo lo necesario para que el microcontrolador trabaje. Esta placa al disponer una gran cantidad de pines de entradas y salidas, permitirá conectar en nuestro sistema de control un importante número de dispositivos actuadores (pantalla táctil, sensor de huella y lector NFC/RFID) que se encargaran de hacer el registro de usuarios y equipos.

Es importante notar que funcionalmente todas las placas de Arduino son todas idénticas, la única diferencia es el tamaño de sus capacidades de memoria, y el número de entradas y salidas. Todas las placas se basan en microcontroladores Atmel con un CPU AVR. Por lo tanto, el modelado será válido para todas las placas de Arduino, independientemente del microcontrolador que usen, siendo la única restricción el número de entradas y salidas y sobre todo la función del proyecto que se desee realizar. En nuestro proyecto resalta la utilidad del Arduino Mega 2560 por su cantidad de pines y por su capacidad memoria al momento de procesar la información.

Figura 3.8. Arduino Mega.



Fuente. Propia de los autores

➤ **Resumen de características técnicas del Arduino Mega 2560.**

Tabla 3.2. Características técnicas del Arduino Mega 2560.

CARACTERÍSTICAS	DESCRIPCIÓN
Microcontrolador	Atmega2560
Voltaje de Operación	5V
Tensión de Entrada (recomendada)	7-12V
Tensión de Salida (limite)	6-20V
Pines digitales de E/S	54 (de las cuales 15 proveen salida PWM)
Pines de Entradas Analógicos	16
Corriente máxima DC por Pin E/S	40Ma
Corriente máxima DC por Pin 3.3V	50Ma
Memoria Flash	256KB (de los cuales 8 es usado por el bootloader)
SRAM	8KB
EEPROM	4KB
Frecuencia de Reloj	16MHz

Fuente. Página web del Arduino [23]

3.1.1.3. Arduino Ethernet Shield.

El Shield Arduino Ethernet [24] tiene la capacidad de conectarse a una red cableada TCP/IP. Aporta la misma funcionalidad que la placa Arduino Ethernet pero en forma de shield complementario acoplado a la placa Arduino Uno. De hecho, este shield tiene el mismo chip

controlador W5100 que la placa Arduino Ethernet, y se configura con la misma librería de programación: la librería “Ethernet”, la cual ya viene por defecto en el lenguaje Arduino.

Las entradas y salidas tienen exactamente la misma disposición y funcionalidad que las del Arduino Uno. Incluso si fuera necesario, se puede conectar sin problemas un segundo shield en la parte superior del shield Ethernet para seguir sumando funcionalidad.

El procedimiento de cargar programas en el microcontrolador del Arduino Uno acoplada al shield Ethernet no varía respecto al realizado normalmente con un Arduino Uno independiente: primero debemos conectar el Arduino Uno a nuestro computador mediante el cable USB, y una vez cargado el programa, como siempre, podremos seguir alimentando la placa vía USB o bien desconectarla del computador y enchufarla a una fuente de alimentación externa. A partir de entonces, al conector RJ-45 del shield le podremos conectar un cable de red (técnicamente, un cable de par trenzado de categoría 5 o 6) de tipo “estándar” si deseamos comunicar el shield a un switch o un router o de tipo “cruzado” si deseamos comunicar el shield directamente a un computador.

El shield requiere 5 V para funcionar. Este voltaje lo aporta la placa Uno mediante el encaje del pin de alimentación correspondiente entre placa y shield. La comunicación entre el chip W5100 y el Arduino Uno se establece mediante los pines 10, 11, 12 y 13 (vía SPI) por lo que estos pines no se pueden utilizar para otro propósito. Esto implica que realmente al usar este shield tenemos 4 entradas/salidas digitales menos.

Este shield también incorpora (tal como lo hace la placa Arduino Ethernet) un zócalo para colocar una tarjeta microSD, la cual se podrá utilizar mediante la librería de programación “SD”,

que viene por defecto en el lenguaje Arduino. Al igual que ocurre con el chip W5100, para poder comunicarse con la tarjeta microSD la placa Arduino Uno utiliza el protocolo SPI, pero para ello esta vez emplea como pin SS el número 4 en vez del 10 (reservado al W5100). Es decir, emplea los pines 4, 11, 12 y 13.

Al igual que la placa Arduino Ethernet, este shield también tiene la posibilidad de acoplársele un módulo PoE. Finalmente, indicar que este shield también dispone de su propio botón de “reset”, el cual reinicia tanto el chip W5100 como la propia placa Arduino, y de una serie de LEDs informativos interesantes de conocer: “PWR” (indica que la placa y el shield reciben alimentación eléctrica), “LINK” (indica la presencia de una conexión de red, y parpadea cuando el shield recibe o transmite datos), “FULLD” (indica que la conexión de red es “full duplex”), “100M” (indica la presencia de una conexión de red de 100 megabits/s en vez de una de 10 megabits/s), “RX” (parpadea cuando el shield recibe datos), “TX” (parpadea cuando el shield envía datos) y “COLL” (parpadea cuando se detectan colisiones de paquetes en la red).

➤ ***Principales características.***

- El conector RJ45 es compatible con PoE y tiene de reserva algunos pines para el montaje del módulo PoE opcional.
- Es compatible con el Arduino Mega, a través de la comunicación SPI (cabecera ICSP).
- Reajuste de alimentación en el IC para eliminar problema de reinicio.
- Inversor Schmitt – trigger en el IC para anular la selección del W5100.
- Todos los indicadores LEDs de estado de W5100 están montados en la placa.
- El Arduino no se sobrecalentará cuando se conecte una alimentación externa de 12 V.

Figura 3.9. Shield Arduino Ethernet.



Fuente. Propia de los autores

➤ *Resumen de características técnicas del Shield Arduino Ethernet.*

Tabla 3.3. Características técnicas del Shield Arduino Ethernet.

ÍTEM	DESCRIPCIÓN
Voltaje de Operación	DC 4 ~ 12V
Estándar de Comunicación	IEEE802.3af
Interfaces de Comunicación	UART, I2C, SPI (ISCP Arduino Mega)

Fuente. Propia de los autores

3.1.1.4. Modulo Pantalla TFT LCD.

En este sistema se optó por emplear una pantalla de cristal líquido de transistores de película fina (Thin Film Transistor-Liquid Crystal Display-“TFT LCD”) para una fácil manipulación y control de este.

Esta pantalla TFT LCD tiene una resolución de 800 por 480 y su tamaño es de 7 pulgadas, su funcionamiento está basado en el controlador SSD1289 para gestionar la pantalla TFT LCD, al ser TFT estos transistores permiten mejorar la calidad de imagen en comparación a las pantallas tradicionales LCD, también consumen menos energía, lo cual la hace ideal para el sistema implementado, gracias a su panel táctil el usuario podrá manipular los menús desarrollados en la interfaz gráfica del sistema.

Lo interesante de esta pantalla TFT LCD es que se cuenta con una librería de apoyo realizado por Henning Karlsen (desarrollador de librerías de software libre para pantalla TFT LCD de cualquier tamaño para Arduino), para la administración de su hardware a través de Arduino, tanto para la parte táctil y para el LCD de la pantalla, aunque se tuvo que modificar esta librería para que funcione con el modelo de controlador de la pantalla empleada y para que funcionase con la resolución de la pantalla de 7 pulgadas .

La pantalla TFT LCD permite leer imágenes y presentarlas como interfaz, para lo cual requiere que dichas imágenes estén en el formato RGB565 o RAW565 (.raw), este formato hace referencia al dato de los pixeles en bruto de un archivo de mapa de bits. En la programación de la placa Arduino se debe especificar las dimensiones de las imágenes a emplear para que se desplieguen adecuadamente en la pantalla TFT LCD. Para insertar imágenes a la pantalla TFT

LCD se deben guardar las imágenes en formato .raw en una memoria micro SD y ponerla en la parte posterior de la pantalla, esta memoria también sirve para otros tipos de archivo.

Figura 3.10. Pantalla TFT LCD.



Fuente. Propia de los autores.

➤ **Resumen de características técnicas del Módulo Pantalla TFT LCD**

Tabla 3.4. Características técnicas del Módulo Pantalla TFT LCD.

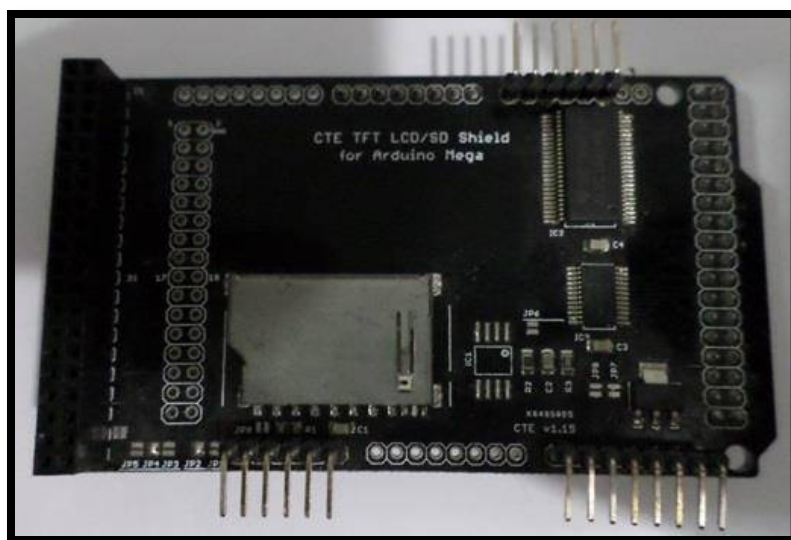
ÍTEM	DESCRIPCIÓN
Tamaño de la Pantalla	7 pulgadas
Resolución de la Pantalla	800 x 480
Controlador	SSD1963
Tipo de LCD	TFT transmisión normal Blanca
Corriente de Consumo	400 mA
Voltaje de Operación	3.3 ~ 5 V
Interfaz de Comunicación	Interfaz de bus paralelo de 8/16 bits

Fuente. Propia de los autores.

3.1.1.5. CTE LCD TFT/SD shield.

Este Shield CTE LCD TFT/SD es un shield adaptador de pines que sirve como acople para la pantalla TFT LCD y este se pueda conectar directamente sin cables con el Arduino Mega. Tiene una ranura SD que sirve para poder guardar imágenes que se mostraran en la pantalla, o simplemente otro archivo que requiera la aplicación. Esta versión de shield LCD tiene 40 pines que son compatibles con el Arduino Mega.

Figura 3.11. Shield LCD TFT/SD para la Pantalla TFT LCD.



Fuente. Propia de los autores.

3.1.1.6. Sensor de huellas GT-511C5.

El GT-511C5 es una versión actualizada del GT-511C3 que tiene una mayor capacidad de memoria. El modulo para este prototipo puede almacenar hasta 2000 huellas dactilares diferentes (que es 10 veces más que la versión antigua) y es capaz de reconocer la huella a 360°, Es pequeño y fácil de montar. La interfaz de conexión con el conector JST-SH tiene cuatro señales: VCC, GND, Tx y Rx.

➤ *Características técnicas del Sensor GT-511C5.*

Tabla 3.5. Características técnicas del Sensor GT-511C5.

ÍTEM		DESCRIPCIÓN
CPU		ARM Cortex M3 Core (holtek HT32F2755)
Sensor		Sensor óptico
Área efectiva del sensor		14 x 12.5(mm)
Tamaño de imagen		202 x 258 pixeles
Resolución		450 dpi
El máximo número de huellas		2000 huellas dactilares
Modalidad de coincidencia		1:1, 1:N
Tamaño de la plantilla		496 Bytes (plantilla) + 2bytes (suma de comprobación)
Interface de comunicación		UART, velocidad en baudios predeterminada = 9600bps después del encendido, USB ver1.1, toda velocidad
Tasa de falsa aceptación		<0.001%
Tasa de falso rechazo		<0.1%
Tiempo de enrolamiento		<3seg (3 huellas dactilares)
Tiempo de identificación		1.5seg (2000 huellas dactilares)
Voltaje de operación		DC 3.3 ~ 6V
Corriente de operación		<130mA
Entorno operativo	Temperatura	-20°C ~ +60°C
	Humedad	20% ~ 80%
Entorno de almacenamiento	Temperatura	-20°C ~ +60°C
	Humedad	10% ~ 80%

Fuente. Datasheet del Sensor de Huellas Dactilares GT-511C5 [25].

Figura 3.12. Sensor de Huella GT-511C5.



Fuente. Página web de makerlab electronics [26].

3.1.1.7. Lector USB / RFID.

Para la lectura de etiquetas RFID que contendrá cada equipo se usará un lector RFID con conectividad USB e irá directamente conectado a la PC de control (Servidor) para el registro de equipos y almacenar la información a la base de datos.

El lector envía periódicamente señales para ver si hay alguna etiqueta en sus inmediaciones. Cuando capta una señal de una etiqueta (la cual contiene la información de identificación de esta), extrae la información y se la pasa al subsistema de procesamiento de datos.

➤ ***Principales características.***

- Hecho de ABS+PC con textura.
- Sus dimensiones son 116mm x 75mm x 17.3mm y un peso de 120gr.
- Trabaja con protección IP = IP65.
- Interfaz usado es USB.
- Trabaja a una distancia de 2 a 5cm, y con una lectura de menos de 300ms.
- Utiliza una alimentación de 6-14VDC, 70mA.
- Protocolo de 26 bit Wiegand.
- LED indicador de dos colores.
- Trabaja a una temperatura de -10° a 70°C.
- Humedad de Trabajo es de 10 - 90 % sin condensación.

Figura 3.13. Lector RFID.



Fuente. Página prosperu.com [27]

3.1.1.8. Servidor HP Proliant.

Un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

El sistema utilizara un servidor para el almacenamiento de información de los equipos y de los alumnos, para luego ser entregado a manera de petición a los clientes, es decir al módulo electrónico (Arduino) y al encargado por ello se utilizará un servidor de la marca HP, cuyas características principales son:

➤ *Características técnicas del Servidor HP Proliant.*

Tabla 3.6. Características del Servidor HP PROLIANT.

MODELO	HP PROLIANT DL180 GEN9	
Procesador (GHZ)	INTEL	XEON E5-2609V3 (1.90GHZ, 15 MB CACHÉ, 6 NUCLEOS, 85W)
	Soporta solo un procesador	
Memoria	8 GB DDR4	
	Número de ranuras:	16
Unidades de almacenamiento	Numero de discos soportados	8 LFF
Incorpora	Red	Adaptador Ethernet 36li de 1GB y 2 puertos por controlador
	Controlador	Almacenamiento: Adaptador Host H240
Sistema operativo	No Incluye	
Factor de forma	2U	
Voltaje de alimentación	Potencia de la fuente (W)	550W

Fuente. Página web tienda de computo Perú [28].

3.1.1.9. Disco duro HP.

El servidor HP Proliant DL180 GEN9 necesita en su capacidad de almacenamiento de información un disco duro de la misma marca para almacenar la información de los registros de alumnos y de equipos necesarios para el funcionamiento de este sistema de control. Las características principales del disco duro HP son las siguientes:

➤ **Características técnicas del Disco duro HP.**

Tabla 3.7. Características técnicas del disco duro HP.

Ítems	Descripción
Capacidad de almacenamiento	1 TB
Interface de conexión	SATA
Velocidad de Rotación	7200 RPM
Velocidad de transferencia física	6.0 GB/S
Formato	3.5 PULG
Dimensiones	1.69 x 10.56 x 1191 cm

Fuente. Página web tienda de computo Perú [28].

3.1.1.10. Monitor HP.

Para visualizar los datos del registro de alumnos y de equipos de este sistema de control, se tiene en cuenta el uso de una pantalla HP de 18.5 pulgadas, que permita al encargado por medio de esta pantalla visualizar la información de cada alumno y de cada equipo que entra y sale del almacén del laboratorio, por lo siguiente la pantalla HP tiene las siguientes características:

Figura 3.14. Monitor HP.



Fuente. Página web tienda de computo Perú [28].

➤ **Características técnicas Monitor HP.**

Tabla 3.8. Características técnicas del monitor HP.

Ítems	Descripción	
Pantalla	Tamaño	18.5"
	Tipo	Led
	Proporción	Wide
	Resolución Max	1366 x 768
Características Ópticas	Contraste	600:1
	Contraste dinámico	5000000:1
	Brillo	200 CD/M2
Tamaño de pixeles	0.300 x 0.300	
Angulo de visión	90° / 65°	
Entradas / Salidas	D-SUB VGA (15 pines)	
Tiempo de respuesta	5ms	
Voltaje de Alimentación	Auto voltaje (110 – 220 VAC)	
Contenido	Cable de poder, Cable VGA, Guía de usuario, Monitor	

Fuente. Página web tienda de computo Perú [28].

3.1.1.11. Switch Cisco Catalyst 2960C.

Un switch es un dispositivo digital lógico que sirve para conectar varios elementos dentro de una misma red. Estos pueden ser una PC, una impresora, o cualquier aparato electrónico que contenga una tarjeta Ethernet o wifi. En este sistema de control para el registro de préstamos de equipos a los estudiantes se tiene en cuenta contar con un dispositivo que se encargue de hacer la conexión entre la parte electrónica y la base de datos, es por ello que el switch para este sistema tiene las siguientes características:

➤ *Características técnicas Switch Cisco Catalyst 2960C.*

Tabla 3.9. Características Técnicas del Switch Cisco Catalyst 2960C.

Ítems	Descripción
Tipo de dispositivo:	Conmutador - 12 puertos – Gestionado
Tipo incluido	Sobremesa, montaje en rack, montaje en pared - 1U
Tamaño De Tabla De Dirección Mac	8K DE ENTRADAS
Flash Memory:	64 MB
Standard Memory:	128 MB
Rendimiento	Velocidad de reenvío : 10 Gbps
Capacidad	Interfaces virtuales (VLAN) : 255
Admite carcasa Jumbo	9018
Protocolo de direccionamiento	RIP-1, RIP-2
Protocolo de gestión remota	SNMP 1, SNMP 2, RMON 1, RMON 2, RMON 3, RMON 9, Telnet, SNMP 3, SNMP 2c, HTTP, CLI
Puertos	12 x 10/100 + 2 x Gigabit SFP compartido
Alimentación	Poe - CA 120/230 V (50/60 Hz)
Características	Soporte de DHCP, alimentación mediante Ethernet (PoE), negociación automática, soporte BOOTP, soporte ARP, soporte VLAN, señal ascendente automática (MDI/MDI-X automático), snooping IGMP, soporte para Syslog, soporte DiffServ, Broadcast Storm Control, soporte IPv6, Multicast Storm Control, Unicast Storm Control, admite Spanning Tree Protocol (STP), snooping DHCP, soporte de Dynamic Trunking Protocol (DTP), soporte de Port Aggregation Protocol (PAgP), soporte de Trivial File Transfer Protocol (TFTP), soporte de Access Control List (ACL), Quality of Service (QoS), Dynamic ARP Inspection (DAI), STP Root Guard, Uni-Directional Link Detection (UDLD), sin ventilador, relé DHCP, Protocolo de control de adición de enlaces (LACP), Remote Switch Port Analyzer (RSPAN)
Protocolo de gestión remota	SNMP 1, SNMP 2, RMON 1, RMON 2, RMON 3, RMON 9, Telnet, SNMP 3, SNMP 2c, HTTP, HTTPS, TFTP, SSH
Cumplimiento de normas	IEEE 802.3, IEEE 802.3u, IEEE 802.3z, IEEE 802.1D, IEEE 802.1Q, IEEE 802.3ab, IEEE 802.1p, IEEE 802.3af, IEEE 802.3x, IEEE 802.3ad (LACP), IEEE 802.1w, IEEE 802.1x, IEEE 802.1s, IEEE 802.1ae, IEEE 802.1ab (LLDP)
Peso	26.9 cm x 23.8 cm x 4.4 cm
Dimensiones (Ancho x Profundidad x Altura)	26.9 cm x 23.8 cm x 4.4 cm

Fuente. Página web ds3comunicaciones [29].

3.1.1.12. Etiquetas RFID.

Una etiqueta RFID es un dispositivo pequeño, que puede ser adherido o incorporada a un producto, animal o persona. Contienen antenas para permitirles recibir y responder a peticiones por radiofrecuencia desde un emisor/receptor RFID.

Para este sistema se utilizarán etiquetas RFID pasivas, puesto que cada etiqueta de cada equipo irá adherida pegada a él.

Figura 3.15. Etiqueta RFID.



Fuente. Propia de los autores.

3.2. Diseño del Software

Para el diseño del software se tuvieron en cuenta dos partes importantes: la programación de los controladores Arduino (Mega-Uno) para el Módulo Electrónico y la programación del Software de Aplicación (Visual Basic - Base de datos) para la PC de control.

Se presenta a continuación para la programación del Arduino, el IDE del Arduino, el diagrama de flujo del controlador principal, el Arduino Mega y el diagrama de flujo del Arduino Uno puente de comunicación entre la parte electrónica y la PC de control, a través del shield Arduino Ethernet, y por último se hace mención del software de aplicación hecho en Visual Basic

con su respectivo diagrama de flujo de como interactúa con la parte electrónica y la base de datos en SQL Server.

3.2.1. Programación del Arduino

La parte electrónica de este sistema se programó a base de la plataforma Arduino en la que este mismo tiene su propio entorno de desarrollo, puesto que existen otros IDE para el Arduino, el IDE del propio Arduino es muy básico y funcional, además hay desarrolladores de esta plataforma que facilitan librerías para diferentes tipos de funcionalidades y control que a continuación se mostraran.

3.2.1.1. Entorno de desarrollo integrado (IDE) Arduino.

Un IDE del Arduino es un entorno de programación que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica, además de permitir cargar el programa ya compilado en la memoria flash del microcontrolador de Arduino a través del puerto serial de la computadora.

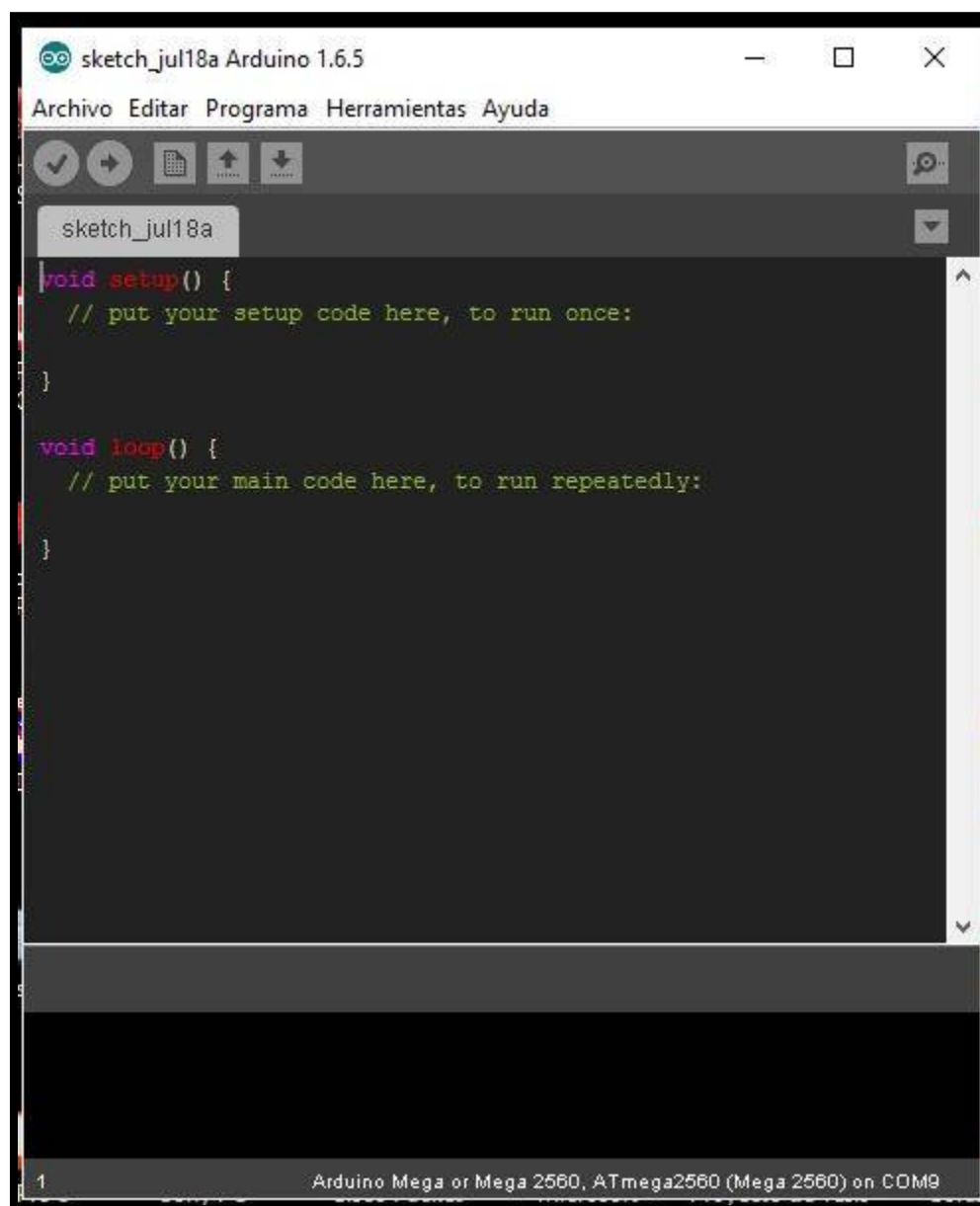
En este sistema se necesitó el uso de un IDE que permitiera escribir y editar el código (también llamado sketch donde sobre él se escribirá todo el comportamiento que tendrá este proyecto) que nos permita comprobar que no hayamos cometido ningún error de código y que además nos permita, cuando ya estemos seguros de que el sketch es correcto después de compilarlo, grabarlo en la memoria del microcontrolador de la placa Arduino para que este se convierta a partir de entonces en el ejecutor autónomo de dicho programa.

Un sketch está estructurado en dos partes o bloques:

El bloque setup (): se ejecuta una sola vez cuando comienza el programa. Es el método encargado de recoger la configuración. Se suele emplear para determinar si un determinado ‘pin’ es salida o entrada, establecer su valor inicial, e inicializar el puerto serie.

El bloque loop (): se ejecuta de forma ininterrumpida, una y otra vez. Con este bucle logramos que nuestro programa responda ante los distintos eventos que se produzcan en el proyecto.

Figura 3.16. Sketch del Arduino.



Fuente: Propia de los autores

3.2.1.2. Librerías del Arduino usadas en el sistema

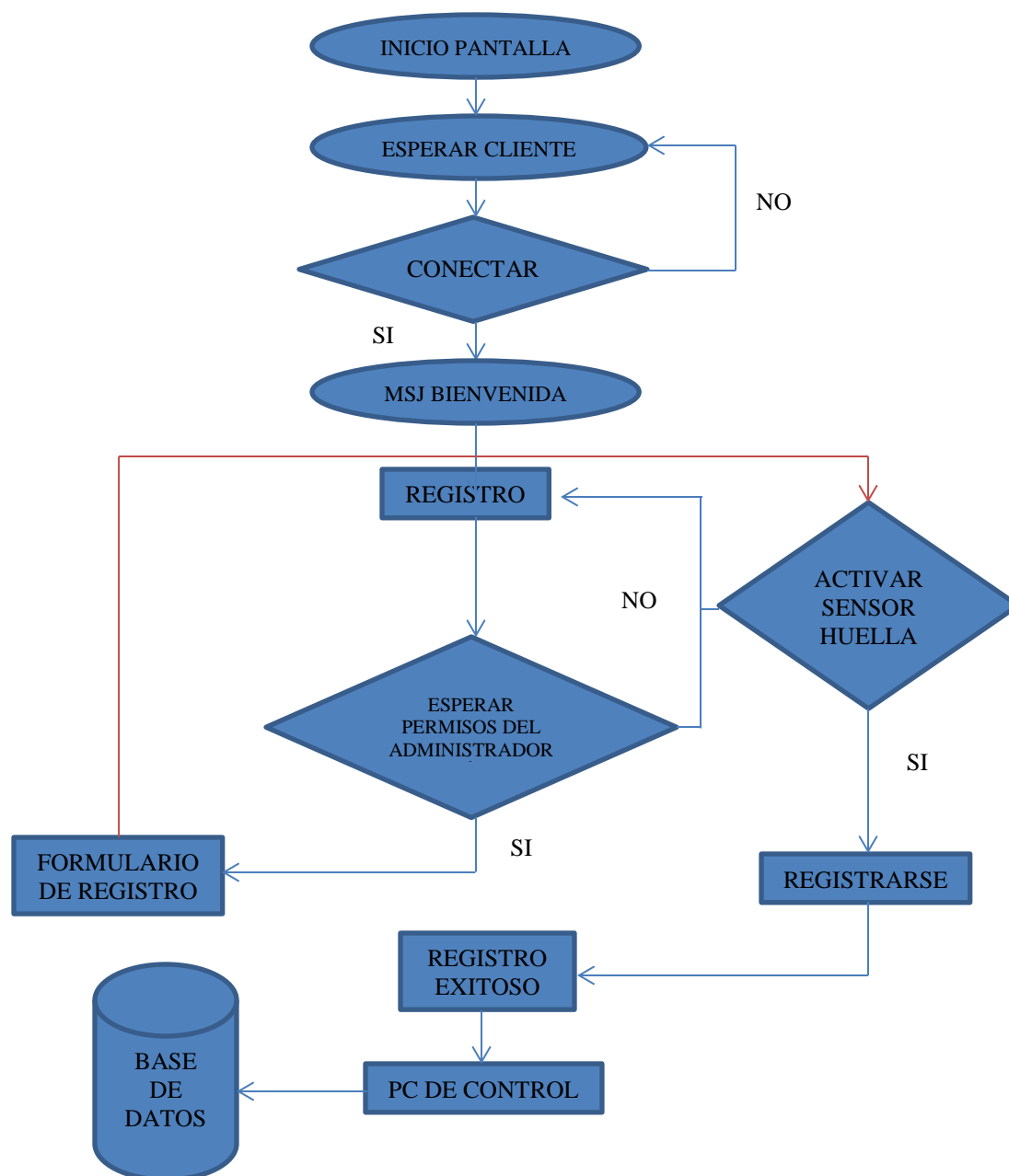
Tabla 3.10. Librerías usadas en el Arduino.

N°	LIBRERÍA	DESCRIPCIÓN	UTILIDAD
1	UTFT	Para la visualización	Arduino Mega 2560
2	Utouch - URtouch	Para el manejo de la parte táctil de la pantalla	Arduino Mega 2560
3	UTFT_Buttons	Para usar botones y extender el uso de las librerías UTFT y URTouch.	Arduino Mega 2560
4	tinyFAT	Es compatible con formato FAT16 tarjetas SD de hasta 2 GB de tamaño. La tarjeta SD se debe conectar a los SPI- pins en la placa Arduino.	Arduino Mega 2560
5	UTFT_tinyFAT	Añade una forma sencilla de cargar las imágenes a las tarjetas SD. Las imágenes deben estar en el formato .raw.	Arduino Mega 2560
6	SD	Para leer y escribir datos en la tarjeta SD del Arduino. Puedes gestionar todos tus datos de forma eficaz.	Arduino Mega 2560
7	Wire	Para el manejo de comunicaciones I2C en Arduino. Esta librería está incluida con las básicas que trae el programa de Arduino.	Arduino Mega 2560
8	SPI	Para la gestión del protocolo SPI en Arduino	Arduino Uno
9	ETHERNET	Permite que Arduino pueda conectarse a internet. Puede servir como un servidor de aceptar conexiones entrantes o un cliente que realiza los salientes.	Arduino Uno

Fuente: Propia de los autores.

3.2.1.3. Diagrama de flujo para el proceso de registro de usuarios.

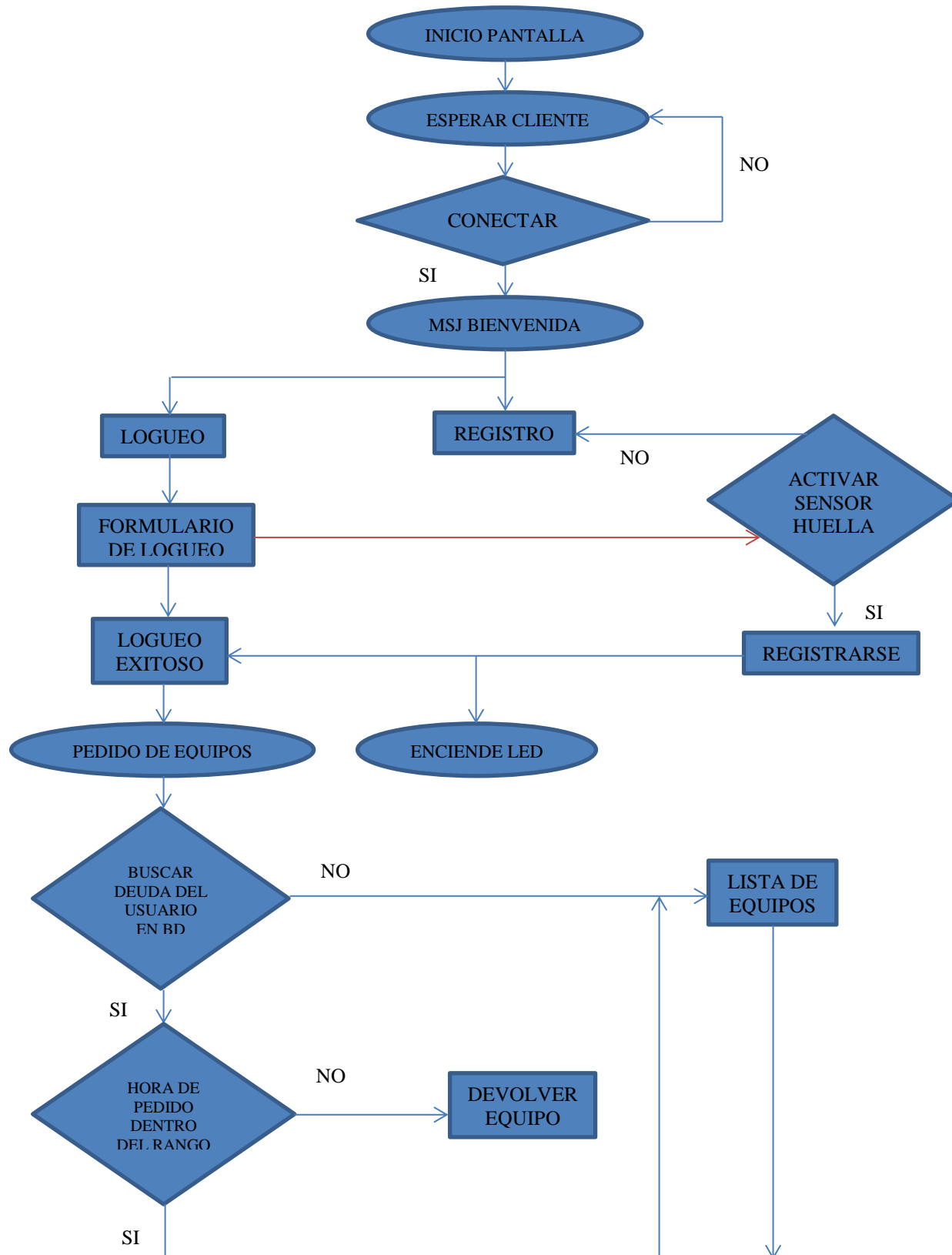
Figura 3.17. Diagrama de flujo para el registro de alumnos.

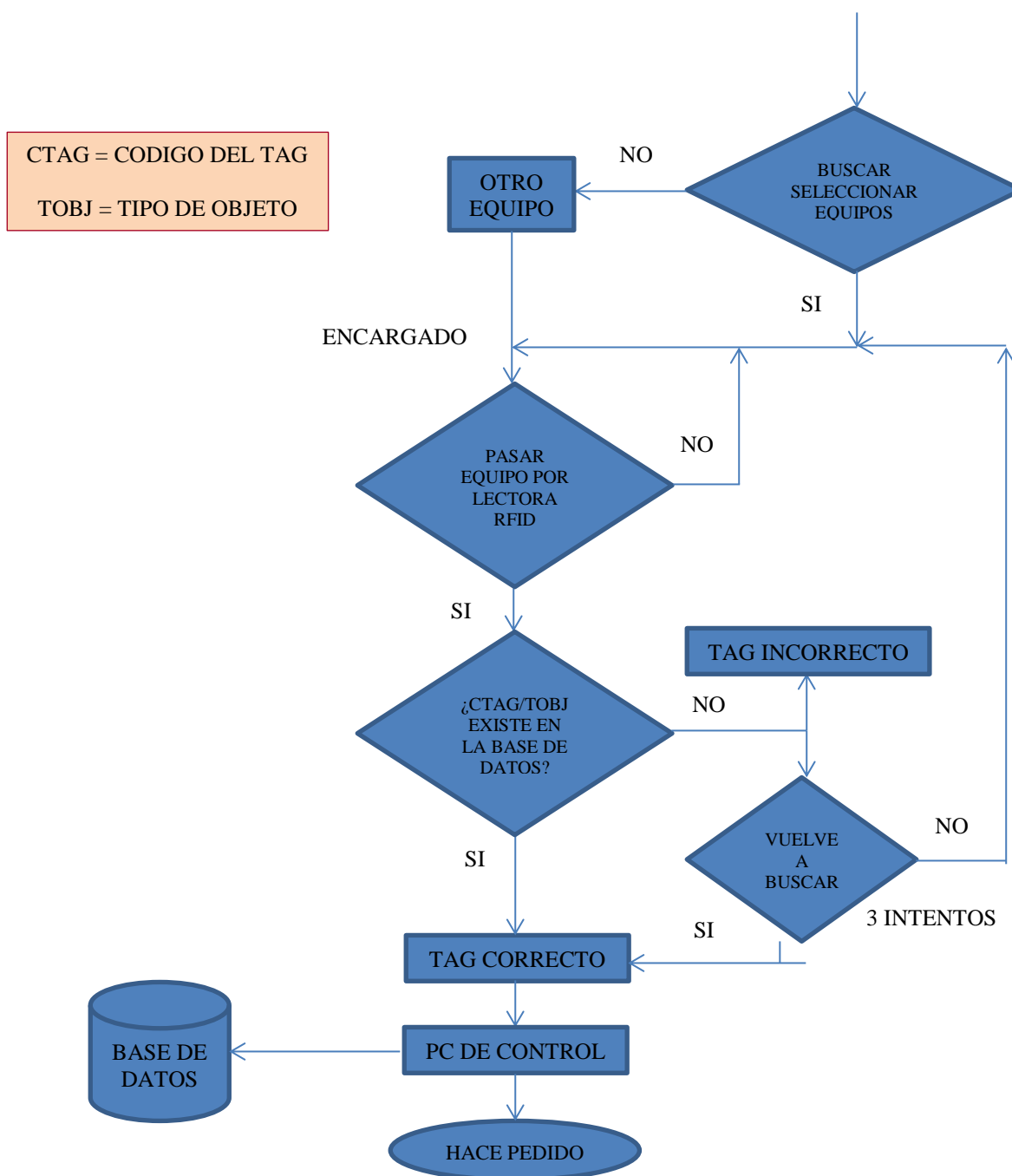


Fuente: Propia de los autores.

3.2.1.4. Diagrama de flujo para el Pedido de Equipos

Figura 3.18. Diagrama de flujo para el pedido de equipos.

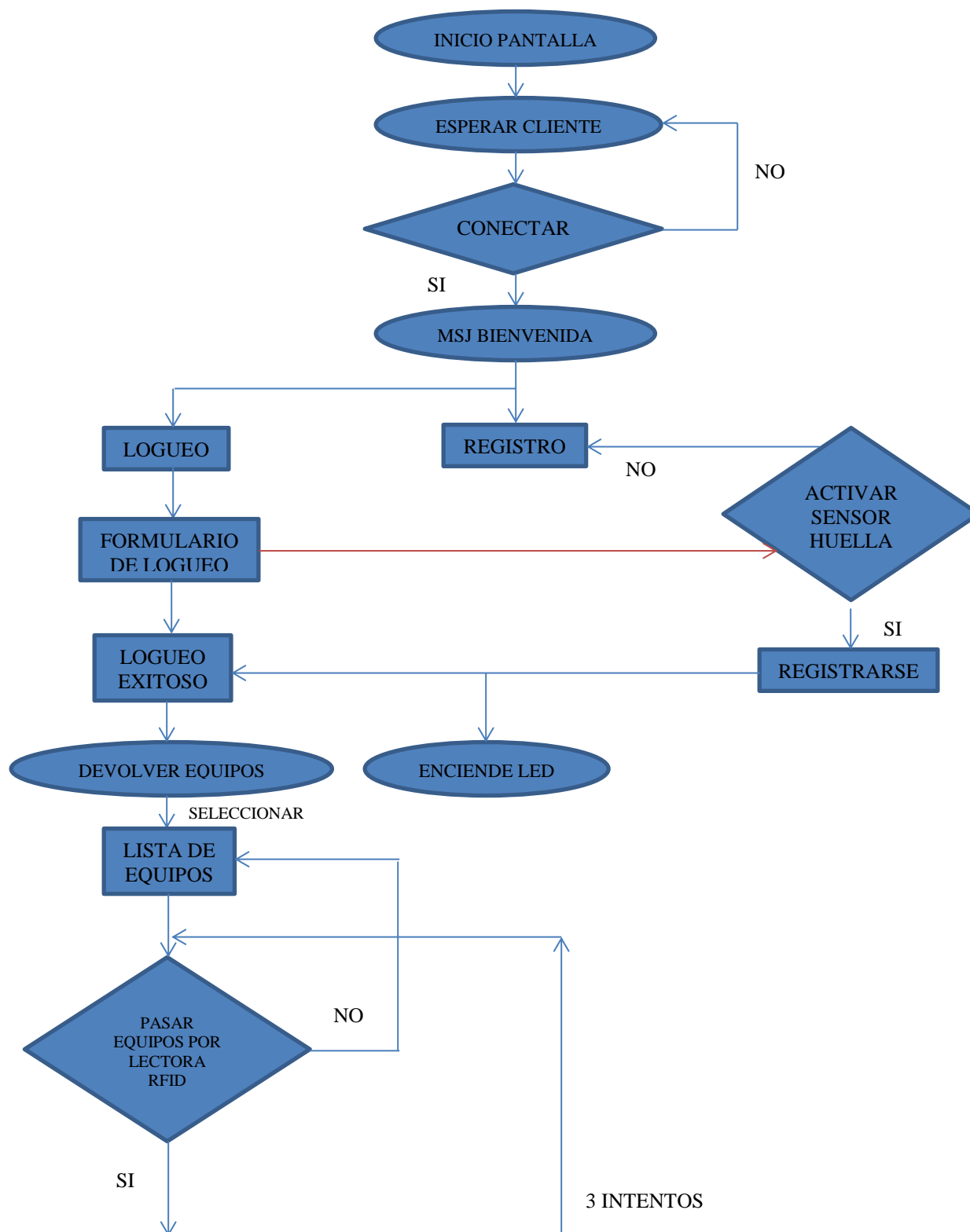


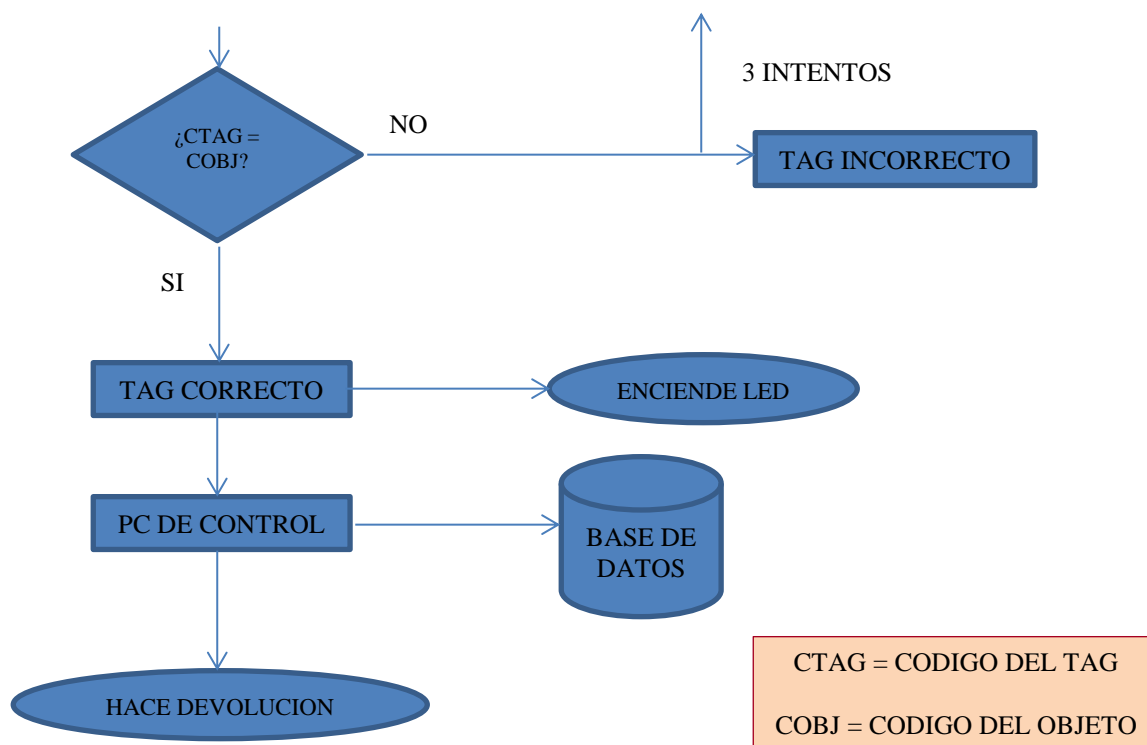


Fuente: Propia de los autores.

3.2.1.5. Diagrama de Flujo para la Devolución de Equipos

Figura 3.19. Diagrama de flujo para devolución de equipos.

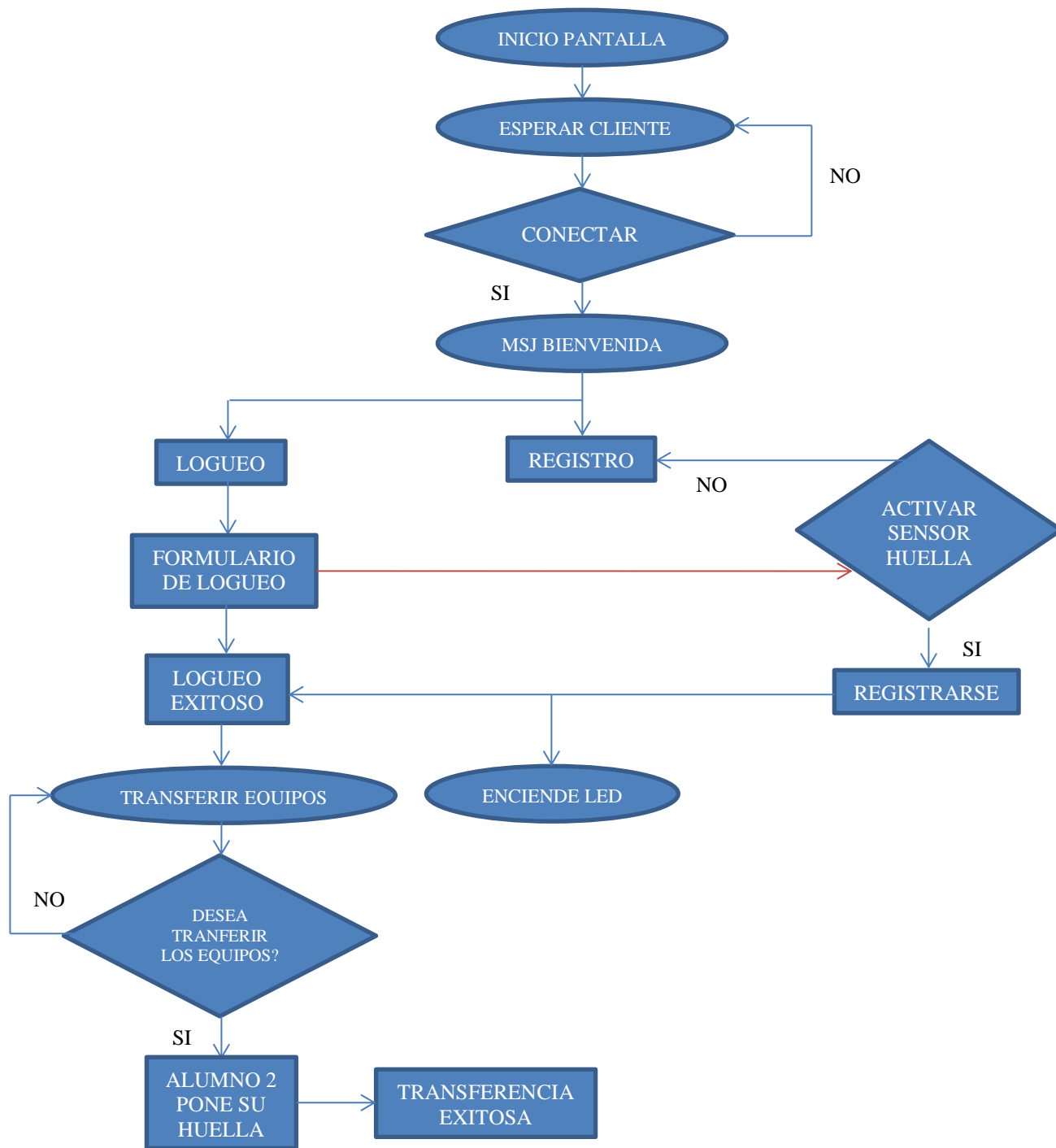




Fuente: Propia de los autores

3.2.1.6. Diagrama de flujo para la transferencia de equipos.

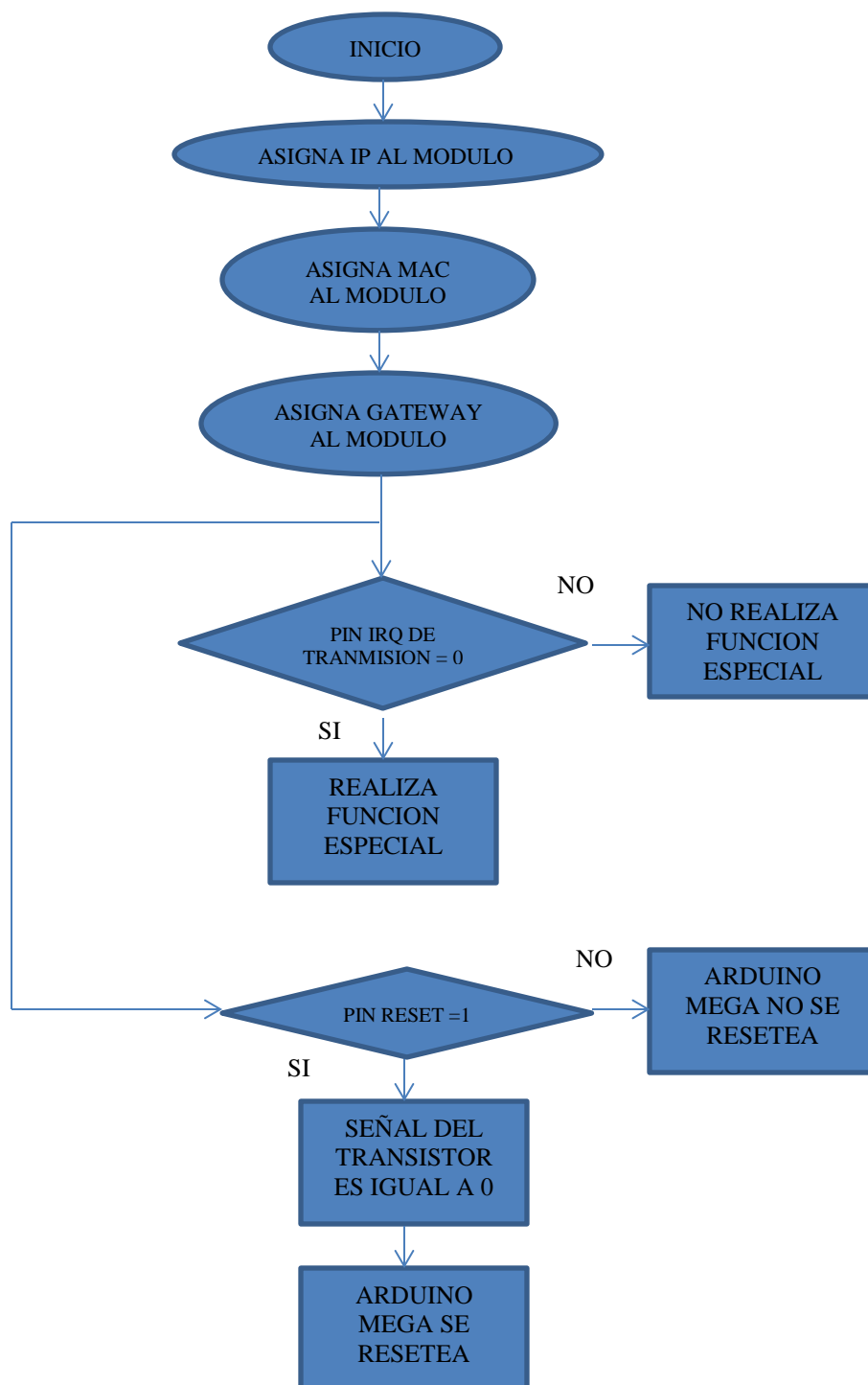
Figura 3.20. Diagrama de flujo para transferencia de equipos.



Fuente: Propia de los autores

3.2.1.7. Diagrama de flujo de la programación del Arduino Uno

Figura 3.21. Diagrama de flujo del Arduino Uno



Fuente: Propia de los autores

El funcionamiento de este algoritmo es simular una interrupción a través de un transistor, puesto que el Arduino puede resetearse por software al momento de cargar un programa, se vio la necesidad de hacer este montaje para sincronizar todo el sistema.

El Arduino Mega al utilizar diferentes tipos de módulos (sensores, pantalla, etc.) al mismo tiempo, dificulta el uso de las interrupciones, debido a que este se sobrecarga.

Existe un pin de aviso de transmisión que cuando el Arduino Uno está en funcionamiento, manda un pulso para resetear el Arduino Mega, este pin de aviso le indica que no realice ninguna de las funciones especiales como por ejemplo borrar huella, cambiar fondo, identificar tarjeta RFID, etc.

El pin Reset es el que envía la señal al transistor para resetear desde el inicio al Arduino Mega.

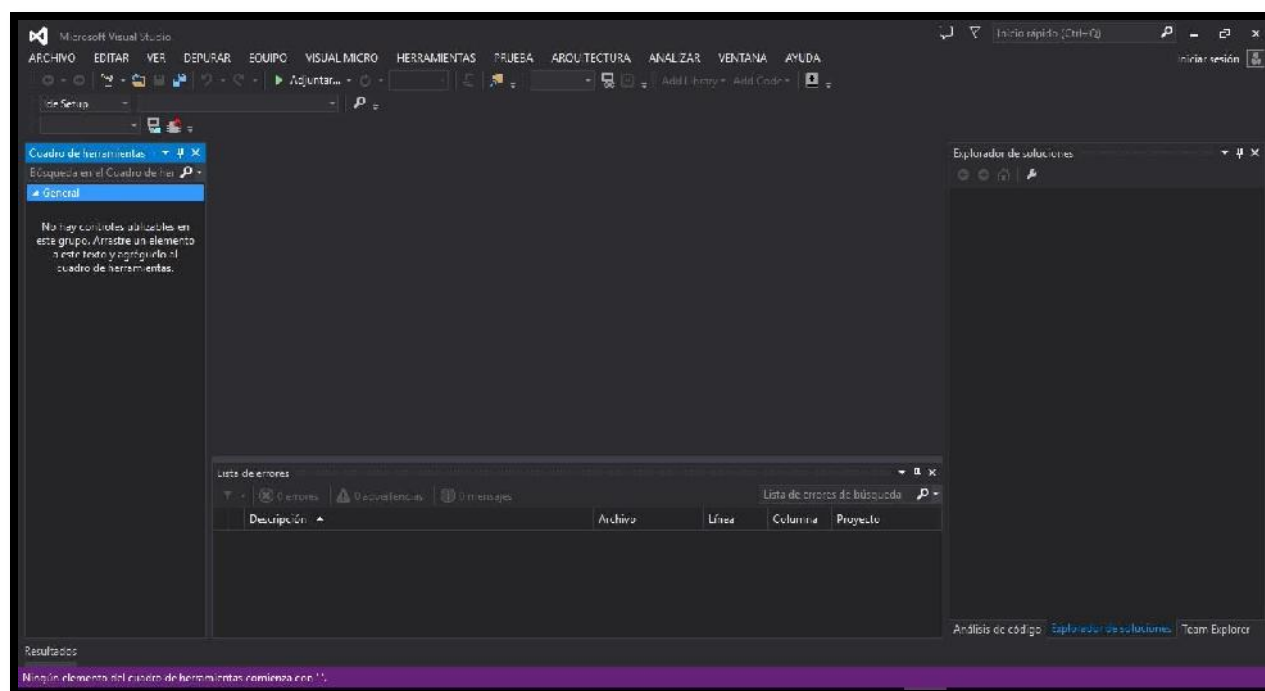
3.2.2. Programación del software de aplicación

Para la programación del software de aplicación se utilizó como entorno de desarrollo el Visual Studio 2013 con el lenguaje de programación Visual Basic.

3.2.2.1. Entorno de desarrollo integrado IDE de Microsoft Visual Studio.

Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP. Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos, consolas, etc.

Figura 3.22. IDE Visual Studio



Fuente: Propia de los autores

3.2.2.2. Librerías del Visual Basic usadas en el sistema.

Tabla 3.11. Librerías para el Visual Basic.

N°	LIBRERÍA	DESCRIPCIÓN	UTILIDAD
1	System.Threading	Espacio de nombres que proporciona clases e interfaces que habilitan la programación multiproceso.	GENERAR RETARDOS Y SUBPROCESOS.
2	System.Net	Espacio de nombres que contiene clases que proporcionan una interfaz de programación sencilla para diversos protocolos de red.	PROTOCOLO TCP
3	System.Data.SqlClient	Espacio de nombres que describe una colección de clases que se utiliza para obtener acceso a una base de datos de SQL Server en el espacio administrado.	SQL SERVER
4	System.Net.Sockets	Espacio de nombres proporciona una implementación administrada de la interfaz Windows Sockets (Winsock) para desarrolladores que necesitan controlar rigurosamente el acceso a la red.	PROTOCOLO TCP
5	System.IO	Espacio de nombres System.IO que permiten leer y escribir en los archivos y secuencias de datos, así como tipos que proporcionan compatibilidad básica con los archivos y directorios.	PUERTOS DE ENTRADA/SALIDA.

Fuente: Propia de los autores.

3.2.2.3. Diagrama de Flujo de la Programación del Visual Basic y el SQL Server.

Para la programación del Visual Basic se hizo con el método de *programación orientada a objetos*; puesto que permitirá una mejor comprensión, orden y manipulación del código, es decir es fácil de hacer un cambio de código. Mientras que para la programación de la base de datos en SQL Server se utilizó el mismo entorno de desarrollo del Visual Studio, es decir el Visual Basic se conecta con el SQL Server a través de la clase SqlConnection, la cual permite por medio de comandos crear base de datos, crear tablas, eliminar tablas, etc. Y se pudo verificar dicha programación a través del SQL Server Management Studio. A continuación se describen algunos comandos básicos utilizados en la programación de la base de datos del sistema.

Definiendo como es almacenada la información tenemos los siguientes comandos:

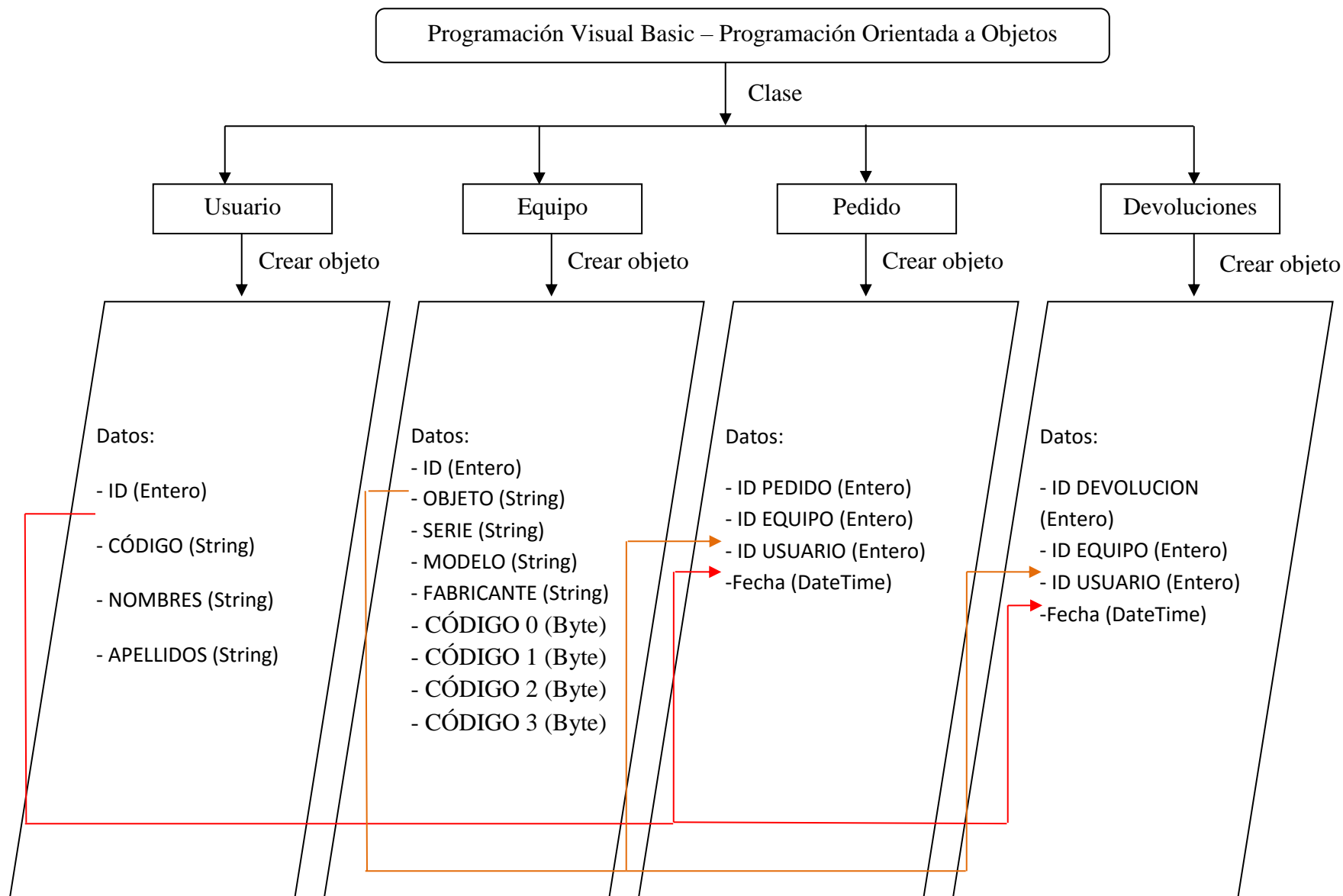
- **CREATE DATABASE:** Se utiliza para crear una nueva base de datos vacía.
- **DROP DATABASE:** Se utiliza para eliminar completamente una base de datos existente.
- **CREATE TABLE:** Se utiliza para crear una nueva tabla, donde la información se almacena realmente.
- **ALTER TABLE:** Se utiliza para modificar una tabla ya existente.
- **DROP TABLE:** Se utiliza para eliminar por completo una tabla existente.

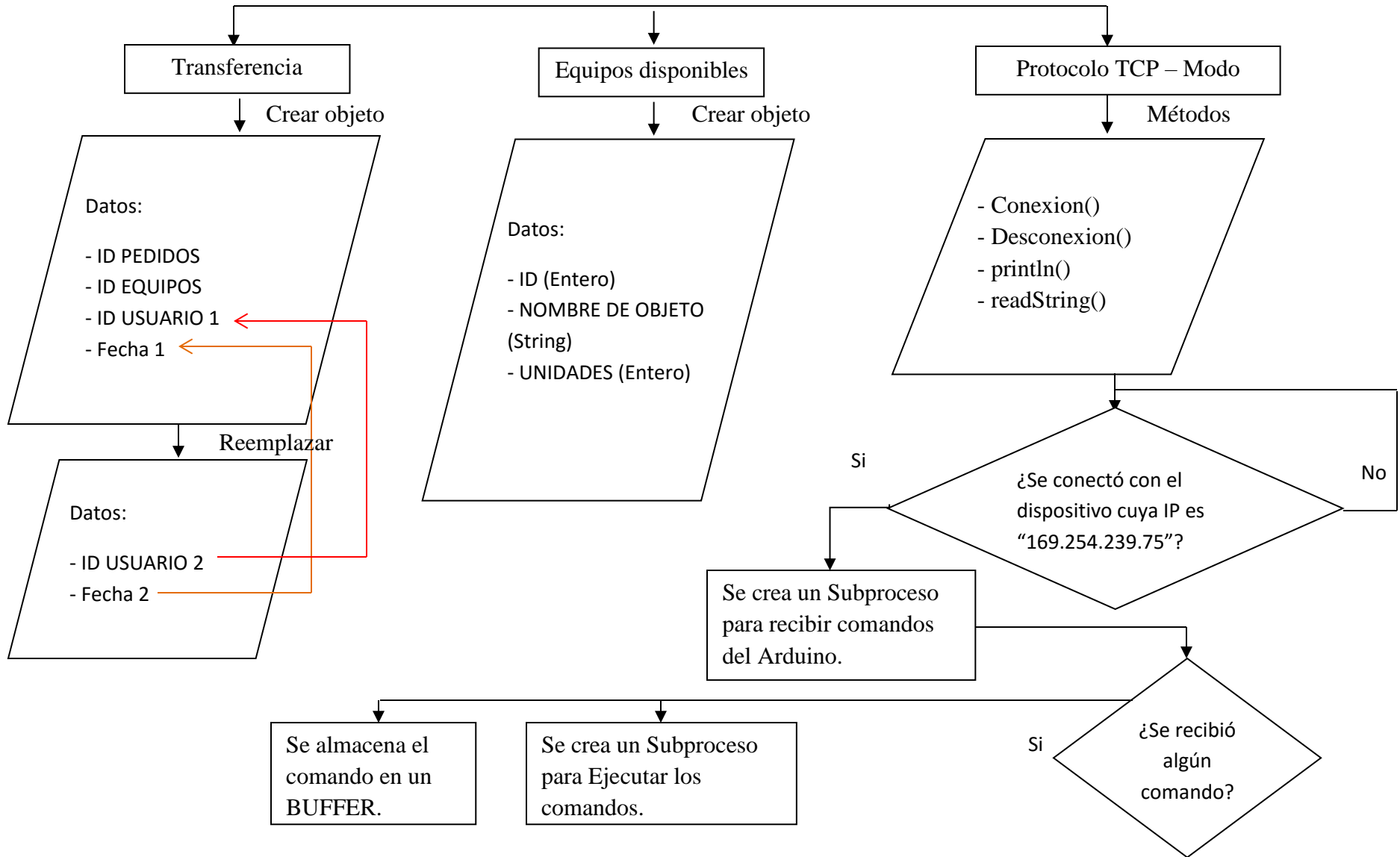
Dependiendo de la manipulación de los datos tenemos los siguientes comandos:

- **SELECT** Se utiliza cuando quieres leer (o seleccionar) tus datos.
- **INSERT:** Se utiliza cuando quieres añadir (o insertar) nuevos datos.
- **UPDATE:** Se utiliza cuando quieres cambiar (o actualizar) datos existentes.
- **DELETE:** Se utiliza cuando quieres eliminar (o borrar) datos existentes.
- **REPLACE:** Se utiliza cuando quieres añadir o cambiar (o reemplazar) datos nuevos o ya existentes.
- **TRUNCATE:** Se utiliza cuando quieres vaciar (o borrar) todos los datos de la plantilla.

A continuación se muestra el diagrama de flujo de la programación del Visual Basic con su respectiva información para las tablas de la base de datos.

Figura 3.23. Diagrama de flujo de la programación del Visual Basic y SQL Server





Fuente: Propia de los autores

3.2.2.4. Resumen de los comandos principales para la aplicación en Visual Basic.

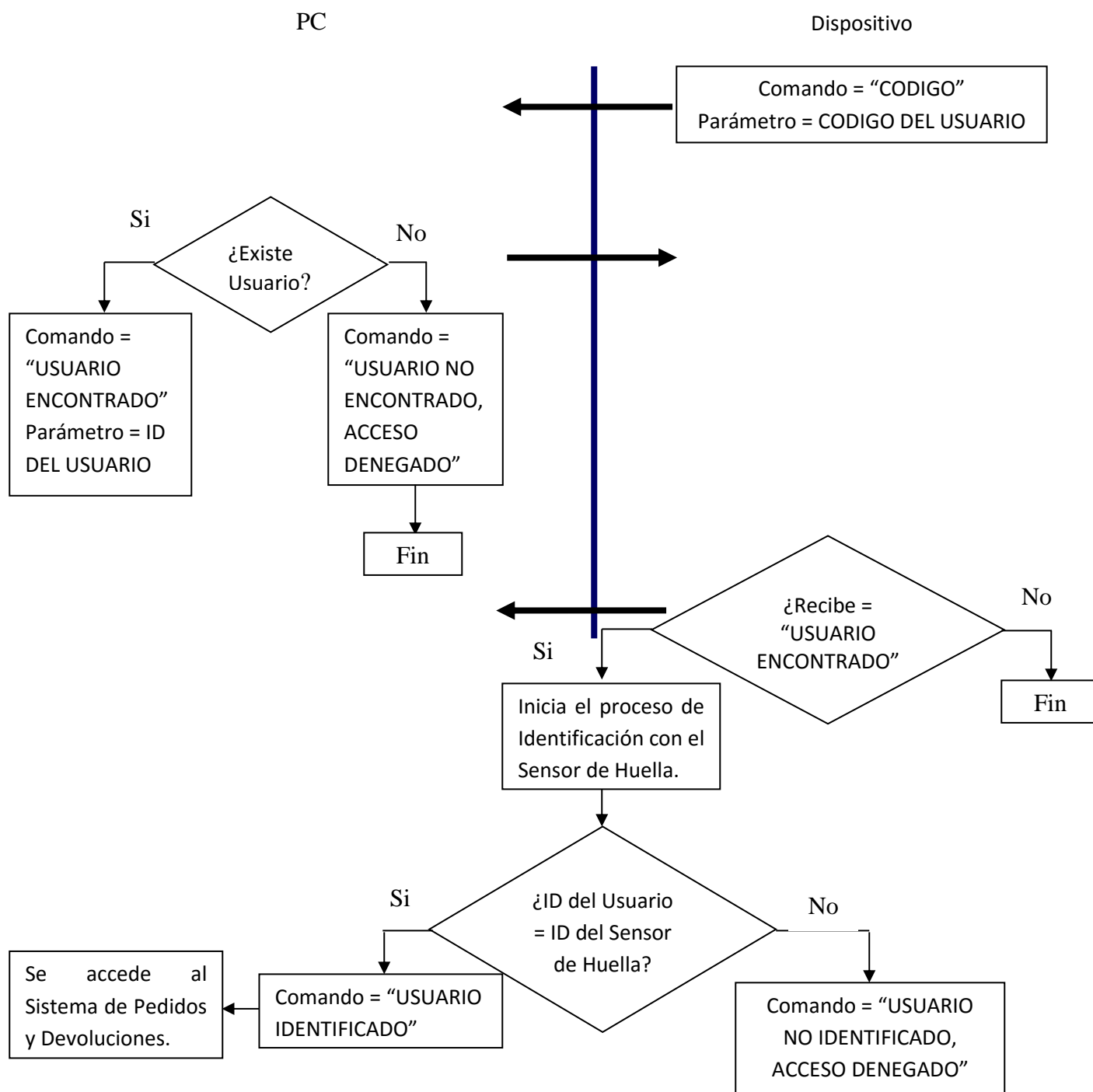
Tabla 3.12. Comandos para el programa en Visual Basic.

PC Host (Programa de Visual Basic)	
COMANDOS DE COMUNICACIÓN	DESCRIPCIÓN
CODIGO USUARIO	Identifica si el Código del Usuario ingresado existe en la base de datos de SQL Server, y si existe procede a comprobar la Huella Dactilar del Usuario.
IDENTIFICAR SENSOR DE HUELLA	Identifica si el Sensor de Huella Dactilar está conectado correctamente y especifica las características técnicas de éste.
BORRAR HUELLAS	Borra toda la base de datos de las Huellas grabadas en la Memoria EEPROM del Sensor de Huella Dactilar.
GRABAR HUELLA	Graba la Huella del Usuario con un ID específico.
REGISTRO	Registra el Código, Nombres y Apellidos del Usuario con un ID específico.
IDENTIFICAR R F I D	Identifica si el Lector RFID está conectado correctamente y especifica las características técnicas de éste.
LEER ID TARJETA	Lee el ID de la Tarjeta o Tag colocado en el Lector RFID, el cual consta de 4 bytes.
CLAVE R F I D	Busca la Clave ó ID de la Tarjeta RFID asignada a un Objeto en la Base de SQL Server y se lo envía al Dispositivo Arduino.
LISTA MATERIALES	Busca la Lista de Materiales Disponibles en la Base de Datos de SQL Server y se lo envía al Dispositivo Arduino.
LISTA PENDIENTES	Busca la Lista de Materiales Pendientes para devolver de un Usuario específico en la Base de Datos de SQL Server y se lo envía al Dispositivo Arduino.
OBJETO RECONOCIDO	Si un Objeto ha sido reconocido por el Lector RFID se reproduce un Sonido de Alarma en la PC.
ELIMINAR OBJETO DE LA LISTA	Si un objeto ha sido devuelto, con este comando se procede a eliminarlo de la Lista de Materiales Pendientes para devolver.
AGREGAR OBJETO A LA LISTA	Si un objeto ha sido devuelto, con este comando se procede a eliminarlo de la Lista de Materiales Pendientes para devolver.
OTRO EQUIPO	Con este comando el usuario puede hacer un pedido especial digitando el nombre del Objeto específico.

Fuente: Propia de los autores

3.2.2.5. Comando: “CODIGO USUARIO”.

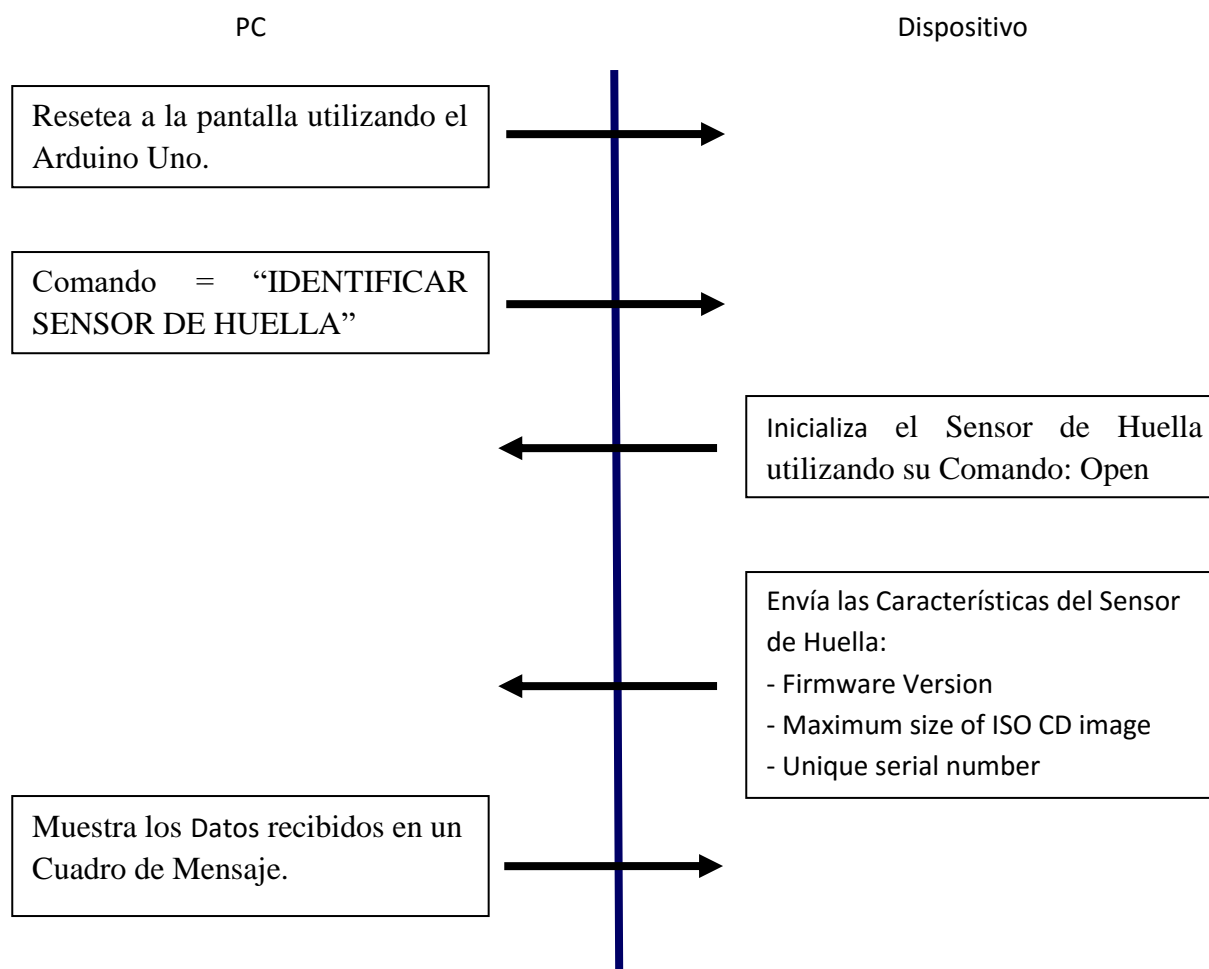
Figura 3.24. Comando Código de usuario.



Fuente: Propia de los autores

3.2.2.6. Comando: “IDENTIFICAR SENSOR DE HUELLA”.

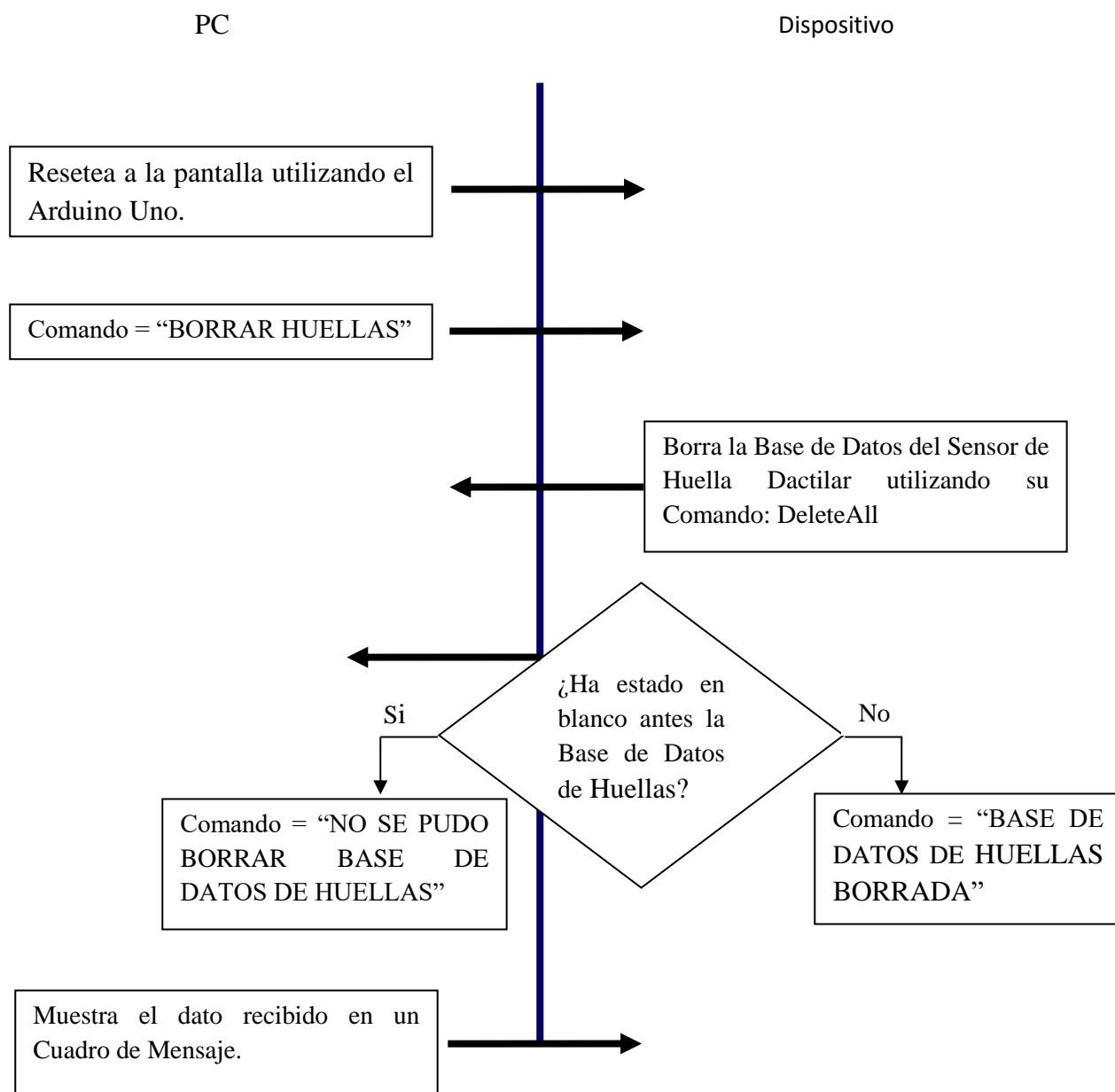
Figura 3.25. Comando Identificar Sensor de Huella.



Fuente: Propia de los autores

3.2.2.7. Comando: “BORRAR HUELLAS”.

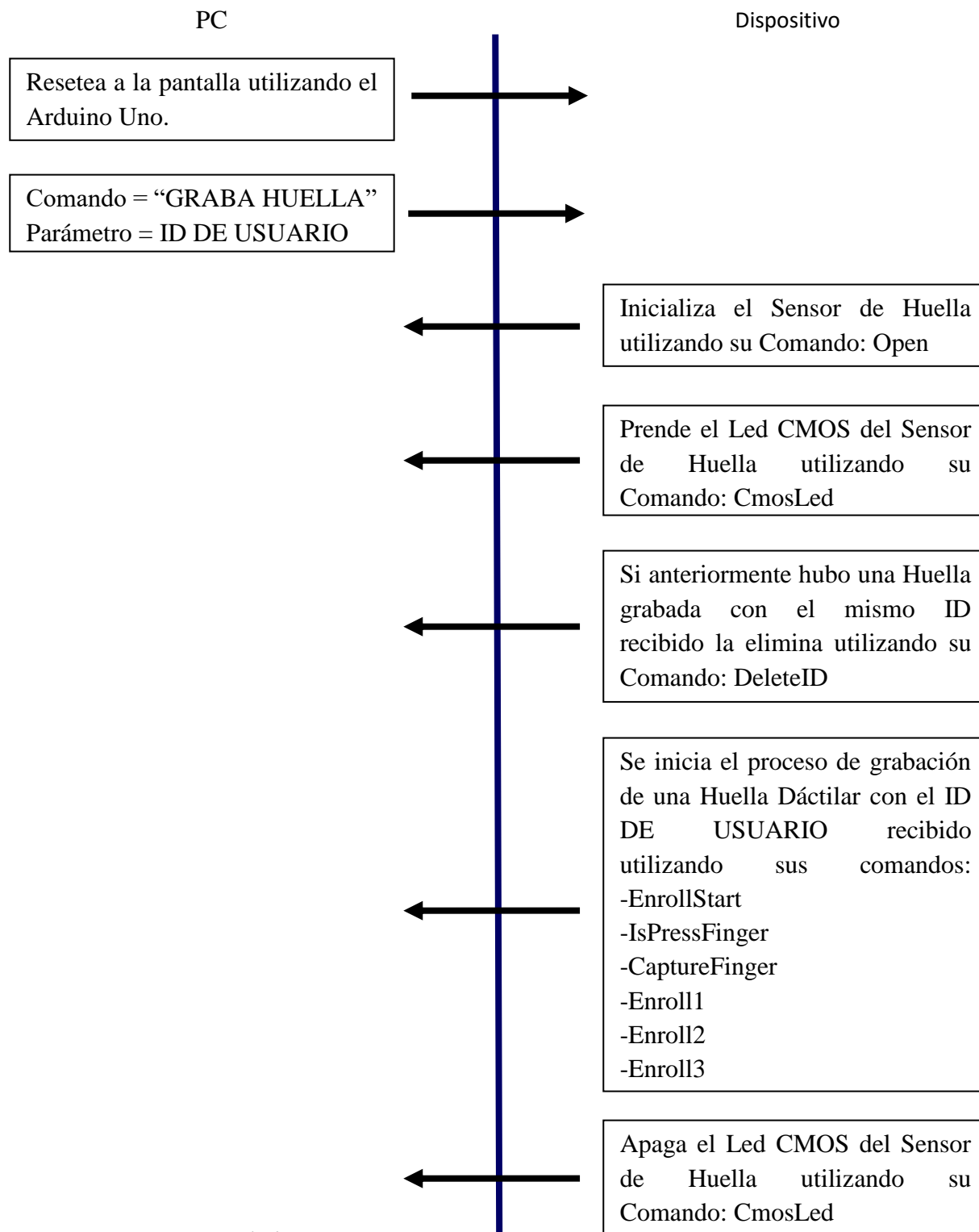
Figura 3.26. Comando Identificar Borrarr Huella.



Fuente: Propia de los autores

3.2.2.8. Comando: “GRABAR HUELLA”.

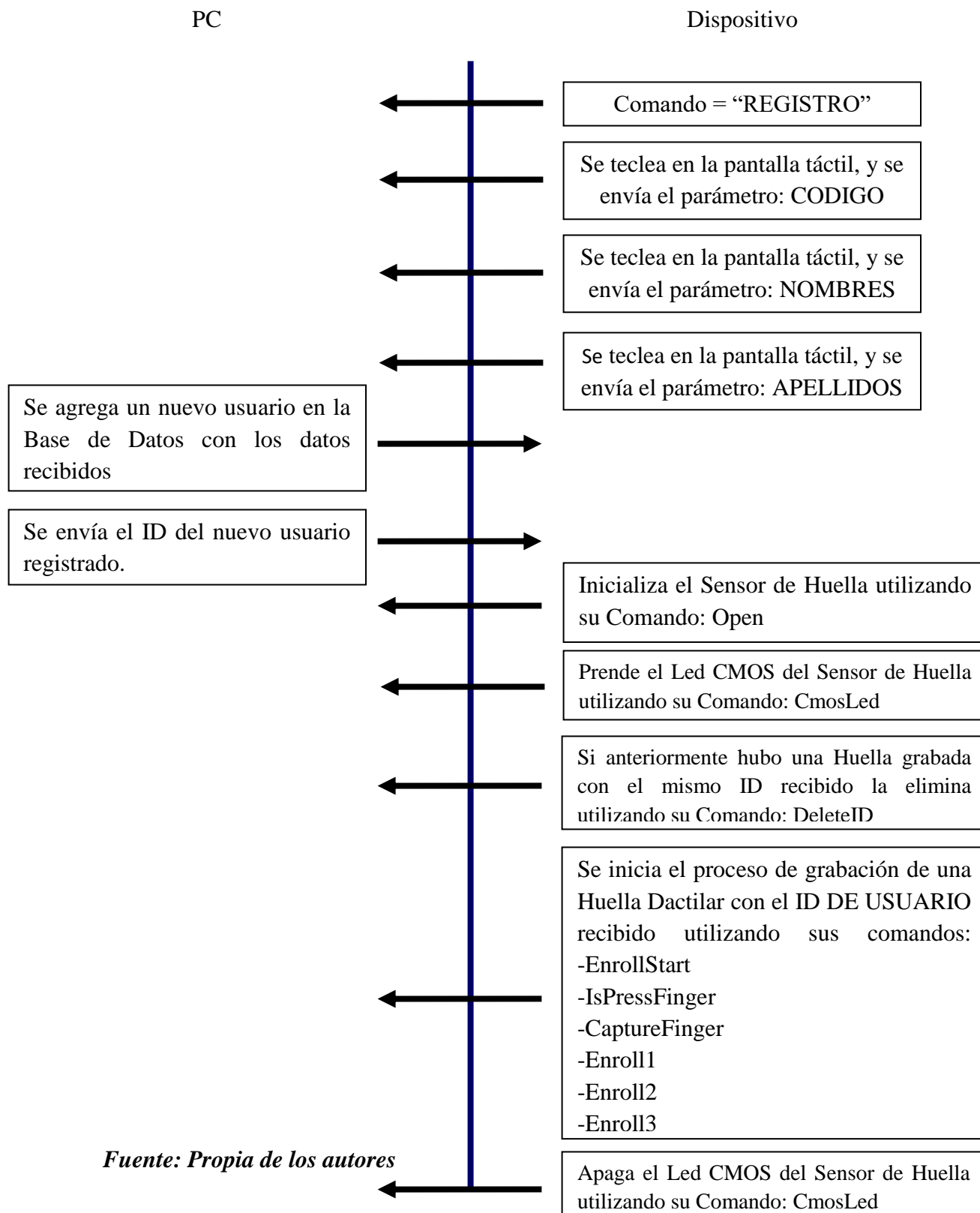
Figura 3.27. Comando Grabar Huella.



Fuente: Propia de los autores

3.2.2.9. Comando: “REGISTRO”.

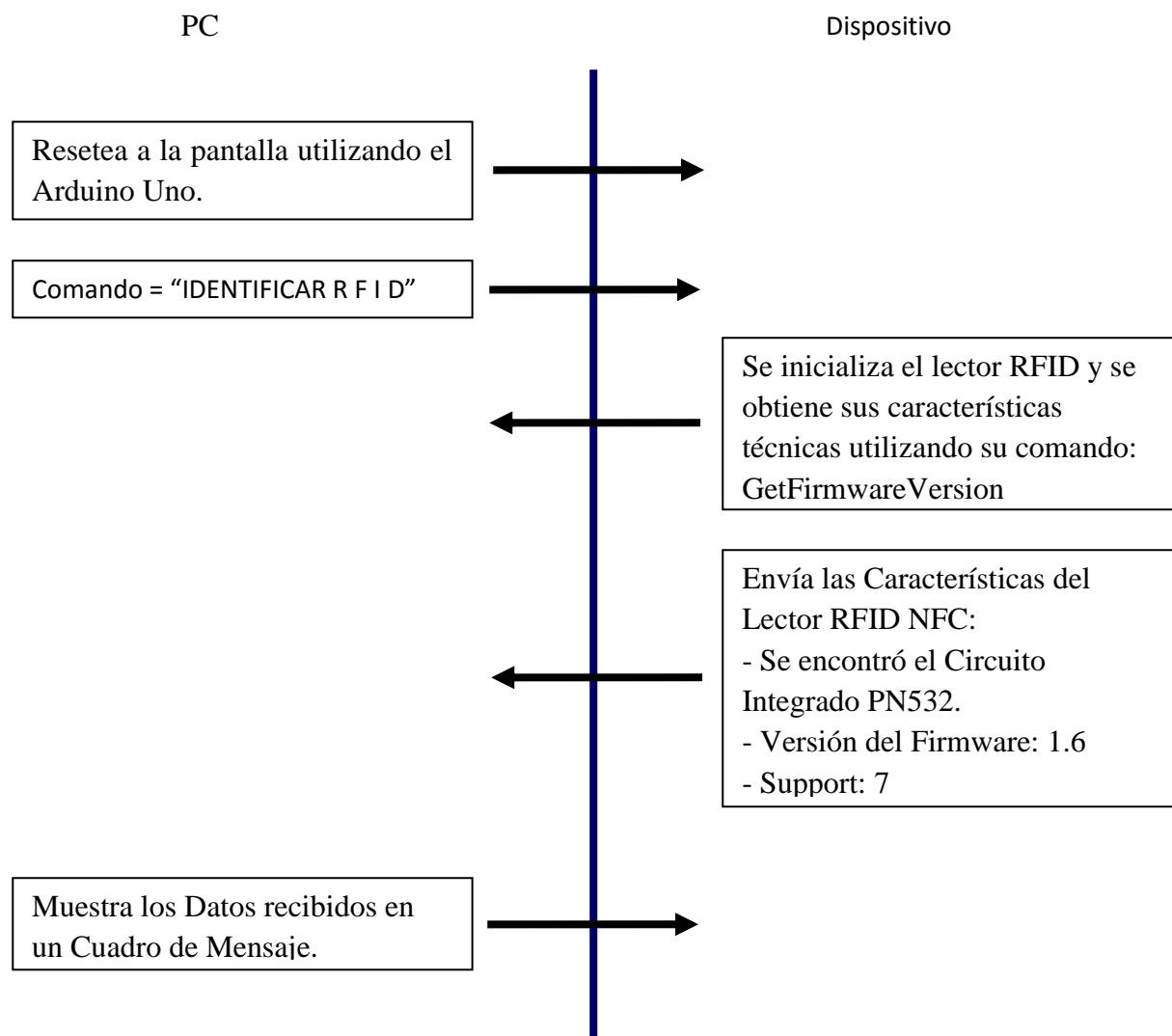
Figura 3.28. Comando Registro.



Fuente: Propia de los autores

3.2.2.10. Comando: “IDENTIFICAR R F I D”.

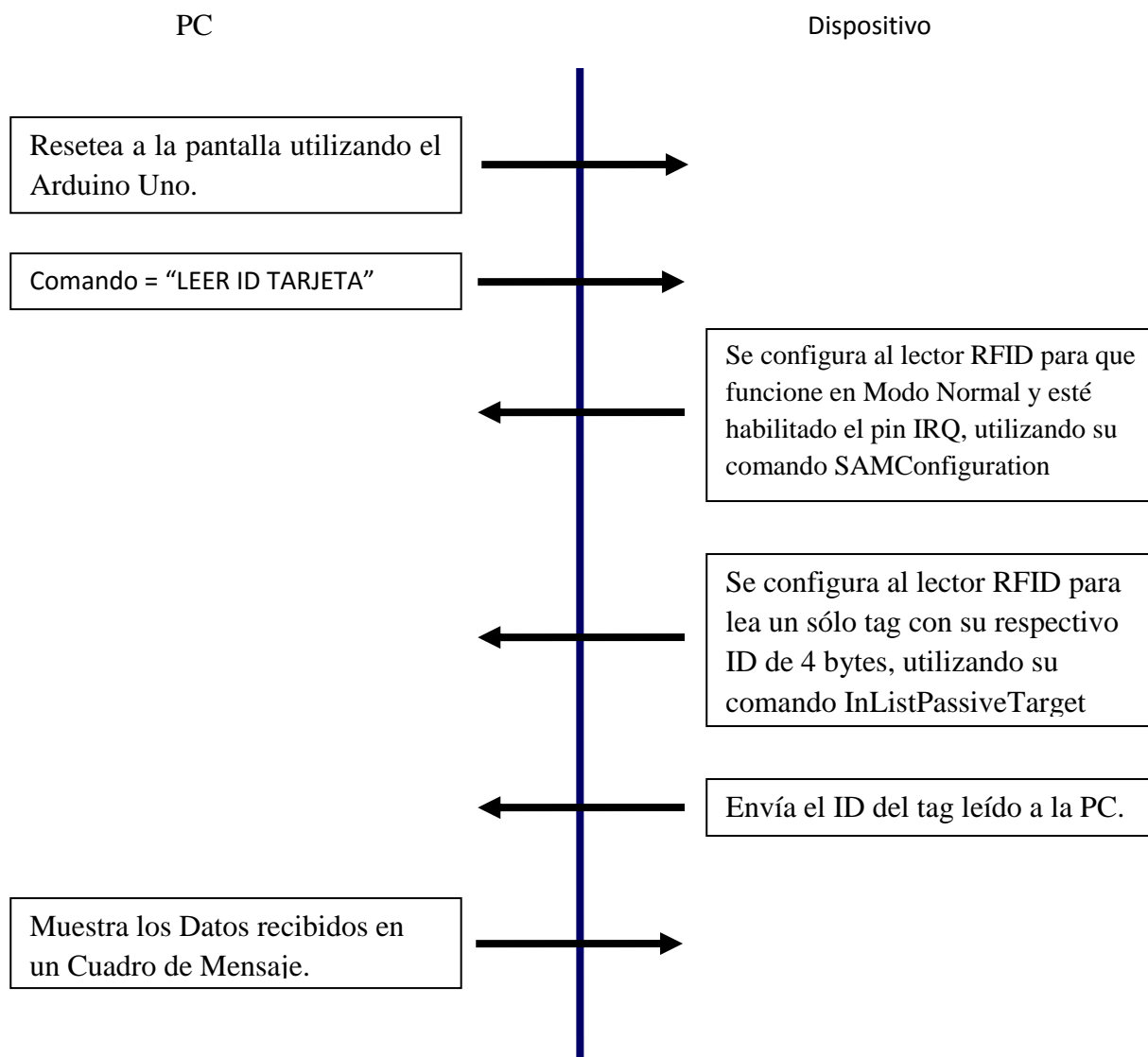
Figura 3.29. Comando Identificar RFID.



Fuente: Propia de los autores

3.2.2.11. Comando: “LEER ID TARJETA”.

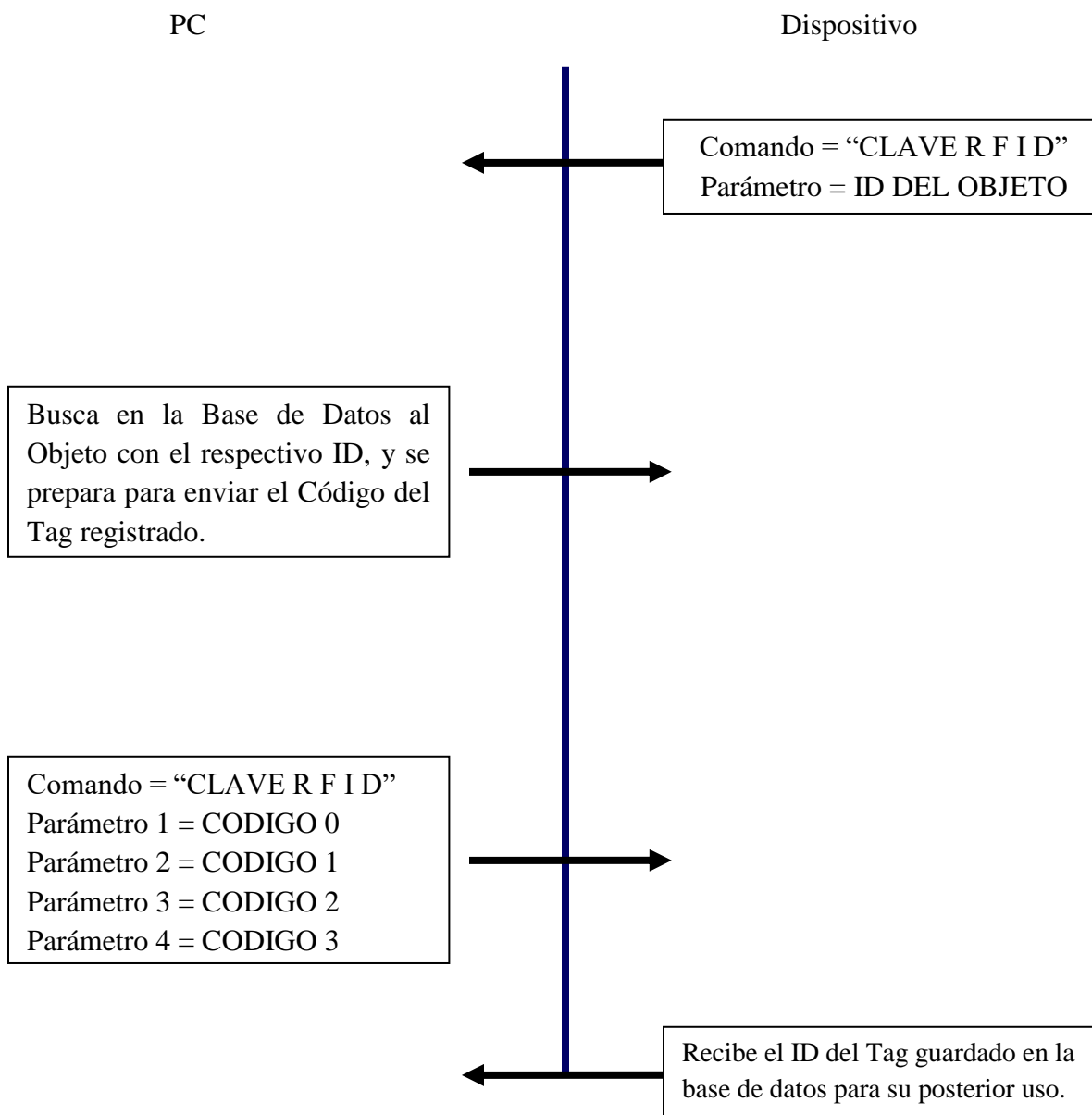
Figura 3.30. Comando Leer ID Targeta.



Fuente: Propia de los autores

3.2.2.12. Comando: “CLAVE R F I D”.

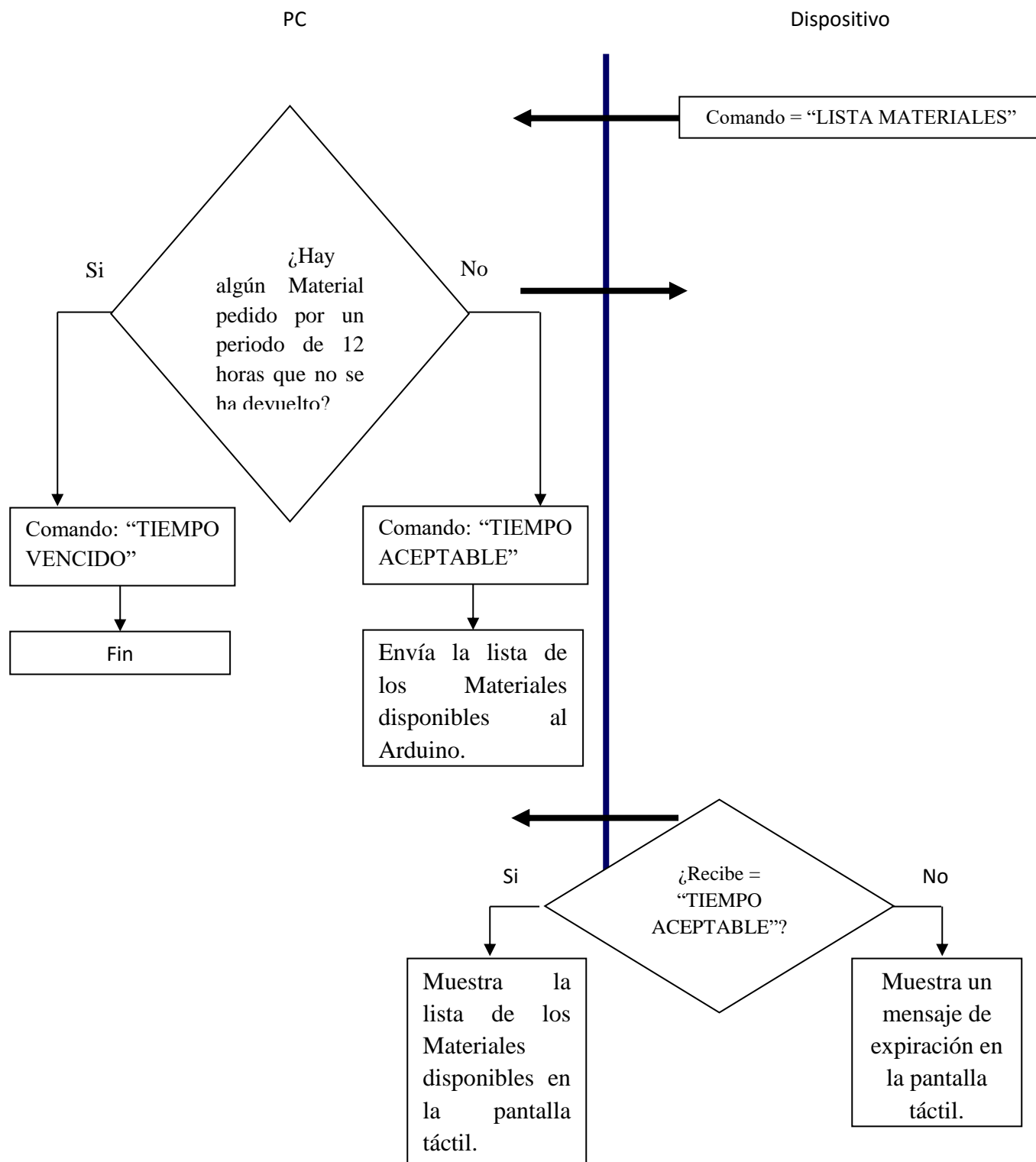
Figura 3.31. Comando Clave RFID.



Fuente: Propia de los autores

3.2.2.13. Comando: “LISTA MATERIALES”.

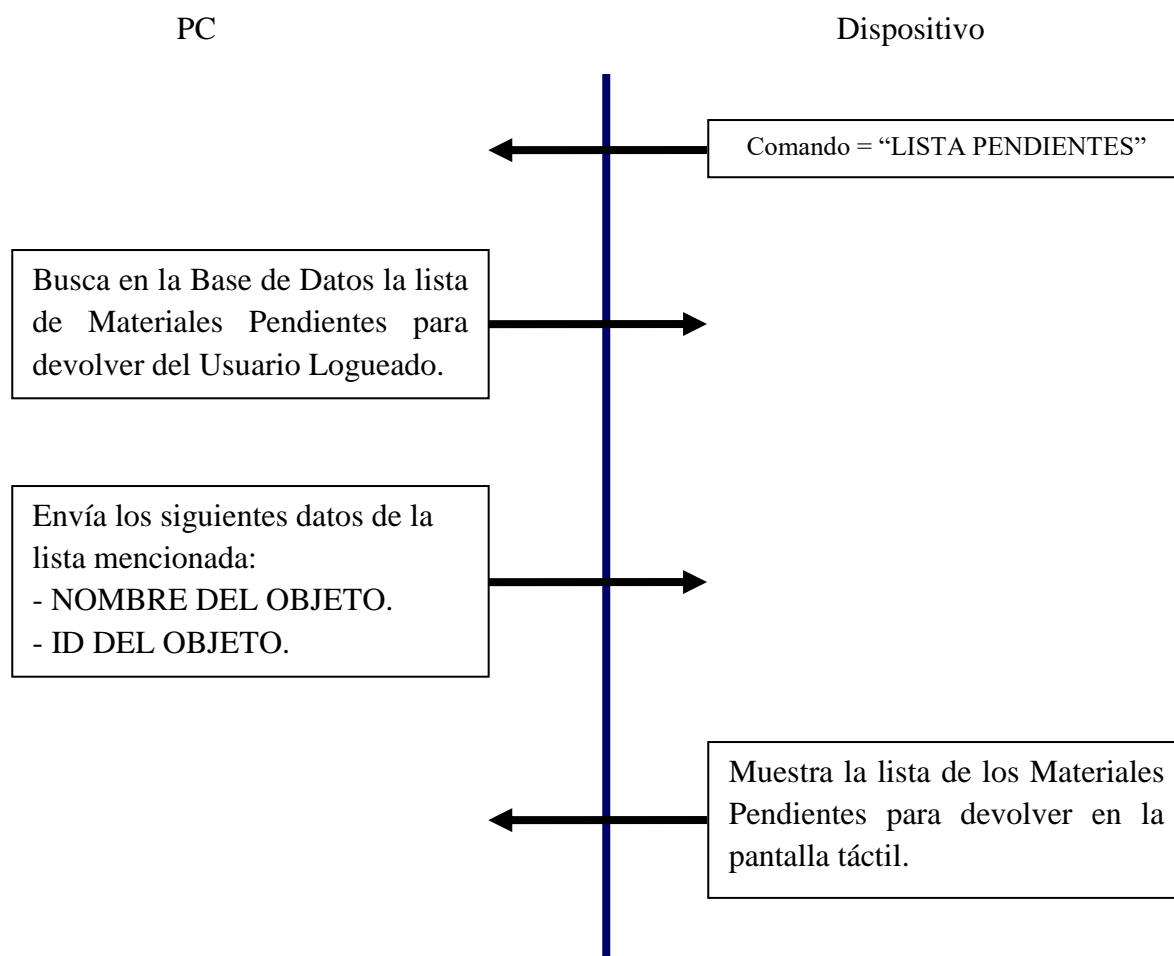
Figura 3.32. Comando Lista Materiales.



Fuente: Propia de los autores

3.2.2.14. Comando: “LISTA PENDIENTES”.

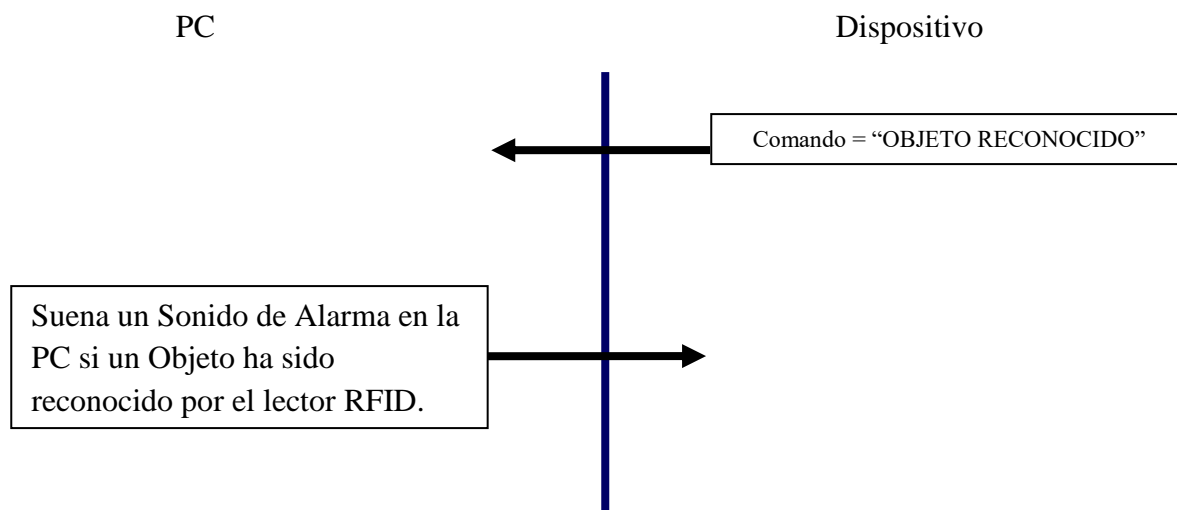
Figura 3.33. Comando Lista Pendientes.



Fuente: Propia de los autores

3.2.2.15. Comando: “OBJETO RECONOCIDO”.

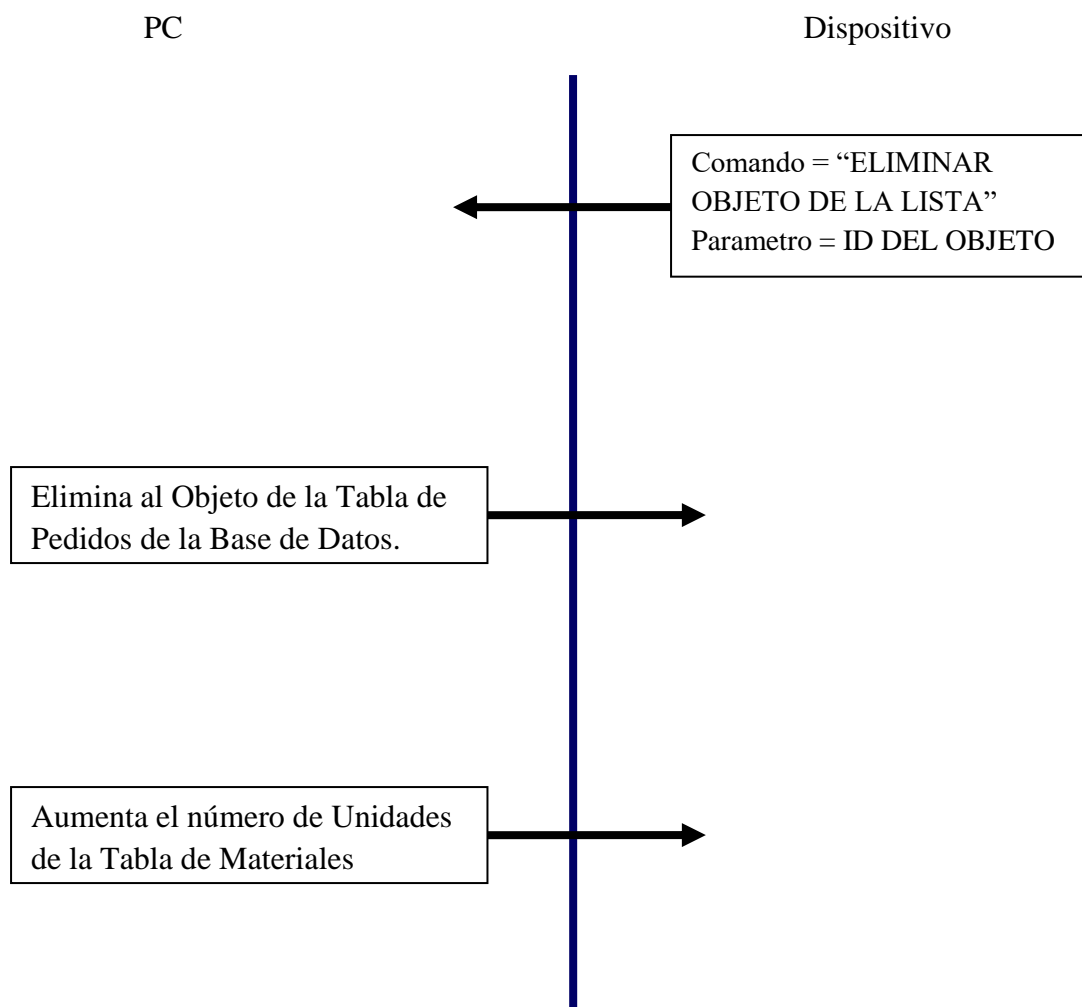
Figura 3.34. Comando Objeto Reconocido.



Fuente: Propia de los autores

3.2.2.16. Comando: “ELIMINAR OBJETO DE LA LISTA”.

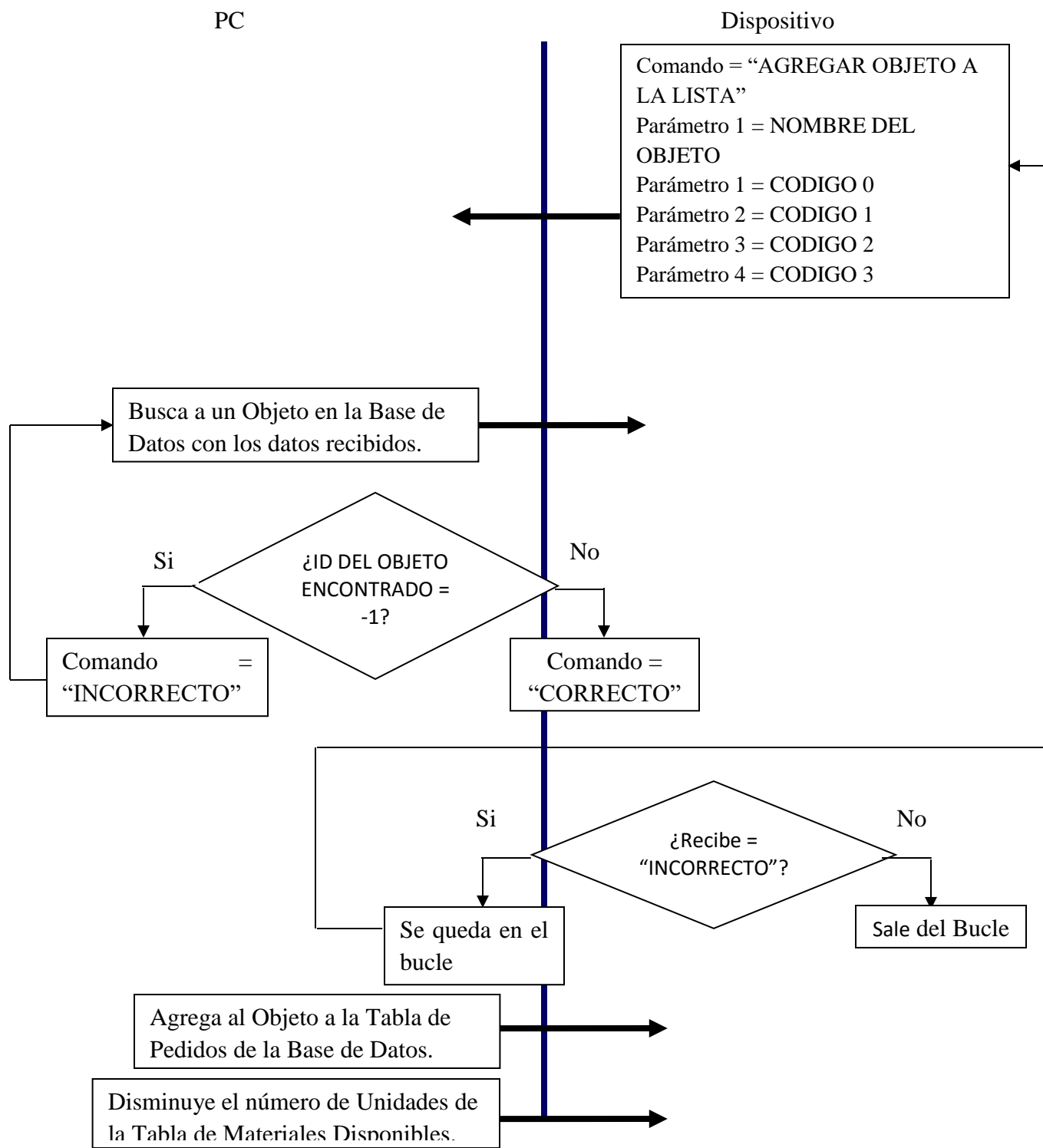
Figura 3.35. Comando Eliminar Objeto de la Lista.



Fuente: Propia de los autores

3.2.2.17. Comando: “AGREGAR OBJETO A LA LISTA”.

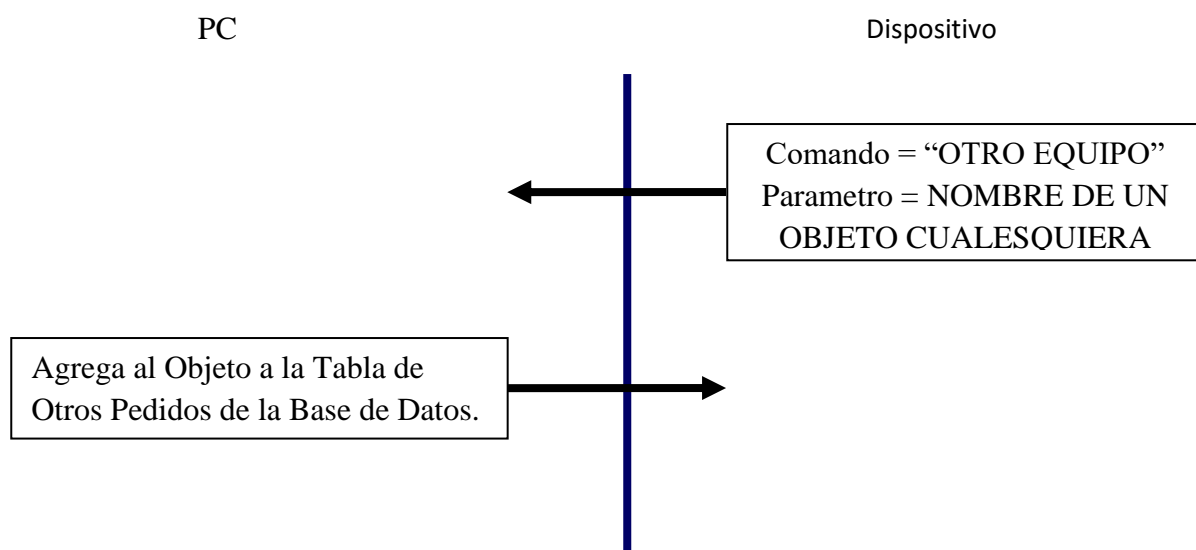
Figura 3.36. Comando Agregar Objeto a la Lista.



Fuente: Propia de los autores

3.2.2.18. Comando: “OTRO EQUIPO”.

Figura 3.37. Comando otro equipo.



Fuente: Propia de los autores

3.3. Costos del sistema

Teniendo en cuenta la descripción de los equipos que se utilizarán para ejecutar este sistema de control de registro de préstamos de equipos a los estudiantes, para el laboratorio de Ingeniería Electrónica, es necesario desglosar ordenadamente los gastos de los recursos que se emplearán para realizar este sistema, es así que se tiene lo siguiente:

3.3.1. Costos de equipamiento.

Tabla 3.13. Costos de equipamiento.

EQUIPOS	CANTIDAD	COSTO UNIDAD (S/.)	COSTO TOTAL (S/.)
Servidor HP Proliant DL 180 GEN9	1	6965.53	6965.53
Lector RFID	1	300	300
Tarjetas RFID	500	1.50	750
Monitor HP 18.5'	1	315.26	315.26
Disco Duro HP	1	723.83	723.83
Microsoft Windows Server 2012 R2	1	2846.42	2846.42
Switch CISCO	1	3201.16	3201.16
			15102.2

Fuente: Propia de los autores

3.3.2. Costos de componentes electrónicos.

Tabla 3.14. Costos de componentes electrónicos.

PRODUCTOS	CANTIDAD	COSTO UNIDAD (S/)	COSTO TOTAL(S/)
Pantalla Táctil LCD/TFT de 7" (SSD1963)	1	245.00	245.00
Placa de Arduino Mega 2560	1	80.00	80.00
Shield LCD/TFT para Arduino Mega 2560	1	41.00	41.00
Placa de Arduino Uno	1	65.00	65.00
Shield ArduinoEthernet W5100	1	52.00	52.00
Sensor de huella digital (GT-511C5)	1	350.00	350.00
Conector de Sensor de Huella Digital	1	3.00	3.00
Fuente de Alimentación de 2.5A– 5v	1	20.00	20.00
Conector RJ45	4	1.00	4.00
Pulsador	1	1.00	1.00
Interruptor unipolar luminoso rojo	1	6.20	6.20
Base enchufe Macho	1	1.00	1.00
Cordon de Alimentación AC	1	3.00	3.00
Transistor BC548	1	1.00	1.00
Placa de fibra de vidrio (7x7cm)	1	3.00	3.00
Soldadura de estaño	5	0.5	2.5
Cable Ethernet UTP	4	1.00	4.00
Cables Jumper	40	0.25	10
			888.70

Fuente: Propia de los autores

3.3.3. Costos del diseño e implementación del sistema.

Tabla 3.15. Costos del diseño e implementación del sistema.

	ELEMENTOS	DETALLE	COSTO(S/)
Diseño y desarrollo de Software	SQL	Software para la Base de Datos	1800
	Visual Basic	Software para la interacción del interfaz gráfico de usuario con el Arduino y sus terminales.	2000
Diseño y Construcción de Hardware	Arduino Mega Arduino Uno	Programación de los controladores de la parte electrónica para que interactúen con el Servidor.	1500
	Diseño de Circuitos	Conexión de los dispositivos del prototipo para que conformen la parte electrónica.	1200
			6500

Fuente: Propia de los autores

3.3.4. Costos del sistema general.

Tabla 3.16. Costos del sistema general.

ELEMENTOS	COSTOS TOTAL (S/)
Costos de equipamiento	15102.2
Costos de componentes electrónicos	888.70
Costo de diseño e implementación del sistema	6500
	S/22490.9

Fuente: Propia de los autores

CAPITULO IV

4. RESULTADOS

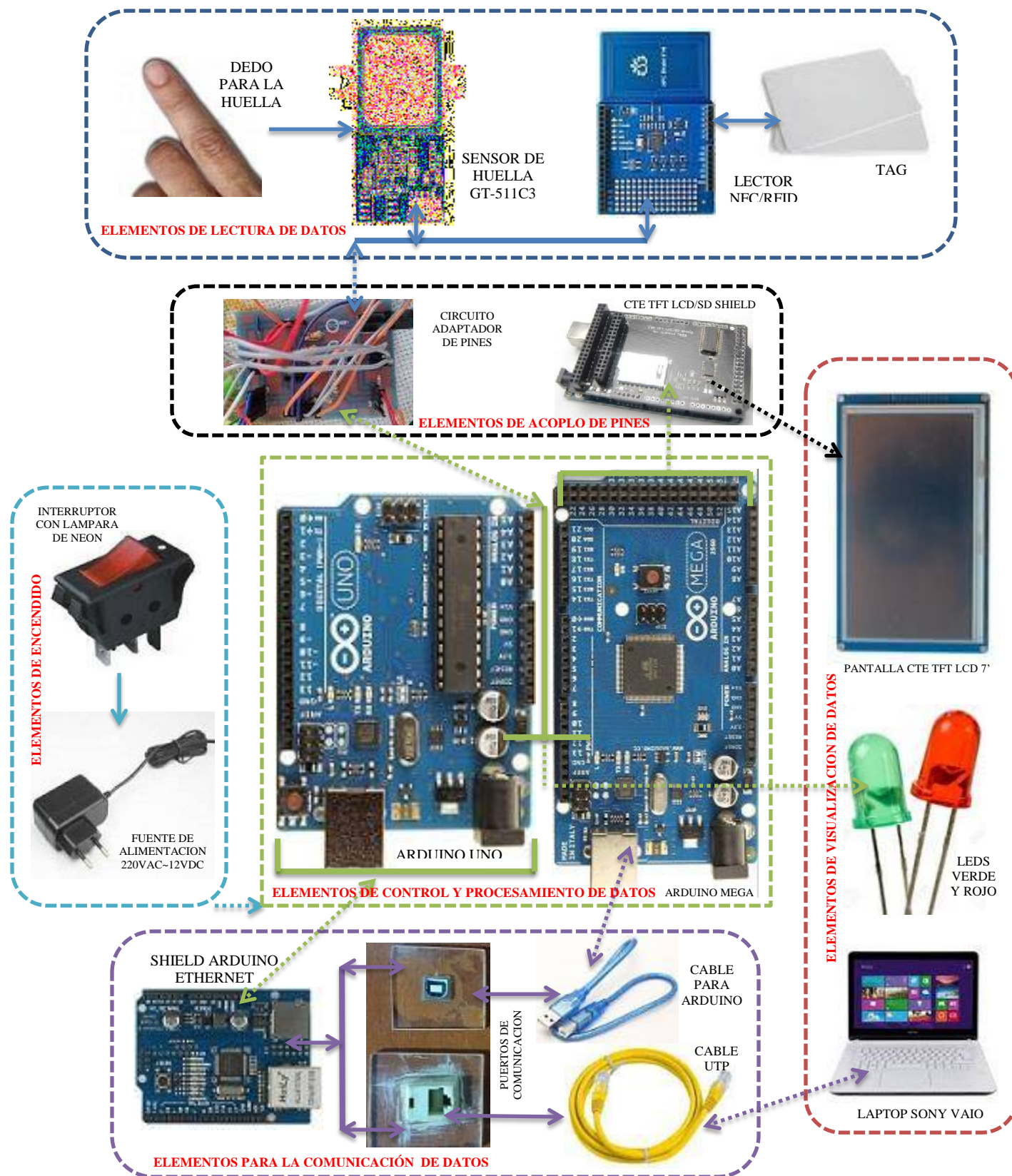
4.1. Diseño del prototipo

Para el presente proyecto se ha diseñado un prototipo que cumple con todas las características presentadas en el diseño del sistema de control para el registro de préstamos de equipos, en la que demuestra que utilizando la plataforma Arduino como hardware se puede realizar de manera satisfactoria este sistema. Este prototipo cuenta con dos partes importantes: un Módulo Electrónico en la que los usuarios por medio de una interfaz interactiva en una pantalla táctil pueden registrarse, pedir equipos y devolverlos, y una PC de control en la que el administrador tendrá interfaz para poder controlar y manipular la información de los usuarios y equipos registrados en la base de datos.

4.1.1. Modulo electrónico

- Elementos de encendido del módulo electrónico.
- Elementos de lectura de datos
- Elementos de control y procesamiento de datos
- Elementos de comunicación de datos
- Elementos acopladores de pines
- Elementos de visualización de datos

Figura 4.1. Esquema general del prototipo



Fuente. Propia de los autores

4.1.1.1. Elementos de encendido del módulo electrónico

En el encendido del módulo electrónico se utilizaron tres elementos destacables, un interruptor para el encendido del módulo en el módulo electrónico, un botón reset para el reinicio del módulo y un adaptador de voltaje de 220VAC~5VDC para la alimentación del Arduino y el circuito adaptador de pines.

➤ ***Interruptor con lámpara de Neón.***

El uso de interruptores que se iluminan cuando se enciende un aparato electrónico, o más bien un sistema es estético y útil a la vez. El interruptor para este proyecto se caracteriza por tener tres pines y su forma de conectarse debe realizarse correctamente para que se ilumine.

La manera de conectar los interruptores luminosos de 3 patas (dos plateadas y una dorada) es la siguiente: Primero se identifica el cable con la fase o que tiene corriente y el cable con neutro con el multímetro, la fase se conecta a la pata plateada de la orilla y el neutro a la pata dorada, en la pata plateada del centro se le conecta un cable para la salida de corriente al activar el interruptor.

Figura 4.2. Interruptor con lámpara de neón



Fuente. Propia de los autores

➤ **Botón reset.**

El botón reset es un pulsador que sirve para restaurar el estado inicial de un dispositivo.

Cuando encendemos el modulo electrónico, se inicia el sistema con una cierta configuración. Si luego de ejecutar diversas tareas deseamos volver al estado inicial, podemos apagar y volver a prender el modulo electrónico, o simplemente, pulsar la tecla de reset para reiniciar el sistema.

En este sentido, el botón de reset suele utilizarse cuando el sistema se traba, bloquea. En caso de que un elemento del sistema presente fallos y haga que el sistema no responda, se puede pulsar reset para que el sistema se reinicie.

Figura 4.3. Boton Reset



Fuente. Propia de los autores

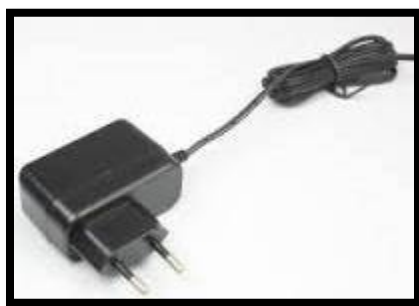
➤ **Fuente de alimentación.**

La fuente de alimentación consiste en un adaptador AC-DC que se utiliza para el Arduino y para el circuito acoplador de pines las cuales estos pueden funcionar desde los 5VDC hasta 12VDC. Este adaptador convierte la corriente alterna de alto voltaje que tomamos del enchufe en la pared a corriente directa de bajo voltaje para el sistema que estamos encendiendo.

Este módulo electrónico contiene en funcionamiento varios elementos las cuales están haciendo un consumo de corriente mayor a lo soportado por los Arduinos, para ello los elementos

se están alimentando de manera independiente a través del circuito acoplador de pines con una fuente de alimentación de 5VDC y con 2.5A.

Figura 4.4. Adaptador AC/DC



Fuente. Propia de los autores

4.1.1.2. Elementos de lectura de datos

La información que se maneja en este prototipo está dada por la lectura de datos que hace el sensor de huellas a través de la huella registrada del dedo de un individuo y el shield RFID/NFC a través de las tag, las cuales el Arduino es el encargado de hacer el procesamiento y control de esta información.

➤ Sensor de huella digital GT-511C3.

El modulo para este prototipo puede almacenar hasta 200 huellas dactilares diferentes (que es 10 veces más que la versión antigua) y es capaz de reconocer la huella a 360°, puesto que para la implementación del sistema se utilizaría la versión más actualizada, el módulo GT-511C5 que puede almacenar hasta 2000 huellas, suficientes para acaparar el registro de todos los alumnos dela escuela de Ingeniería Electrónica. Es pequeño y fácil de montar. Para comenzar, simplemente registrar cada huella digital que desea almacenar enviando el comando

correspondiente y pulsando con el dedo sobre el lector tres veces. La interfaz de conexión con el conector JST-SH tiene cuatro señales: VCC, GND, Tx y Rx.

Principales características.

- Alta precisión en la identificación de huellas y alta velocidad con el Algoritmo SmackFinger 3.0.
- Sensor óptico ultra fino.
- Descarga imágenes de huellas dactilares en el dispositivo.
- Lee y escribe plantillas de huellas digitales y base de datos.
- Protocolo UART simple (por defecto 9600).
- Capaz de verificación 1:1 y 1: N de identificación.

Figura 4.5. Sensor de huella GT-511C3



Fuente. Propia de los autores

Características técnicas del Sensor GT-511C3

Tabla 4.1. Características técnicas del Sensor GT-511C3.

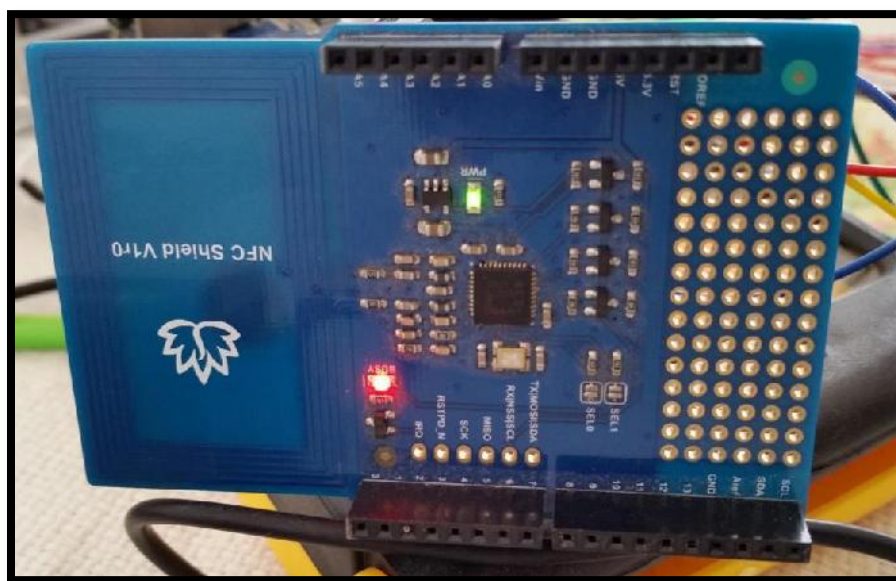
ÍTEM		DESCRIPCIÓN
CPU		ARM Cortex M3 Core (holtek HT32F2755)
Sensor		Sensor óptico
Área efectiva del sensor		14 x 12.5(mm)
Tamaño de imagen		202 x 258 pixeles
Resolución		450 dpi
El máximo número de huellas		200 huellas dactilares
Modalidad de coincidencia		1:1, 1:N
Tamaño de la plantilla		496 Bytes (plantilla) + 2bytes (suma de comprobación)
Interface de comunicación		UART, velocidad en baudios predeterminada = 9600bps después del encendido, USB ver1.1, toda velocidad
Tasa de falsa aceptación		<0.001%
Tasa de falso rechazo		<0.1%
Tiempo de enrolamiento		<3seg (3 huellas dactilares)
Tiempo de identificación		1.0seg (200 huellas dactilares)
Voltaje de operación		DC 3.3 ~ 6V
Corriente de operación		<130Ma
Entorno operativo	Temperatura	-20°C ~ +60°C
	Humedad	20% ~ 80%
Entorno de almacenamiento	Temperatura	-20°C ~ +60°C
	Humedad	20% ~ 80%

Fuente. Datasheet del Sensor de Huellas Dactilares GT-511C3 [30].

➤ *Shield NFC/RFID 13.56 MHz.*

Para nuestro proyecto se utilizó este shield NFC/RFID para la seguridad de los equipos al momento de la lectura de su identificación. Su frecuencia de funcionamiento es de 13.56 MHz (alta frecuencia). Soporta los estándares de comunicación I2C, UART, SPI, para nuestro proyecto utilizamos el que viene por defecto (I2C) donde el Arduino Mega 2560 es el maestro y el shield NFC/RFID es el esclavo.

Figura 4.6. Shield NFC/RFID 13.56 MHz.



Fuente. Propia de los autores

Este shield utiliza un chip PN532, este chip es uno de los más populares y compatibles que se encuentran en el mercado. A continuación describiremos algunas características de este chip.

Chipset PN532

El chipset PN532 es un módulo de transmisión altamente integrado para comunicación sin contacto a 13.56MHz incluyendo la funcionalidad del microcontrolador basado en un núcleo de 80C51 con 40 kbytes de memoria ROM y 1 kbyte de memoria RAM. El PN533 puede operar en 6 diferentes modos:

- Modo Lectura/Escritura (ISO/IEC 14443-A/MIFARE)
- Modo Lectura/Escritura (ISO/IEC14443-B)
- Modo Lectura/Escritura (FeliCa)
- Modo Emulación de tarjeta MIFARE clásico de 1k o tarjeta MIFARE clásico de 4k (ISO / IEC 14443 /MIFARE)
- Modo Emulación de tarjeta FeliCa
- Modo Peer-to-Peer (ISO/IEC 18092/ECMA 340).

El chipset PN532 contiene modulación y demodulación completamente integrado para diferentes métodos de comunicación por proximidad y protocolos a 13.56MHz con un firmware para diferentes modos de compatibilidad y las interfaces que requiere el controlador. El firmware incorporado y el hardware interno soportan el manejo del protocolo del controlador para diferentes interfaces como I2C, SPI y UART.

Principales características

- Circuitería analógica altamente integrada para respuestas de demodulación y decodificación.

- Controladores de búfer de salida para conectar una antena con un número mínimo de componentes externo.
- Integrado con un detector de nivel de radiofrecuencia.
- Integrado con un lector de datos.
- La distancia típica de operación en modo lectura/escritura es aproximadamente 50mm dependiendo de la antena.
- Soporta las interfaces I2C, SPI y High Speed Serial UART.
- Interface de radio frecuencia integrada para el estándar NFCIP-1 a 424kBaud.
- Cuenta con la posibilidad para comunicarse con una interfaz de radio frecuencia por encima de 424kBaud utilizando un circuito analógico externo.

Características Técnicas del Shield NFC/RFID

Tabla 4.2. Características Técnicas del Shield NFC/RFID.

ÍTEM	DESCRIPCIÓN
Dimensiones	3.2”(81.6mm)x2.1”(53.4mm)x0.063”(1.6mm)
Voltaje de Operación	DC 3.3 ~ 5 V
Corriente de Operación	150 mA
Frecuencia de Operación	13.56 MHz
Estándar Compatible	ISO 14443 tipo A y tipo B interfaz
Interfaces de Comunicación	I2C, SPI, UART
Distancia de Lectura/ Escritura	Lectura probada de Testeo 1”(25mm)

Fuente. Propia de los autores

➤ ***Etiqueta RFID pasiva (tag).***

En proyecto se utilizaron etiquetas pasivas por su bajo coste y puesto que las etiquetas pasivas no incluyen una batería. Este tipo de etiqueta obtiene energía del lector, que emite ondas electromagnéticas que inducen una corriente en la antena de la etiqueta.

Figura 4.7. Etiqueta RFID pasiva



Fuente. Propia de los autores

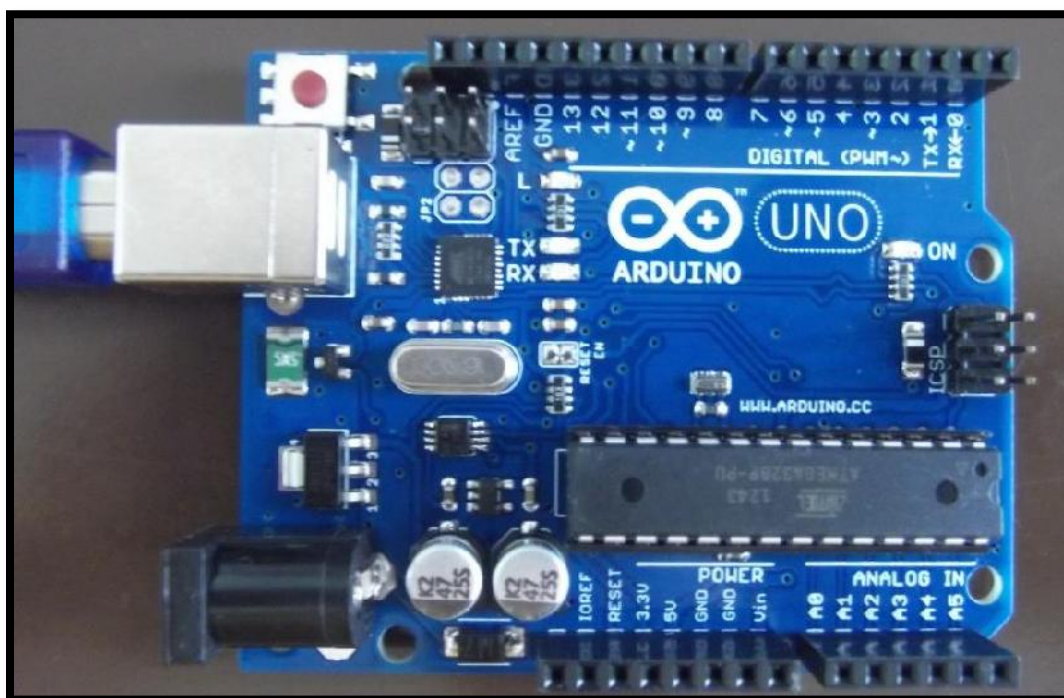
4.1.1.3. Elementos de control y procesamiento de datos.

El control y el procesamiento de los datos están a cargo del Arduino Mega que es el que interactúa con la mayoría de los elementos permitiéndole su control y el Arduino Uno que procesa la información para comunicarlo con la PC de control.

➤ *Arduino Uno.*

El Arduino Uno es una placa microcontroladora basada en el ATmega328. Tiene 14 pines de entrada/salida (E/S) digital (de los cuales 6 pueden ser usados como salidas PWM), 6 entradas analógicas, un oscilador de cuarzo a 16MHz, una conexión USB, un conector para alimentación, una cabecera ICSP, y un botón de reset.

Figura 4.8. Arduino Uno R3

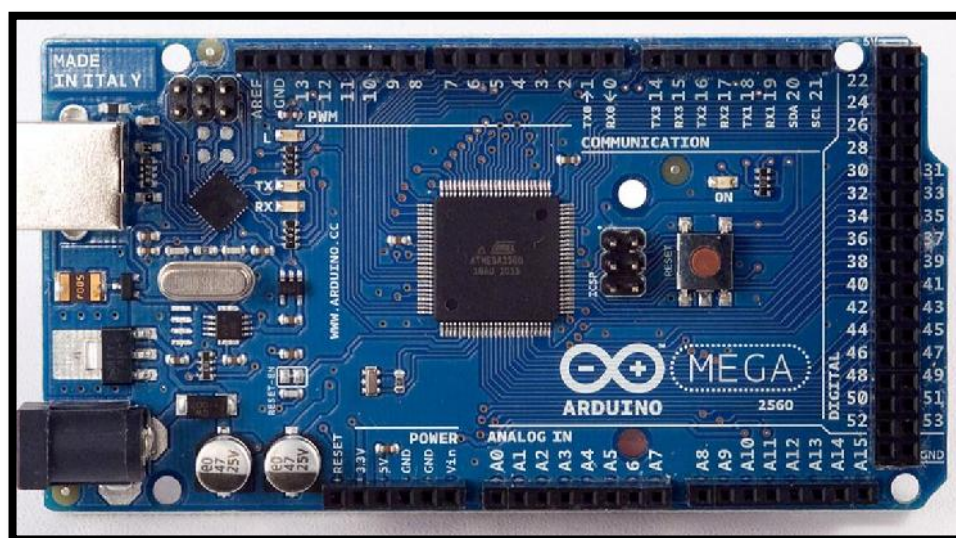


Fuente. Propia de los autores

➤ *Arduino Mega 2560*

Está basado en el microcontrolador ATmega2560. Tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serial por hardware), un oscilador de cuarzo 16 MHz, conexión USB, jack de alimentación, conector ICSP y botón de reset.

Figura 4.9. Arduino Mega R3



Fuente. Propia de los autores

4.1.1.4. Elementos de comunicación de datos

Estos son los elementos que intervienen en la comunicación de sistema es decir la comunicación entre el módulo electrónico y la PC de control a través del Shield Arduino Ethernet con un cable UTP. Además interviene dos puertos de comunicación, uno para la comunicación Ethernet y la otra para comunicación serial a través de un cable USB para la programación del Arduino Mega.

➤ *Arduino Ethernet Shield*

El Shield Arduino Ethernet tiene la capacidad de conectarse a una red cableada TCP/IP. Aporta la misma funcionalidad que la placa Arduino Ethernet pero en forma de shield complementario acoplado a la placa Arduino Uno.

Las entradas y salidas tienen exactamente la misma disposición y funcionalidad que las del Arduino Uno. Incluso si fuera necesario, se puede conectar sin problemas un segundo shield en la parte superior del shield Ethernet para seguir sumando funcionalidad.

El shield requiere 5 V para funcionar. Este voltaje lo aporta la placa Uno mediante el encaje del pin de alimentación correspondiente entre placa y shield. La comunicación entre el chip W5100 y el Arduino Uno se establece mediante los pines 10, 11, 12 y 13 (vía SPI) por lo que estos pines no se pueden utilizar para otro propósito. Esto implica que realmente al usar este shield tenemos 4 entradas/salidas digitales menos.

Figura 4.10. Shield Arduino Ethernet



Fuente. Propia de los autores

Principales características

- El conector RJ45 es compatible con PoE y tiene de reserva algunos pines para el montaje del módulo PoE opcional.
- Es compatible con el Arduino Mega, a través de la comunicación SPI (cabecera ICSP).
- Reajuste de alimentación en el IC para eliminar problema de reinicio.

- Inversor Schmitt – trigger en el IC para anular la selección del W5100.
- Todos los indicadores LEDs de estado de W5100 están montados en la placa.
- El Arduino no se sobrecalentará cuando se conecte una alimentación externa de 12 V.

Características técnicas del Shield Arduino Ethernet.

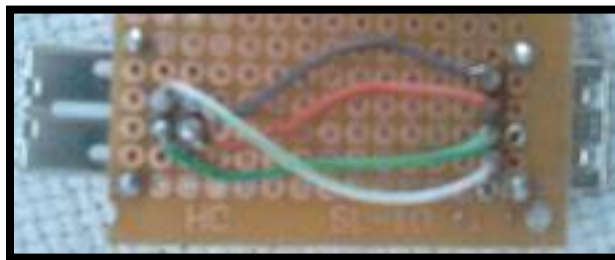
Tabla 4.3. Características Técnicas del Shield Arduino Ethernet.

ÍTEM	DESCRIPCIÓN
Dimensiones	52.34mm x 68.58mm
Voltaje de Operación	DC 4 ~ 12V
Estándar de Comunicación	IEEE802.3af
Interfaces de Comunicación	UART, I2C, SPI (ISCP Arduino Mega)

Fuente. Propia de los autores

➤ Puertos de comunicación.

Este sistema cuenta con dos puertos de comunicación, el primero para la comunicación entre el módulo electrónico y la PC de control y el segundo puerto para la comunicación entre el Arduino Mega y una PC cualquiera que contenga el IDE del Arduino para su respectiva programación, o para cambio del código fuente. Para ello se utilizó un conector unión RJ45 que unirá la conexión interna de la comunicación Ethernet y la conexión externa de dicha comunicación. También se utilizó un conector USB tipo A hembra para la conexión serial interna y para la conexión externa un conector USB tipo B hembra, ambos conectores van conectados entre sí.

Figura 4.11. Conector*Fuente. Propia de los autores**Figura 4.12. Conexión de los conectores**Fuente. Propia de los autores*

➤ ***Cable de par trenzado no blindado (UTP).***

UTP de las siglas en inglés Unshielded twisted pair el cuál no tiene ningún tipo de aislante que permita una mayor protección contra interferencias electromagnéticas. Es el tipo de cable que más se utiliza para aplicaciones cotidianas de red. Es por ello que para este sistema se usó como medio de transmisión dos cables de red UTP categoría 5 que tiene un ancho de banda de 100MHz, una velocidad de 100Mbps y es esencial para la comunicación Ethernet. El primero para la comunicación interna entre el Shield Arduino Ethernet y el conector unión RJ45 y el segundo cable UTP para la comunicación externa desde el conector unión RJ45 hasta la PC de control.

Figura 4.13. Cable



Fuente. Propia de los autores

➤ ***Cable USB para Arduino.***

Este es un cable estándar USB 2.0 usa el conector más común de tipo A macho a tipo B macho. Compatible con la mayoría de las placas Arduino, para este proyecto compatible con el Arduino Uno y Arduino Mega. Se utilizaron dos cables USB como medio de transmisión de datos, el primero para la comunicación interna del Arduino Mega 2560 y el conector USB tipo A hembra y el segundo para la comunicación externa entre el conector USB tipo B hembra hasta la PC donde permitirá reprogramar el Arduino Mega. Se utilizó también un cable USB como medio alimentación para energizar el Arduino Uno a través del adaptador AC/DC.

Figura 4.14. Cable USB para



Fuente. Propia de los autores

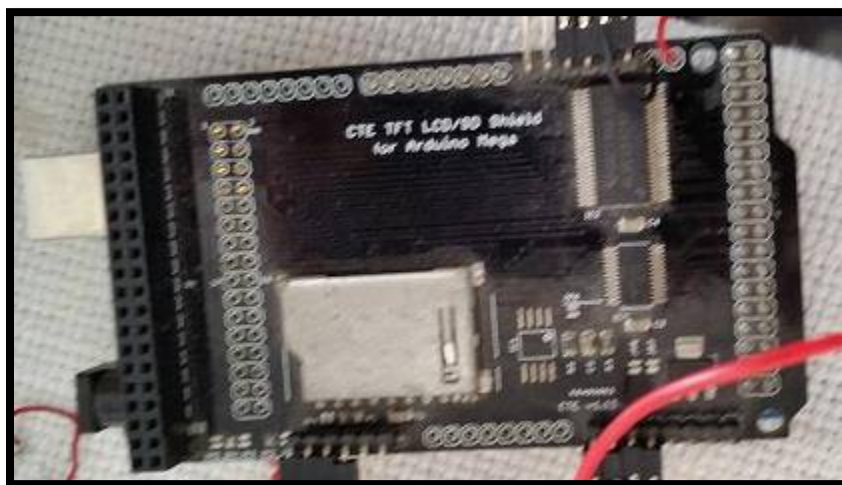
4.1.1.5. Elementos Acopladores de pines.

En este proyecto se utilizaron dos elementos acopladores de pines; el primero el Shield CTE LCD TFT para el acoplar los pines del Arduino Mega con la Pantalla TFT LCD y el segundo para acoplar los pines del Arduino (Mega y Uno) con los demás dispositivos (sensor de huella, lector RFID/NFC, leds, reset).

➤ *CTE LCD TFT/SD shield.*

Para poder controlar la pantalla TFT LCD con el Arduino Mega se empleó este Shield CTE LCD TFT/SD que es un shield adaptador de pines que sirve como acople para la pantalla TFT LCD y este se pueda conectar directamente sin cables con el Arduino Mega. Tiene una ranura SD que sirve para poder guardar imágenes que se mostraran en la pantalla, o simplemente otro archivo que requiera la aplicación. Esta versión de shield LCD tiene 40 pines que son compatibles con el Arduino Mega.

Figura 4.15. Shield LCD TFT/SD para la Pantalla TFT LCD.

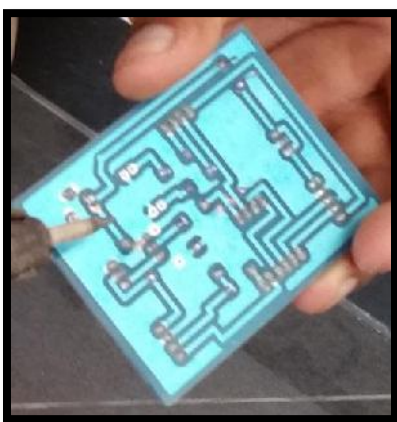


Fuente. Propia de los autores.

➤ ***Circuito adaptador de pines.***

Este es un acoplamiento de pines usados por todos los dispositivos del Módulo Electrónico, en un solo circuito, para esto se tiene conocimiento de todos los pines que se van a utilizar en el Arduino Mega y en el Arduino Uno con sus respectivos pines en los que irán conectados los demás componentes del sistema.

Figura 4.16. Circuito adaptador de pines



Fuente. Propia de los autores.

4.1.1.6. Elementos de visualización de datos

Los elementos que se usaron en este proyecto para visualizar los datos obtenidos de los sensores (sensor de huella, lector RFID/NFC) son: una pantalla táctil de 7 pulgadas, dos indicadores leds y una laptop de la marca sony.

➤ ***Modulo Pantalla TFT LCD***

Esta pantalla TFT LCD tiene una resolución de 800 por 480 y su tamaño es de 7 pulgadas, su funcionamiento está basado en el controlador SSD1289 para gestionar la pantalla TFT LCD, al ser TFT estos transistores permiten mejorar la calidad de imagen en comparación a las pantallas

tradicionales LCD, también consumen menos energía, lo cual la hace ideal para el sistema implementado, gracias a su panel táctil el usuario podrá manipular los menús desarrollados en la interfaz gráfica del sistema.

Lo interesante de esta pantalla TFT LCD es que se cuenta con una librería de apoyo realizado por Henning Karlsen (desarrollador de librerías de software libre para pantalla TFT LCD de cualquier tamaño para Arduino), para la administración de su hardware a través de Arduino, tanto para la parte táctil y para el LCD de la pantalla, aunque se tuvo que modificar esta librería para que funcione con el modelo de controlador de la pantalla empleada y para que funcionase con la resolución de la pantalla de 7 pulgadas .

Figura 4.17. Pantalla TFT LCD.



Fuente. Propia de los autores.

➤ ***Indicadores visuales leds.***

Un diodo Led (light-emitting diode o diodo emisor de luz) además de permitir el paso de la corriente en un solo sentido, en el sentido que la corriente pasa por el diodo, este emite luz.

Para visualizar cuando el usuario ha entrado al sistema con éxito (logueo) se activa un led de color verde, y para indicar la devolución de un equipo de manera correcta se activa un led de color rojo.

Figura 4.18. Diodos leds



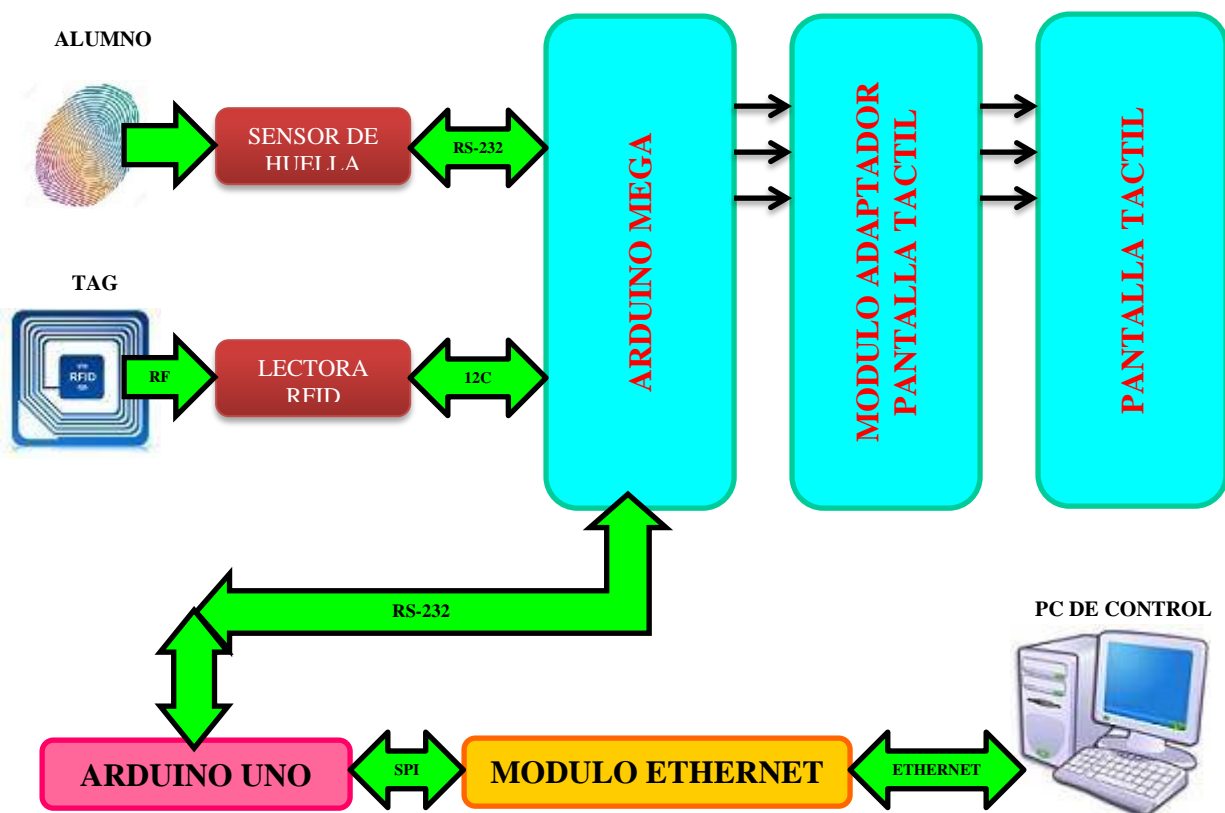
Fuente. Propia de los autores

A continuación se describe el diagrama de bloques del funcionamiento del prototipo, para luego describir el la conexión de los circuitos que conforman este prototipo:

4.1.1.7. Diagrama de bloques del hardware del prototipo

El hardware del prototipo está basado en la plataforma de Arduino, el uso de dos Shields; uno que se encargara de adaptar los pines del Arduino Mega para la pantalla táctil y el otro que se encargara del envío y recepción de los datos a través de la comunicación Ethernet en el Arduino Uno. Además se basa en el uso de dos sensores; el sensor de huella digital para el registro de alumnos y el lector NFC/RFID para el registro de equipos.

Figura 4.19. Diagrama de bloques del hardware del prototipo.



Fuente: Propia de los autores.

Agrupando todos los conceptos presentados anteriormente con las características que debe cumplir el hardware se describirá cada uno de los bloques del prototipo de la Figura anterior.

En primer lugar al pasar cada usuario por el sensor de huella dactilar, este registra su ID y son enviados mediante comunicación serial al Arduino Mega, esta comunicación lo hace por el puertos 2 (Rx2 y Tx2); este procesa los datos para que interactúe con el shield adaptador de la pantalla y posteriormente tenga una salida hacia la pantalla táctil con una interfaz gráfica y luego ser enviada a la PC de Control.

Además el Arduino Mega siempre está atento a la espera de recibir información de su puerto serial (Rx1, Tx1). El puerto serial (Rx1, Tx1) es para la comunicación con el Arduino Uno que es el encargado de procesar la información para enviarlo por comunicación Ethernet a la PC de Control mediante el Shield Arduino Ethernet; mientras que por los pines SDA y SCL (pines 20 y 21 respectivamente) del Arduino Mega, son usados para la comunicación I2C con el NFC/RFID (pines A4 SDA y A5 SCL). Esta comunicación está atenta a la lectura de los Tag de los equipos del laboratorio. Una vez que el Arduino Mega recibe la información del Lector RFID, lo procesa para una salida grafica en la pantalla táctil, este se comunica con el Arduino Uno de manera serial y este mismo se comunica con la PC de Control a través del Shield Arduino Ethernet.

4.1.1.8. Circuito de Conexión del Arduino Mega con el Sensor de Huella GT511C3.

Para la comunicación serial entre el Arduino Mega y sensor de huella GT-511C3 se utilizaron los pines seriales del puerto serial (2), el pin 16 como Tx2 y 17 como Rx2; el Sensor GT-511C3 en sus interfaces de transmisión y recepción generan voltajes lógicos establecidos en valores de 3.3 VDC y 0 VDC por lo cual se debe adecuar la señal de transmisión proveniente de la placa Arduino Mega 2560 (5 VDC) para que pueda ser receptada dentro de los rangos de operación del sensor GT-511C3, para esto se construye un divisor de voltaje con lo cual se puede establecer la comunicación serial entre los dispositivos de manera adecuada con las siguientes características:

Según la conexión de pines del Sensor de Huella Dactilar GT-511C3

Tabla 4.4. Conexión de los pines del sensor GT-511C3.

# pin	Sensor de huella Dactilar GT-511C3	Arduino Mega 2560
1	UART-TX (3.3V TTL)	RX2 (PIN 17)
2	UART-RX (3.3V TTL)	TX2 (PIN 16)
3	GND	GND
4	VIN (3.3V ~ 6V)	3.3V ~5V

Fuente: Propia de los autores

Utilizando la fórmula:

$$V_{out} = V_{in} * \frac{R_2}{R_1 + R_2} \dots \dots \dots (1)$$

Despejando R_1 en la ecuación (1):

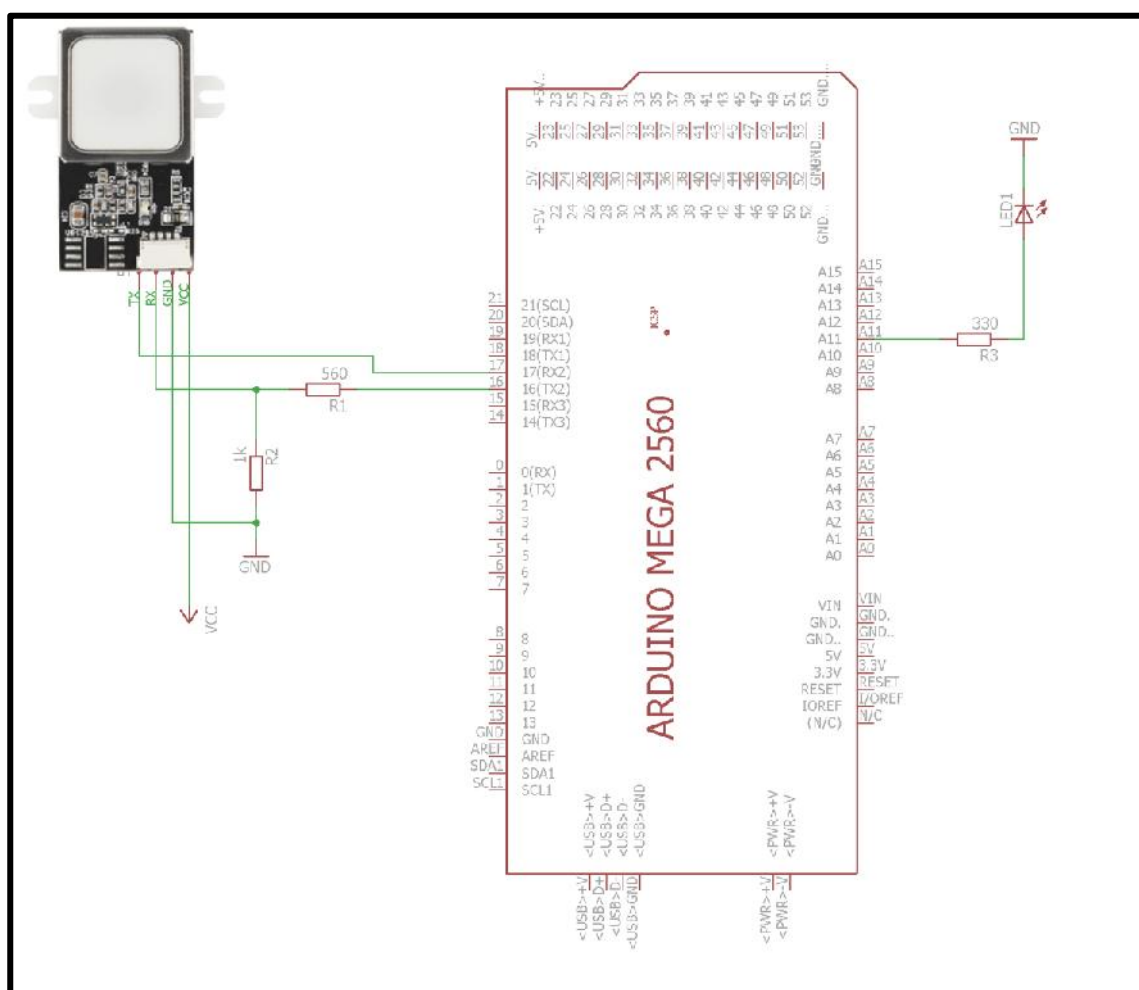
$$R_1 = R_2 \left(\frac{V_{in}}{V_{out}} - 1 \right) \dots \dots \dots (2)$$

Para $R_2 = 1000\Omega$ $V_{in} = 5v$ $V_{out} = 3.3v$

$$R_1 = 1000 \left(\frac{5v}{3.3v} - 1 \right)$$

$$R_1 = 515.15\Omega \cong 560\Omega$$

Figura 4.20. Circuito de conexión del Arduino Mega con el sensor GT-511C3.



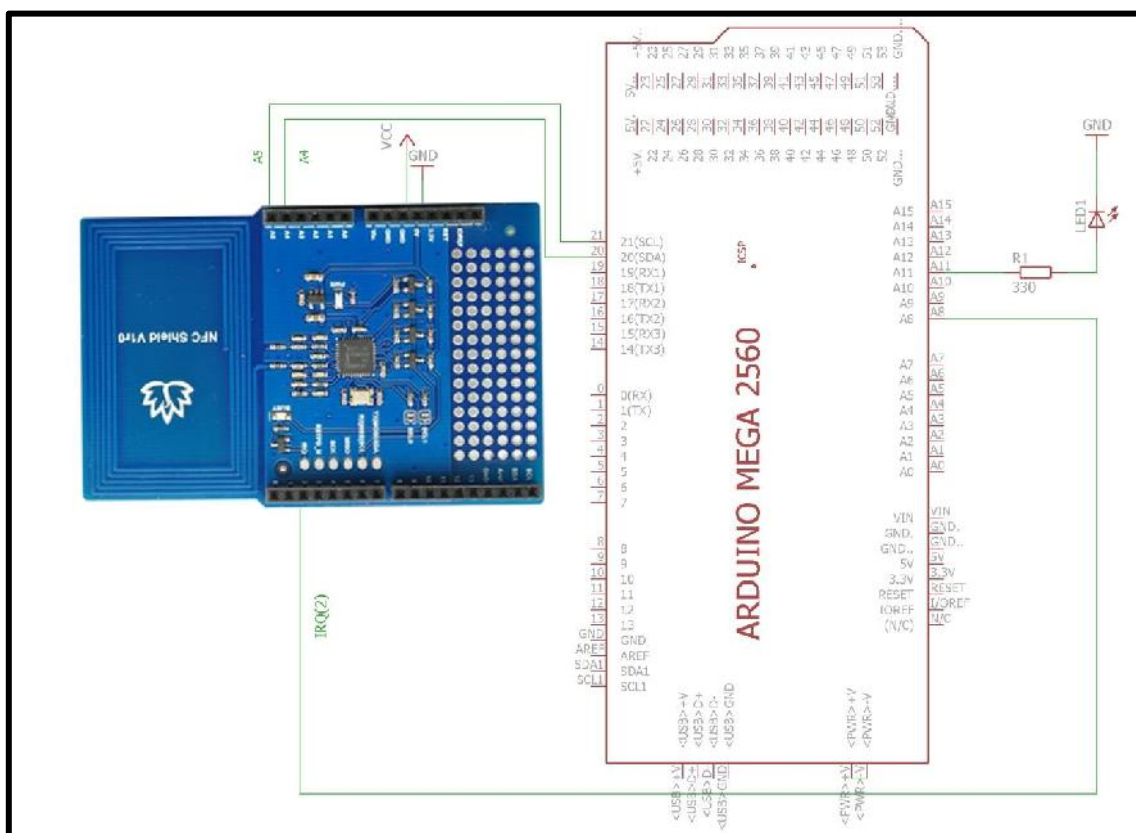
Fuente: Propia de los autores.

4.1.1.9. Circuito de Conexión del Arduino Mega con el Shield NFC/RFID.

Considerando las características técnicas mencionadas en el capítulo anterior acerca del lector NFC/RFID; este se comunica con la placa del Arduino Mega mediante comunicación I2C, donde el maestro es el Arduino Mega y el esclavo es el lector NFC/RFID.

En la figura 3.13 podemos ver el conexionado entre el Arduino Mega y el NFC/RFID; se tienen conectados los pines SDA y SCL (pines 20 y 21 respectivamente) del Arduino Mega, estos son para la comunicación I2C con el NFC/RFID (pines A4 SDA y A5 SCL). Se tiene también el pin IRQ (pin 2 en el NFC/RFID) que es el pin de aviso de transmisión; es decir el Arduino Mega le envía un dato, este pin se activa y el NFC/RFID se prepara para recibir esos datos.

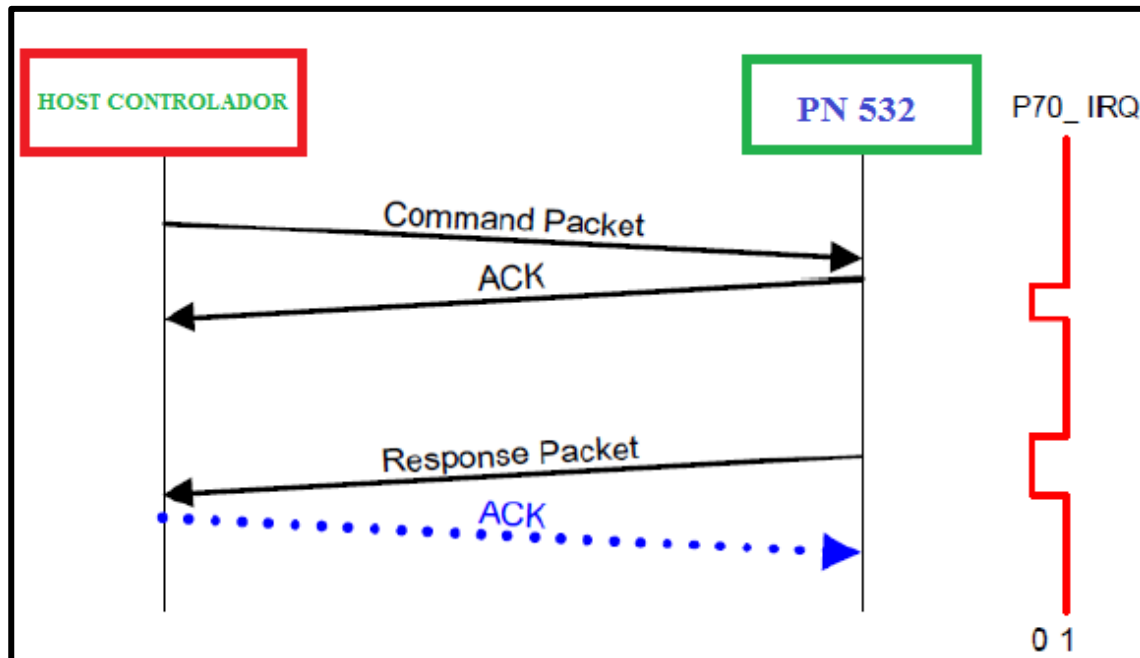
Figura 4.21. Circuito de conexión del Arduino Mega con el Lector NFC/RFID.



Fuente: Propia de los autores.

➤ *Secuencia de mensajes en la comunicación*

Figura 4.22. Intercambio de datos entre el Host Controlador y el PN532.



Fuente. Propia de los autores

La estructura de comunicación es la siguiente:

- El controlador host siempre se configura como maestro en todo el proceso de intercambio de datos.
- El host manda un comando al PN532.
- El PN532 responde con una señal de conformidad para informar al controlador que el comando ha sido recibido satisfactoriamente.
- El PN532 ejecuta el comando.
- El PN532 manda la respuesta correspondiente al host.
- De manera opcional, el controlador puede enviar una señal de conformidad al PN532.

4.1.1.10. Circuito de Conexión del Arduino Mega, Arduino Uno y el Shield Ethernet.

En la figura 3.15 se muestra las conexiones entre el Arduino Mega con el Arduino Uno, este último es el puente de comunicación entre lo que sería la parte electrónica (Arduino, módulos, pantalla táctil, sensores, RFID, etc.) y el software de aplicación (Visual Basic, base de datos).

Para una mejor comprensión de este circuito vamos a resaltar lo antes mencionado en el diagrama de bloques del sistema en general de la figura 3.11.

Se recoge toda la información obtenida de los sensores (sensor de huella, lector RFID, tag), se procesa en el Arduino Mega, se visualiza y se hace un control de estos en la pantalla táctil y se le envía por comunicación serial al Arduino Uno para que este junto con el Shield Arduino Ethernet envíen toda la información a la PC de Control y base de datos por medio de comunicación Ethernet, luego regresa información por la misma comunicación Ethernet al Arduino Uno y este se encarga de enviarlo al Arduino Mega de manera serial y posteriormente visualizarse en la pantalla.

Cabe mencionar que la comunicación entre el Shield Arduino Ethernet y el Arduino Uno se lleva a cabo utilizando un protocolo que se llama SPI (Serial Peripheral Interface).

SPI es un protocolo serial síncrono que se utiliza para comunicar un microcontrolador con otro y con periféricos a distancias cortas. Para hacer una conexión SPI siempre habrá un dispositivo maestro (usualmente un microcontrolador) que controlará uno o varios periféricos (esclavos), se utilizan por lo general 3 líneas de conexión y una de selección que son:

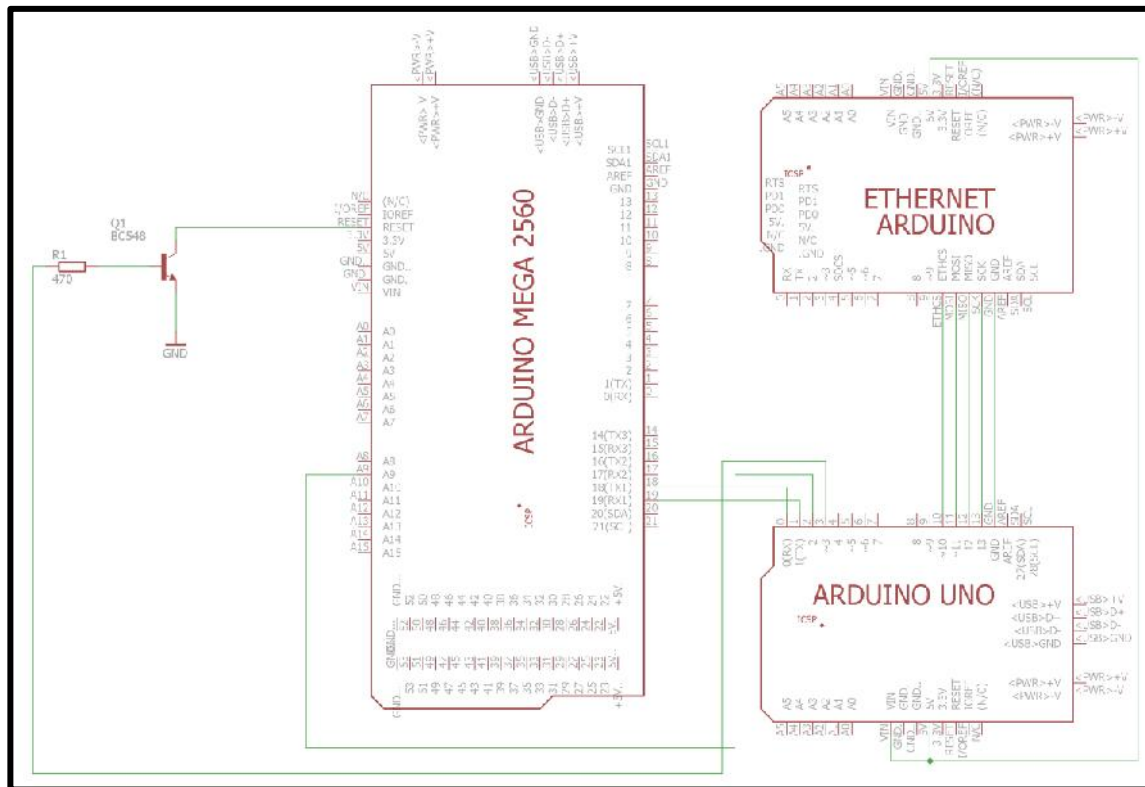
- **SO o MISO (Master In Slave Out).** La línea que utiliza el esclavo para enviar datos al maestro.
- **SI o MOSI (Master Out Slave In).** Datos del maestro al esclavo.
- **SCK (Serial clock).** Pulsos de reloj para sincronizar la comunicación.
- **CS o Select.** Se usa por el master para habilitar o deshabilitar un determinado periférico.

El Arduino Uno tiene 4 líneas para usar el protocolo SPI:

- Pin digital 10 para CS,
- Pin digital 11 para SI,
- Pin digital 12 para SO y
- Pin digital 13 para SCK.

Se utilizó un transistor NPN BC548 para las interrupciones del Arduino Mega a través del Arduino Uno, cabe resaltar que el proceso principal del Arduino Mega es controlar la pantalla, pero a su vez tiene que controlar otros dispositivos, es aquí donde se encuentra la dificultad y hace que el Arduino Mega se sobrecargue, para ello se puso el transistor en modo Colector Abierto para que invierta la señal que envía el Reset del Arduino Uno y no haya conflicto de resets con el reset por software al momento de cargar un programa.

Figura 4.23. Circuito de conexión del Arduino Mega, Arduino Uno y Shield Arduino



Fuente: Propia de los autores

4.1.1. PC de Control

Para almacenar la información en la base de datos y visualizar la información tanto de alumnos como de los equipos registrados en la base de datos se empleó una laptop marca Sony. En esta laptop se ejecutará la aplicación del software y se pondrá en marcha la visualización del registro de los datos obtenidos de los sensores para almacenarlos en la base de datos de esta laptop.

Tabla 4.5. Características técnicas de la PC de Control

ÍTEM	DESCRIPCIÓN
Marca	Sony
Serie	54577384-0015321
Modelo	SVF15215CLW
Procesador	Intel(R) Core(TM) i-3337U CPU @ 1.80Ghz
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Edición de Windows	Windows 10 Home Single Language

Fuente. Propia de los autores

Figura 4.25. PC de control



Fuente. Propia de los autores

4.2. Implementación del prototipo

Para la construcción del prototipo se realizaron una serie de procedimientos, primero se verificó el estado de funcionamiento del sensor de huella digital, el lector NFC/RFID y se hizo la calibración de la pantalla táctil, luego se procedió a implementar el circuito acoplador de pines, para poder hacer la carga del código fuente a los Arduinos y una vez probado el funcionamiento de este prototipo se procedió al ensamblado del Módulo Electrónico donde se le añadieron todos sus elementos que la conforman.

Figura 4.26. Diagrama de bloques del proceso de construcción del prototipo.



Fuente. Propia de los autores

4.2.1. Verificación del estado de elementos principales

4.2.1.1. Verificación del estado del sensor de huella GT-511C3.

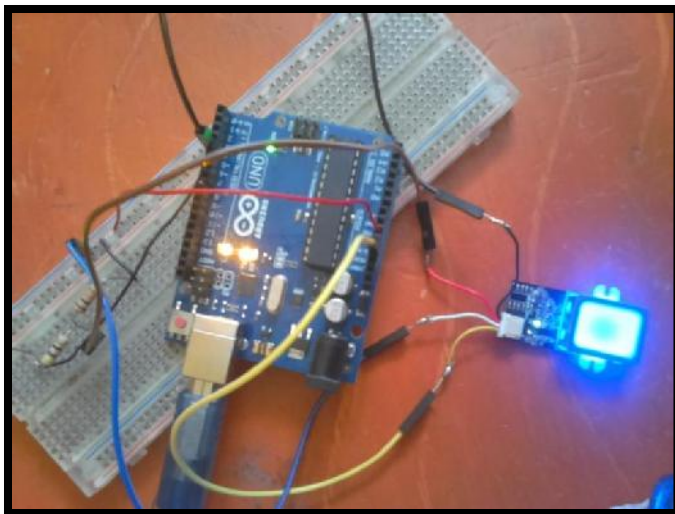
Se descargó la librería FPS_GT511C3.h con los ejemplos de prueba de la página SparkFun. Y se procedió a cargarlos en el Arduino, se hizo el conexionado del Arduino con el Sensor de huella y se hizo lo siguiente.

Se cargó el ejemplo FPS_Blink que es parpadear el led que tiene incorporado el sensor de huella.

Se cargó el ejemplo de prueba FPS_Enroll que consiste en inscribir la huella en la base de datos del sensor presionando cualquiera de los dedos en el sensor por tres veces.

Y por último se cargó el ejemplo de prueba FPS_IDFinger que consiste en identificar la huella, es decir se presiona el dedo inscrito por una sola vez en el sensor, este lo busca en la base de su memoria eeprom y lo reconoce.

Figura 4.27. Verificación del estado del sensor de huella.



Fuente. Propia de los autores

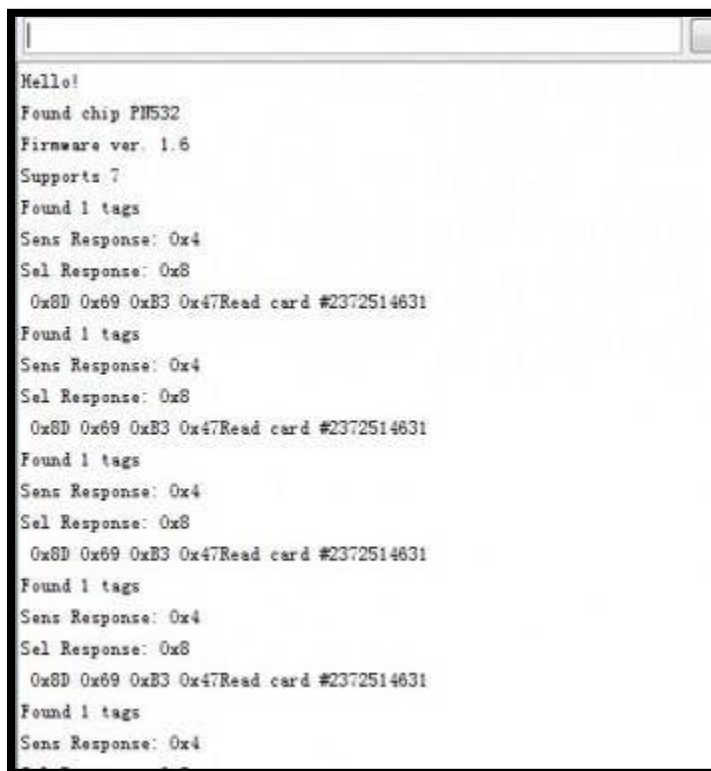
4.2.1.2. Verificación del estado del lector NFC/RFID.

Igual que en el caso del sensor de huella, se descargaron los ejemplos de código de la página botciencia; se cargó el ejemplo en el microcontrolador del Arduino y se hicieron las debidas conexiones del lector con el Arduino:

Una vez realizado lo anterior se pasó la tarjeta RFID sobre el lector NFC/RFID y se obtuvo las siguientes respuestas:

Lectura de la memoria completa sobre una tarjeta MIFARE con el ejemplo readAllMemoryBlocks.

Figura 4.28. Lectura de memoria completa sobre una tarjeta MIFARE.



```

Mello!
Found chip PN532
Firmware ver. 1.6
Supports 7
Found 1 tags
Sens Response: 0x4
Sel Response: 0x8
0x8D 0x69 0xB3 0x47Read card #2372514631
Found 1 tags
Sens Response: 0x4
Sel Response: 0x8
0x8D 0x69 0xB3 0x47Read card #2372514631
Found 1 tags
Sens Response: 0x4
Sel Response: 0x8
0x8D 0x69 0xB3 0x47Read card #2372514631
Found 1 tags
Sens Response: 0x4
Sel Response: 0x8
0x8D 0x69 0xB3 0x47Read card #2372514631
Found 1 tags
Sens Response: 0x4

```

Fuente. Propia de los autores

Escritura de datos en un bloque de memoria MIFARE con el ejemplo
writeMifareMemory.

Figura 4.29. Escritura de datos en un bloque de memoria MIFARE.



```

Hello!
Found chip PN532
Firmware ver. 1.6
Supports 7
Found 1 tags
Sens Response: 0x4
Sel Response: 0x8
0x8D 0x69 0xB3 0x47Read card #2372514631

Write Successful
Read block 0x08:
0 1 2 3 4 5 6 7 8 9 A B C D E F
  
```

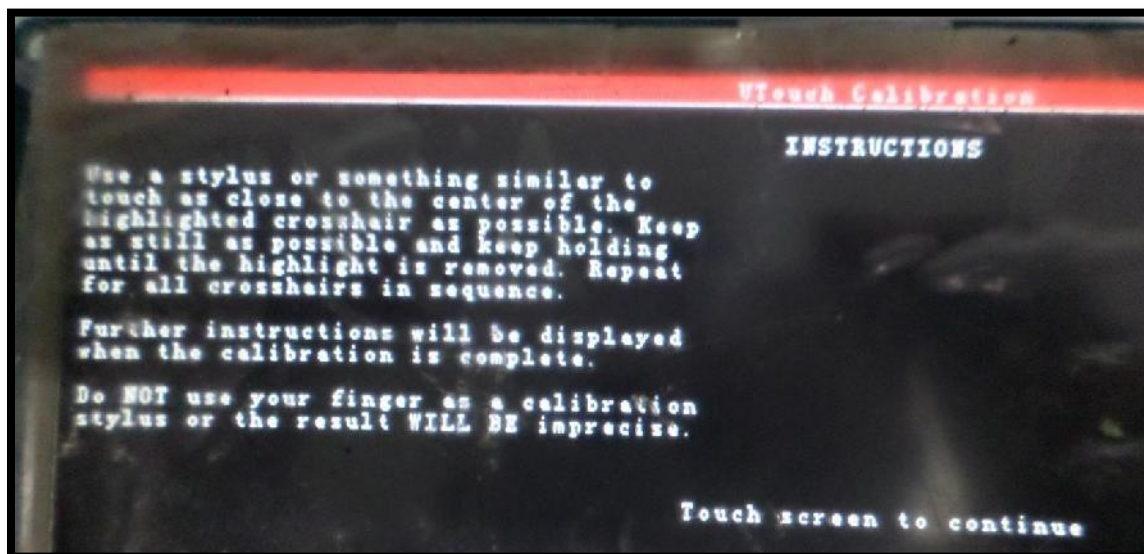
Fuente.

Propia de los autores

4.2.1.3. Calibración de la pantalla táctil.

Para el calibrado de la pantalla táctil debemos utilizar un lápiz táctil o algo similar con punta fina para poder calibrar la pantalla, nos pide que presionemos la pantalla para poder continuar.

Figura 4.30. Menú de instrucciones para calibrar la pantalla táctil.



Fuente. Propia de los autores

Luego presionar los asteriscos según te vaya indicando el cuadradito con un más adentro.

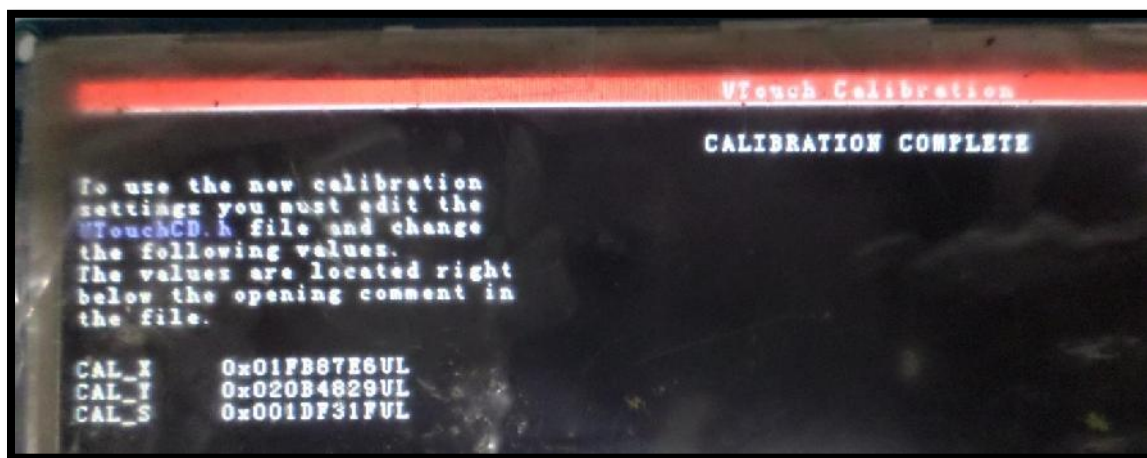
Figura 4.31. Cuadros para presionar en pantalla táctil para su calibración.



Fuente. Propia de los autores

Una vez terminado todo, arroja los valores de la calibración resultante, ahí dice que vayas a la librería “UTouchCD.h” y que actualices los valores.

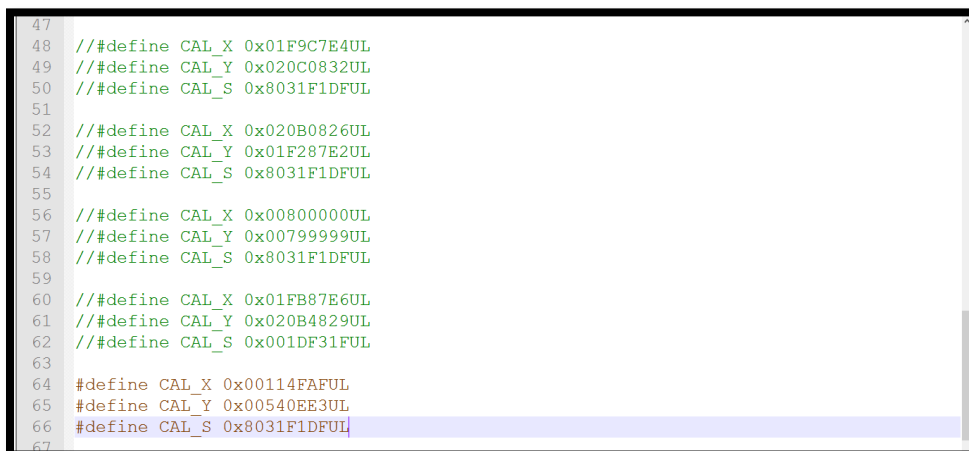
Figura 4.32. Calibración terminada y valores resultantes después de la calibración.



Fuente. Propia de los autores

Los valores obtenidos de la calibración se copian en el archivo “UTouchCD.h”, así como se ve en la figura, en este caso se ha hecho varias veces la calibración hasta encontrar los valores más óptimos posibles. Los valores que no se usan se deben de ir comentando.

Figura 4.33. Valores de calibración copiados en el archivo UTouchCD.h.

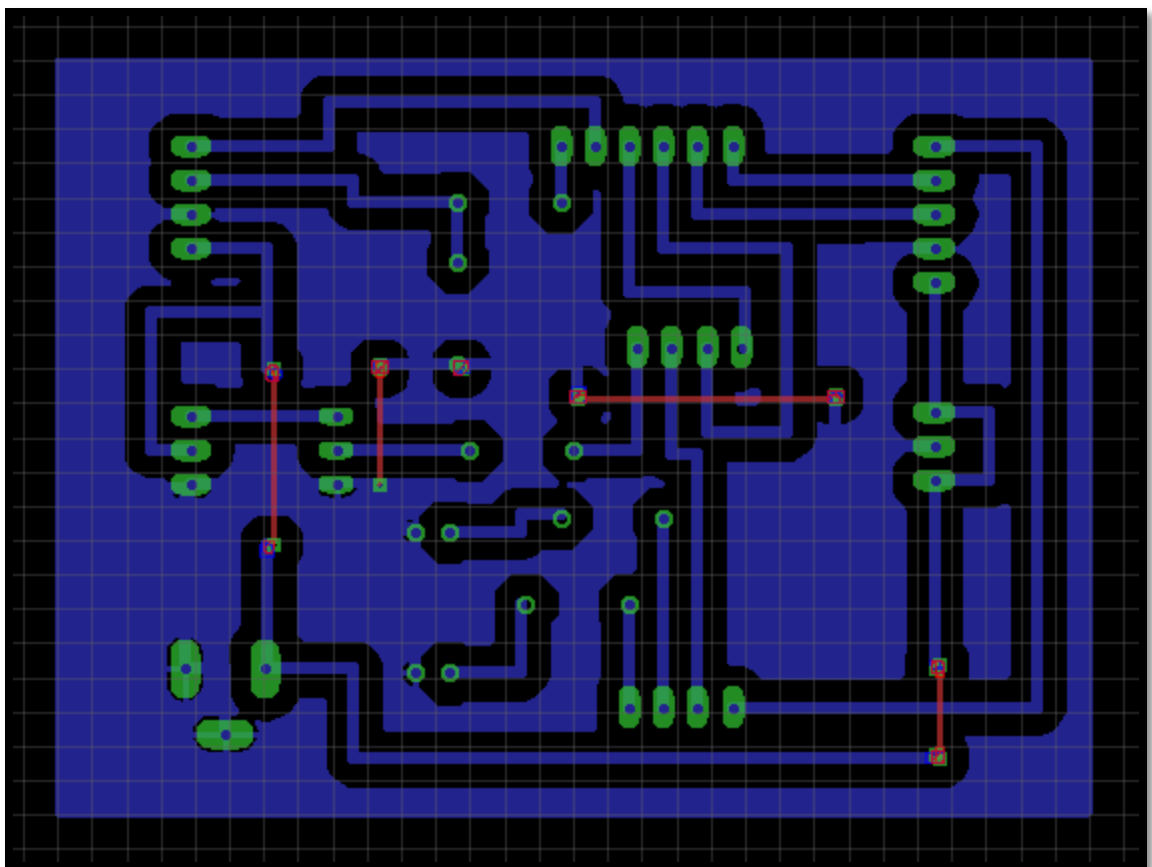


Fuente. Propia de los autores

4.2.2. Implementación del circuito adaptador de pines

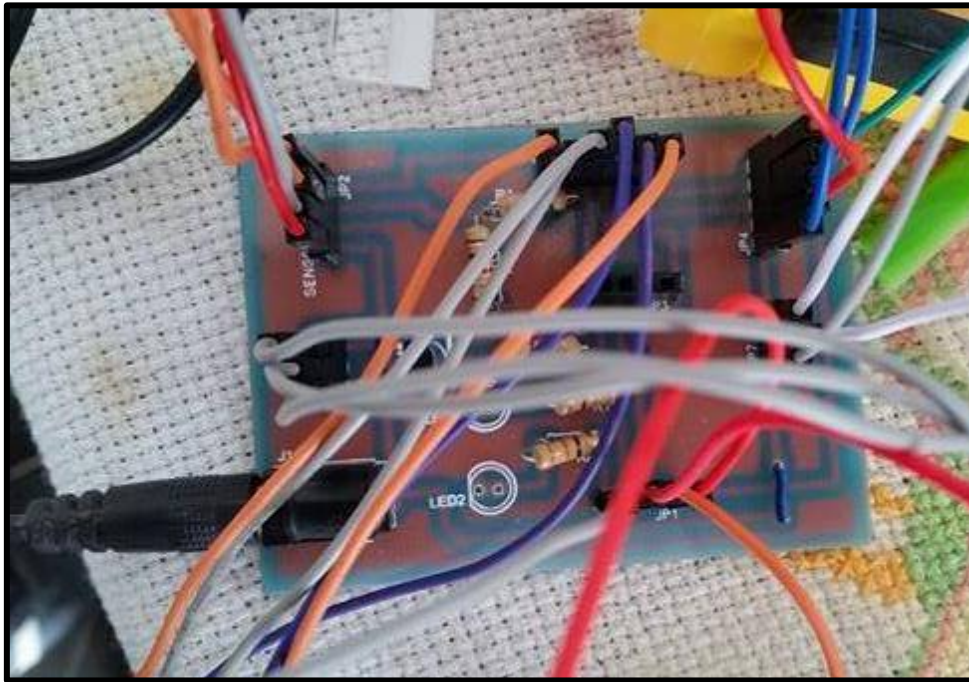
Como se había mencionado anteriormente este circuito es un acoplador de pines en las que un solo circuito conectamos los controladores del sistema, sus sensores y actuadores para dar una mejor organización en el conexionado de estos mismos. Para ello se dispuso de los esquemas de conexión entre el Arduino Mega y el sensor de huella, se dispuso también del esquema de conexión entre Arduino Mega y el lector NFC/RFID, y la conexión entre Arduino Mega, Arduino Uno y el shield Arduino Ethernet, también las conexiones con la pantalla LCD/TFT.

Figura 4.34. PCB del circuito acoplador de pines.



Fuente. Propia de los autores

Figura 4.35. Circuito de acoplamiento de pines implementado.



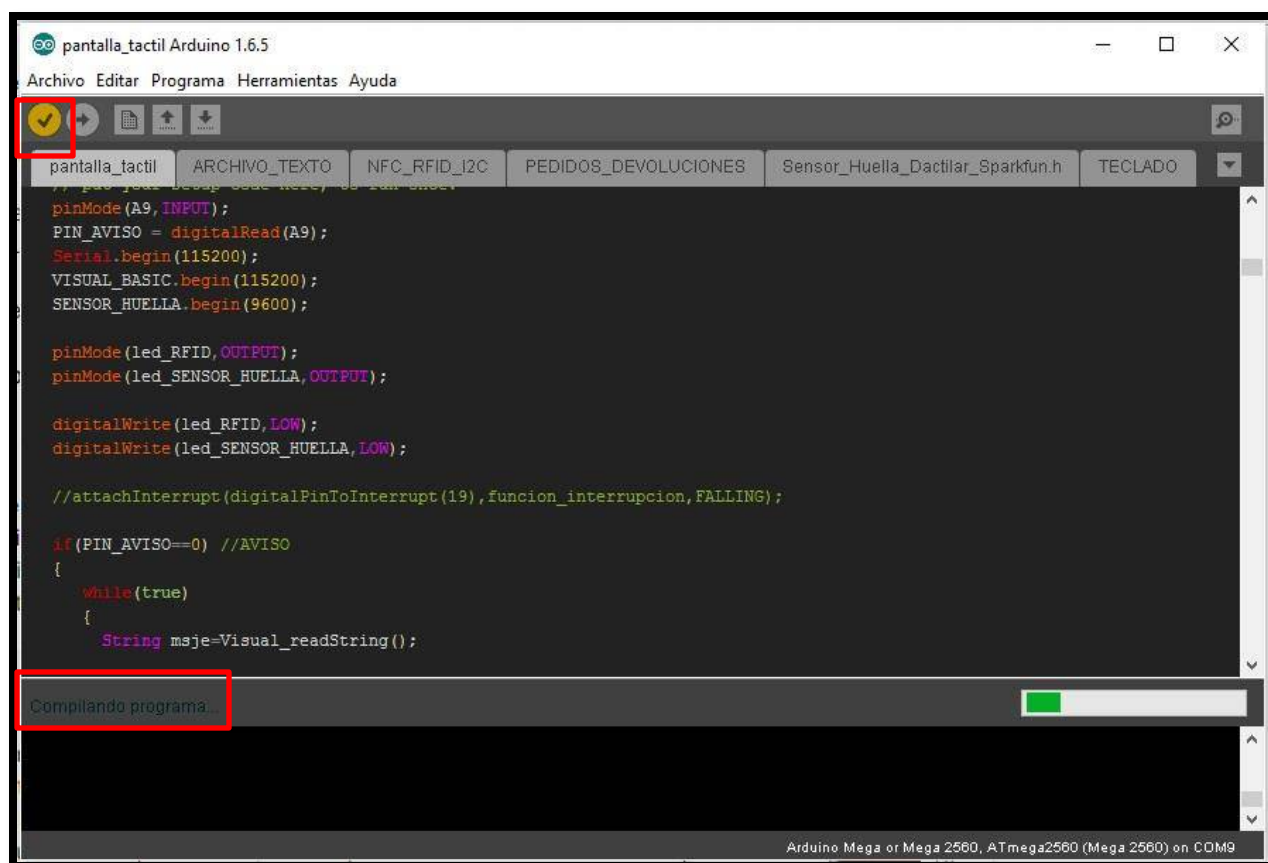
Fuente: Propia de los autores

4.2.3. Compilación y Carga del código fuente del Arduino Mega y Arduino Uno.

4.2.3.1. Compilación del código fuente del Arduino Mega.

Una vez que se tuvo en claro lo que queríamos que hiciera el Arduino se realizó el código fuente para luego proceder a compilarlo lo que significa revisar todo el código que hemos escrito y buscar posibles fallos de sintaxis además de incoherencias. Es decir se verifica que el código fuente sea correcto.

Figura 4.36. Compilación del código fuente del Arduino Mega.

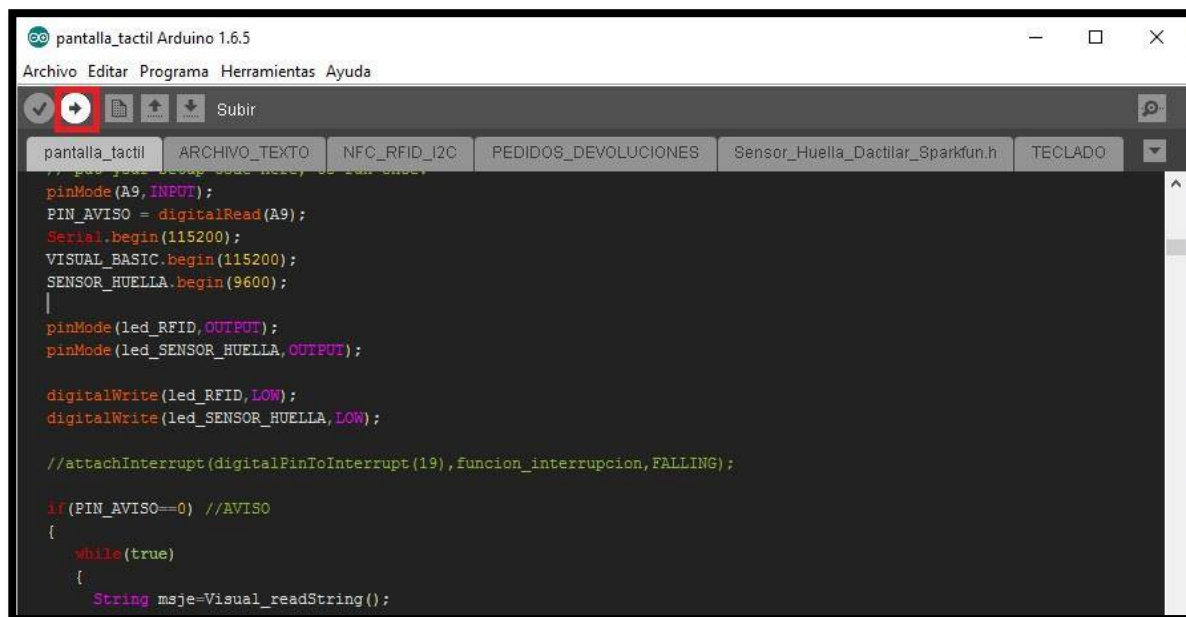


Fuente: Propia de los autores

4.2.3.2. Cargar el código fuente al Arduino Mega

Una vez compilado y asegurado que no existen errores en nuestro código fuente se procede a la carga del código al microcontrolador de Arduino pulsando sobre la flecha que se aprecia en la figura.

Figura 4.37. Subida del código fuente al microcontrolador del Arduino Mega.



Fuente. Propia de los autores

El código fuente del Arduino Mega se puede apreciar en los Anexos de este proyecto.

Para la compilación y carga del código fuente del Arduino Uno se sigue los mismos pasos que en el Arduino Mega e igual manera el código fuente del Arduino Uno se puede apreciar en los Anexos de este proyecto.

4.2.4. Ensamblaje del Módulo Electrónico

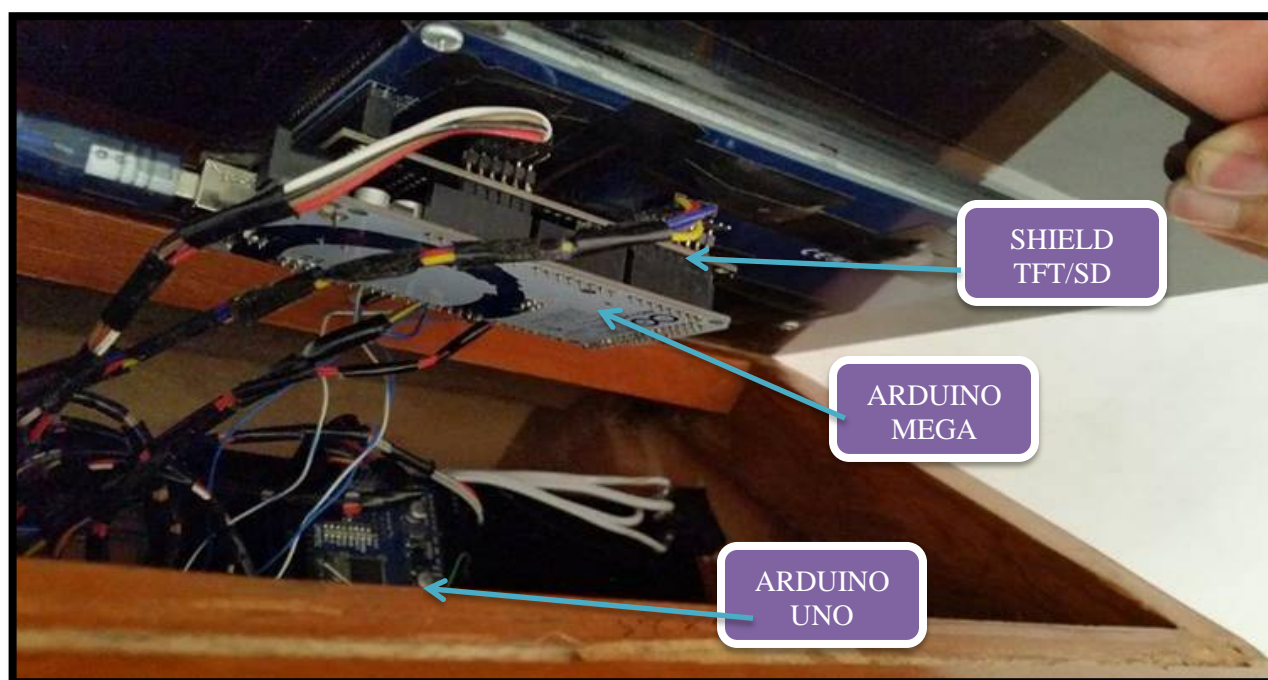
Este Módulo Electrónico fue implementado en un mueble de madera en la cuales este contiene los elementos que interactúan con el usuario y también con la PC de control, se utilizó una pieza de acrílico para empotrar la pantalla, el sensor de huella y los leds en la parte superior de la madera tal como se muestra en la siguiente figura.

Figura 4.1. Ensamblaje de la parte superior del módulo.



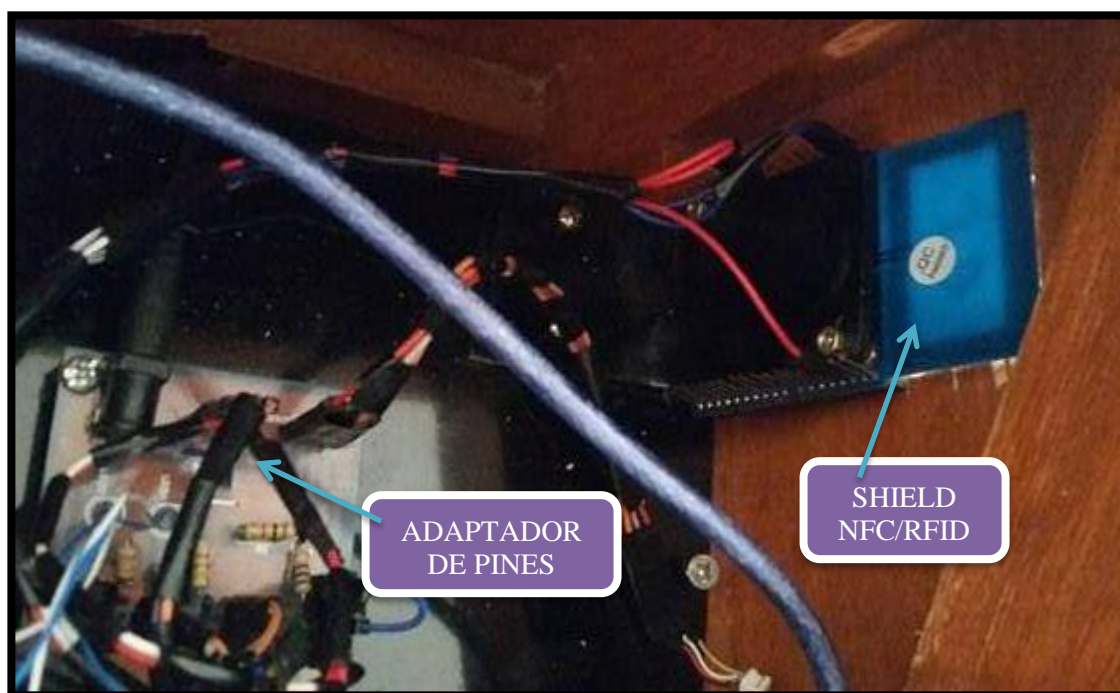
Fuente: Propia de los autores

Figura 4.38. Arduino Mega y Arduino Uno montados en el módulo electrónico.



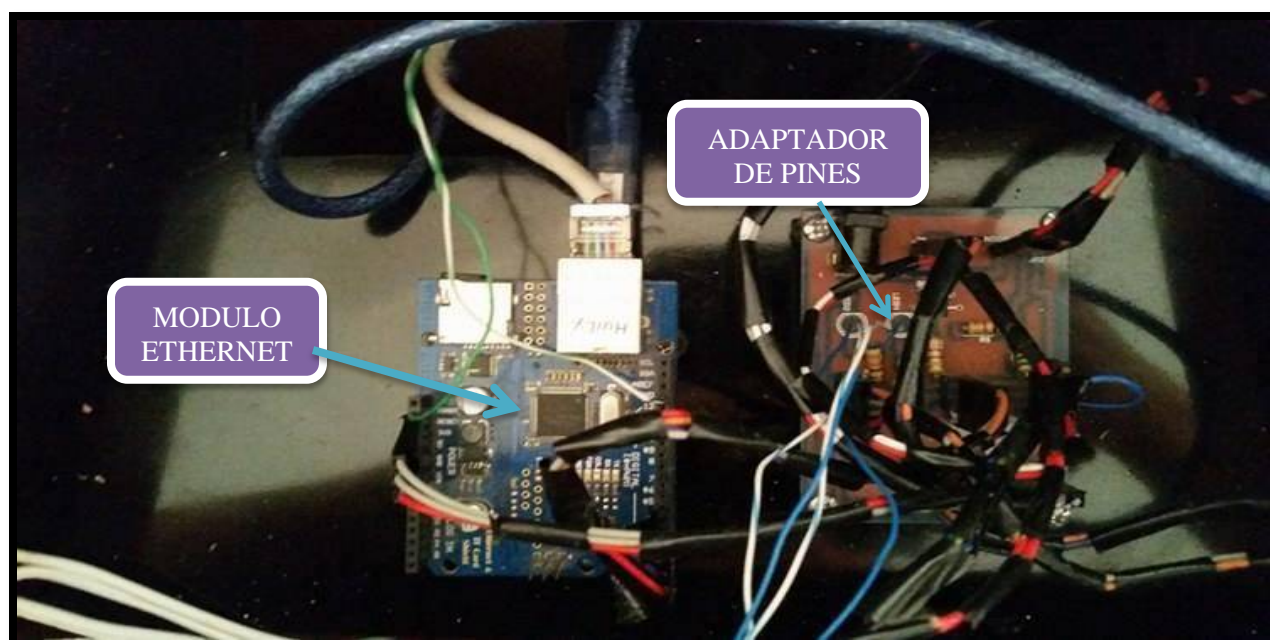
Fuente. Propia de los autores

Figura 4.39. Módulo shield NFC/RFID y adaptador de pines montado en el módulo



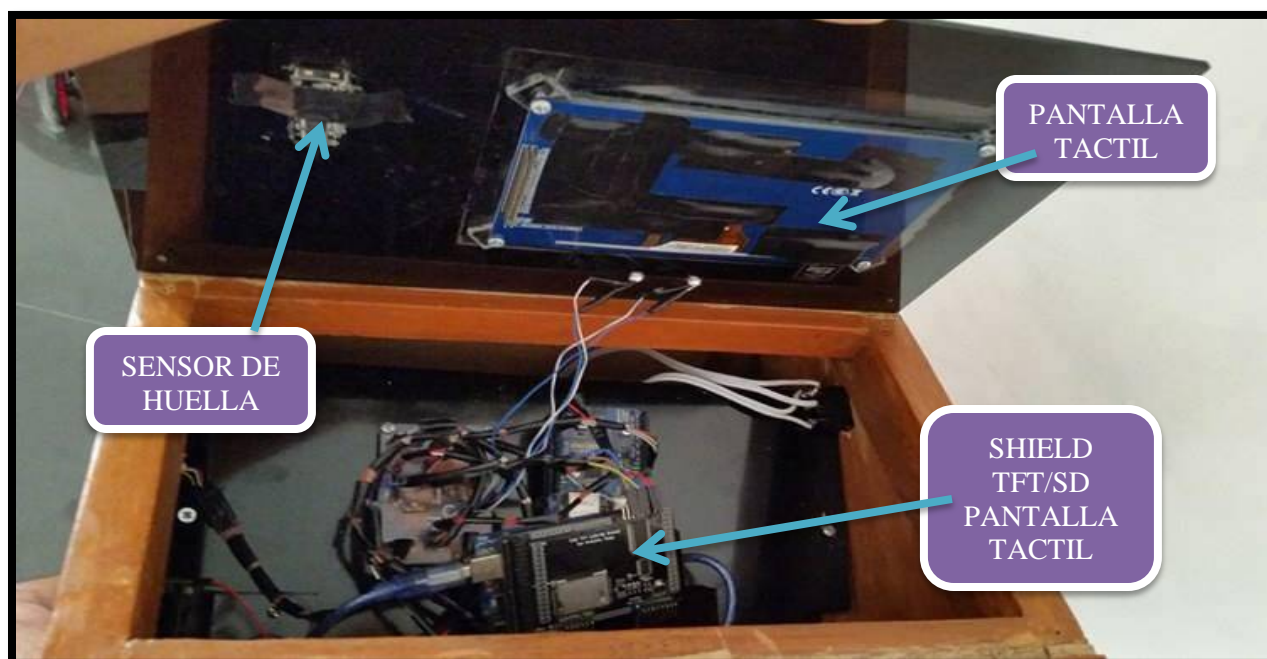
Fuente. Propia de los autores

Figura 4.40. Módulo Ethernet y adaptador de pines montados en el módulo electrónico.



Fuente. Propia de los autores

Figura 4.41. Sensor de huella, pantalla táctil y shield para la pantalla montados en el módulo



Fuente. Propia de los autores

Figura 4.42. Módulo Electrónico terminado.



Fuente: Propia de los autores

4.3. Funcionamiento del prototipo

4.3.1. Interfaz Gráfico de la Pantalla LCD/TFT (Modulo Electrónico)

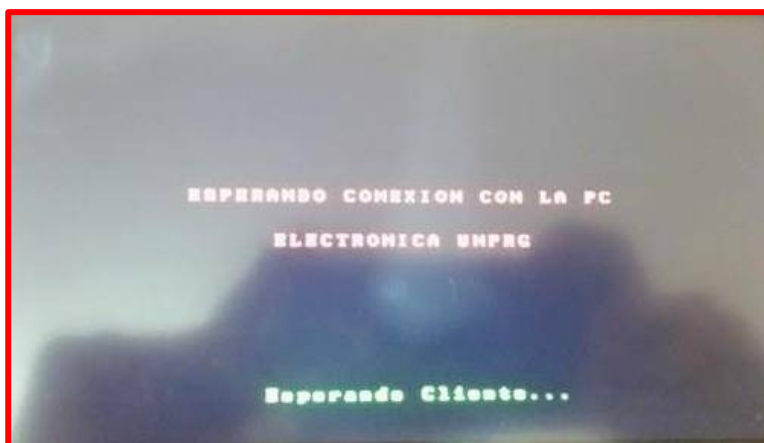
Este prototipo tiene un interfaz que permite al usuario comunicarse con la PC de control, cuenta con una pantalla táctil en la que permite realizar registro de alumnos, pedido, devoluciones y transferencia de equipos, pero siempre siendo supervisado por un administrador.

A continuación describiremos la ventana que muestra el interfaz en el momento que se conecta con la PC de control.

4.3.1.1. Verificación de la conexión entre la PC de control y el Modulo Electrónico.

En la figura 4.43 podemos apreciar una línea de texto que dice “EPERANDO CLIENTE” esto quiere decir que el Módulo Electrónico está esperando conectarse con la PC de control; este trabajo lo realiza el Arduino Uno por medio del Shield Arduino Ethernet donde este mismo es el servidor.

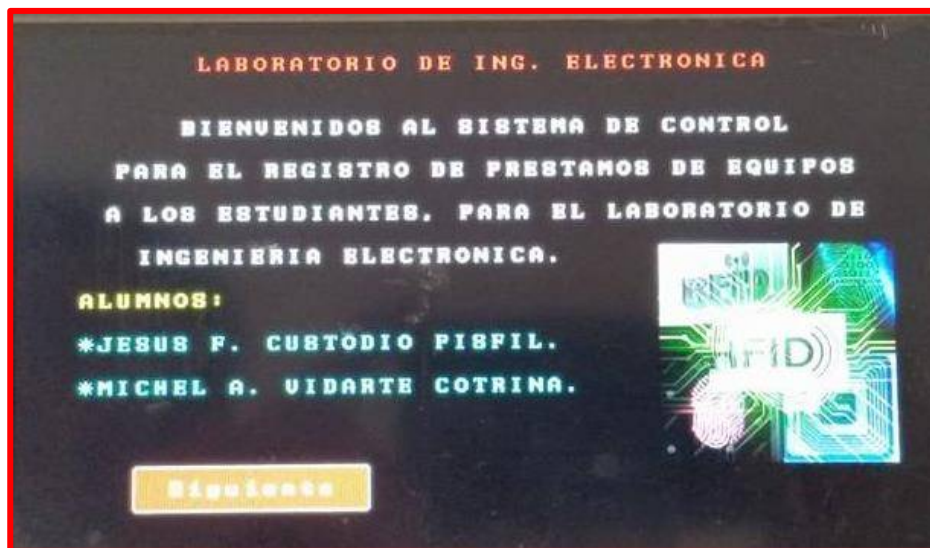
Figura 4.43. Esperando conexión con la PC de control.



Fuente: Propia de los autores

Una vez hecha la conexión entre la PC de control y el Módulo Electrónico se tiene una pantalla de bienvenida, con el nombre del proyecto de Tesis y con un botón “SIGUIENTE” que mostrara la siguiente ventana del interfaz, tal como se muestra en la figura 4.44.

Figura 4.44. Mensaje de bienvenida.



Fuente: Propia de los autores

4.3.1.2. Verificación del proceso de registro de usuarios.

En la figura 4.45 podemos ver un menú con tres opciones en las que el usuario quiera acceder tales como registro, logueo o activar el sensor de huella dactilar este último sirve para la verificación de la conectividad del sensor de huellas GT-511C3 con la PC de control.

Figura 4.45. Menú con opciones de registro y logueo.



Fuente: Propia de los autores

En las siguientes figuras se muestran los formularios de registro, previo a estos formularios, el usuario debe acercarse al administrador portando su DNI o carnet Universitario para que este le dé el permiso de poder registrarse en el sistema por medio de la PC de control. Ya el usuario siendo habilitado para su registro se le pedirá lo siguiente: Código de usuario, Nombres y Apellidos. Y por último el Módulo Electrónico pedirá la colocación de la huella dactilar del mismo.

Figura 4.46. Formulario para poner el código de usuario.



Fuente: Propia de los autores

Figura 4.47. Formulario para poner los Nombres de usuario.

LABORATORIO DE ING. ELECTRONICA

NOMBRES:

1 2 3 4 5 6 7 8 9 0

q w e r t y u i o p

a s d f g h j k l ~

z x c v b n m BORRAR

Back abc Next

ASOCIACIÓN PROFESIONAL DE INGENIEROS ELECTRICOS
FACFyM

Fuente: Propia de los autores

Figura 4.48. Formulario para poner los Apellidos de usuario.

LABORATORIO DE ING. ELECTRONICA

APELLIDOS:

1 2 3 4 5 6 7 8 9 0

q w e r t y u i o p

a s d f g h j k l ~

z x c v b n m BORRAR

Back abc Next

ASOCIACIÓN PROFESIONAL DE INGENIEROS ELECTRICOS
FACFyM

Fuente: Propia de los autores

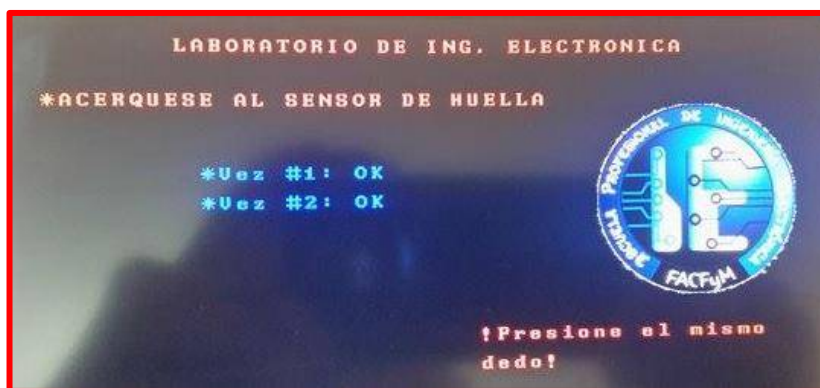
Figura 4.49. Petición de la huella dactilar del usuario.



Fuente: Propia de los autores

En la figura 4.49 se puede apreciar que el Módulo Electrónico pide al usuario se acerque al sensor de huella dactilar para que esta misma la registre. Esta inscripción se hace por 3 veces presionando el mismo dedo para que la base de datos del sensor lo pueda procesar y comparar y que esta misma pueda estar libre de algunas vulnerabilidades. Esto se puede ver en la figura 4.50.

Figura 4.50. Petición de la huella dactilar del usuario por 3 veces.



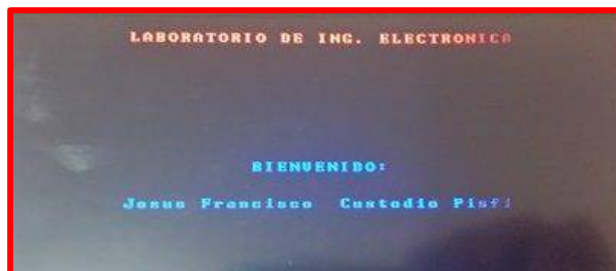
Fuente: Propia de los autores

Una vez el usuario hecho su registro de manera satisfactoria, esta información es almacenada en una base de datos y puede ser visualizada a través de la Interfaz Gráfica de Usuario (aplicación hecha en Visual Studio para la PC de control; ver figura 4.64); lo que permite al usuario poder entrar al sistema haciendo pedidos, devoluciones y transferencia de equipos sin ninguna dificultad.

4.3.1.3. Verificación del proceso de logueo.

En el momento de hacer un pedido de algún equipo, habiendo sido el usuario registrado anteriormente de manera exitosa, se procede al logueo de usuario. Para esto el usuario tiene que llenar un formulario en la que se le pide que ingrese su Código y luego se identifique con su huella dactilar. Una vez realizado este proceso el sistema lo reconocerá y encenderá un led avisando que este se ha logueado exitosamente mientras que en la pantalla táctil se visualiza un mensaje de bienvenida; y es entonces donde el usuario podrá tener acceso a la petición de equipos o si fuese el caso devolverlos o transferirlos. En la figura 4.51 se puede ver como el usuario ha sido reconocido.

Figura 4.51. Mensaje de bienvenida al usuario.

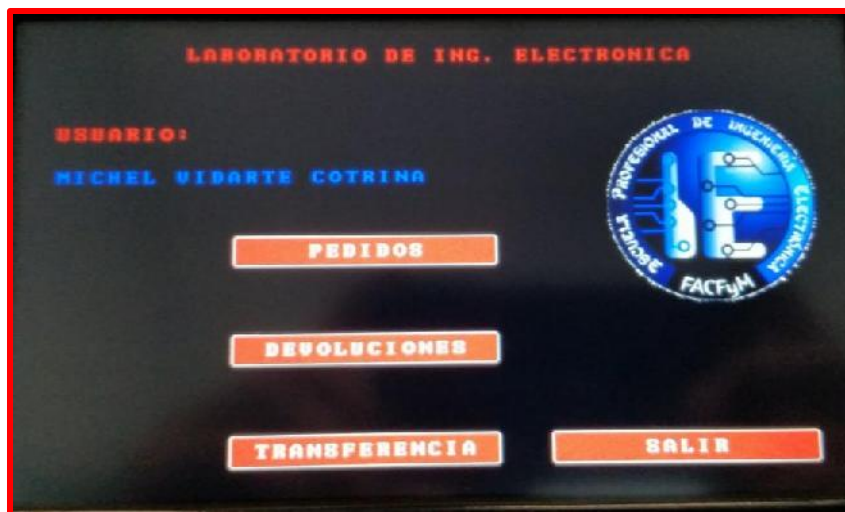


Fuente: Propia de los autores

4.3.1.4. Verificación del proceso de pedido de equipos.

Luego de que el usuario se haya logueado con éxito, podrá pedir, devolver o transferir equipos.

Figura 4.52. Menú de pedido, devoluciones y transferencia de equipos.



Fuente: Propia de los autores

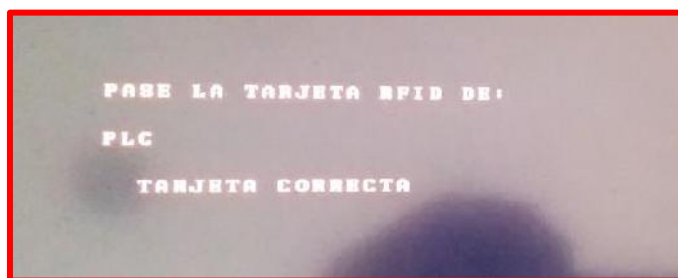
Si el usuario desea hacer un pedido deberá pulsar sobre el botón “PEDIDOS” y automáticamente aparecerá una ventana con todos los equipos que se encuentran registrados en la base de datos, se podrá seleccionar cualesquiera el usuario y deberá pasar cada equipo seleccionado por la lectora NFC/RFID para que este guarde el código del Tag de cada equipo. Si el usuario colocase un equipo con diferente código de Tag del que haya seleccionado aparecerá un mensaje diciendo que la tarjeta es incorrecta. El sistema también guardará en su base de datos, el usuario que hizo el pedido con hora y fecha de salida del almacén. En la figura 4.53 se puede apreciar el listado de algunos equipos registrados en la base de datos.

Figura 4.53. Ventana de selección de equipos.



Fuente: Propia de los autores

Figura 4.54. Petición de los equipos pedidos



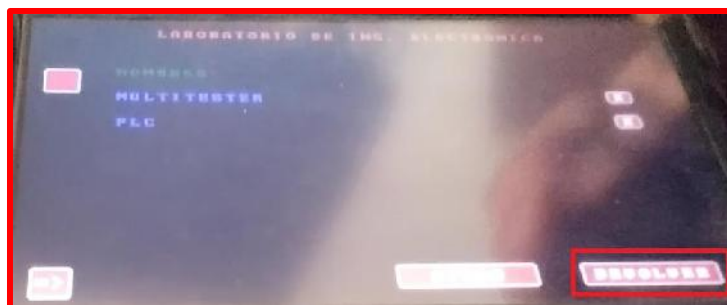
Fuente: Propia de los autores

Luego de que el usuario haya seleccionado los equipos que va a necesitar en sus prácticas académicas, podrá tenerlos consigo por un plazo de 12 horas, las cuales pasado el plazo indicado el usuario ya no podrá hacer más pedidos hasta que devuelva todos los equipos que sacó.

4.3.1.5. Verificación del proceso de Devolución de equipos.

Al momento en que un usuario llega a devolver los equipos, este deberá loguearse (ver figura 4.51) y luego pulsar sobre el botón “DEVOLUCIONES” (ver figura 4.52) y automáticamente aparecerá una ventana con todos los equipos que el usuario haya pedido antes; seleccionará los equipos que desea devolver y deberá pasarlo por la lectora NFC/RFID para que este compare los códigos del Tag del equipo, y pueda eliminarlo de la lista de pedidos del usuario.

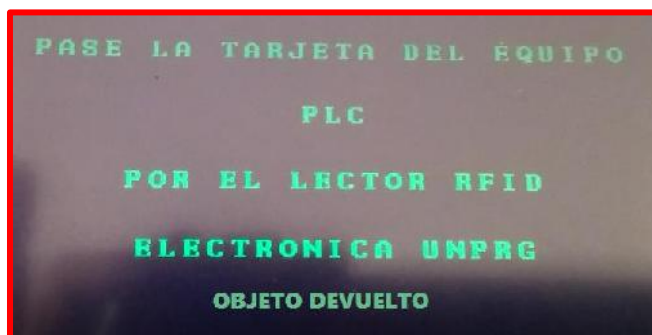
Figura 4.55. Ventana con la lista de equipos para devolver.



Fuente: Propia de los autores

En la figura 4.56 muestra como el Módulo Electrónico pide pasar el Tag del equipo por la lectora NFC/RFID.

Figura 4.56. Ventana de equipo devuelto



Fuente: Propia de los autores

En la figura 4.57 se muestra el lector NFC/RFID en el momento que está haciendo lectura del Tag del equipo.

Figura 4.57. Lector NFC/RFID en funcionamiento.



Fuente: Propia de los autores

Si fuese el caso en que el usuario al momento de devolver un equipo, selecciona un equipo del listado, y pasa por la lectora NFC/RFID un equipo con diferente código de Tag, se mostrará un mensaje diciendo que la tarjeta del equipo es incorrecta; el usuario tendrá tres intentos sino volverá a la ventana anterior.

4.3.1.6. Verificación del proceso de Transferencia de equipos.

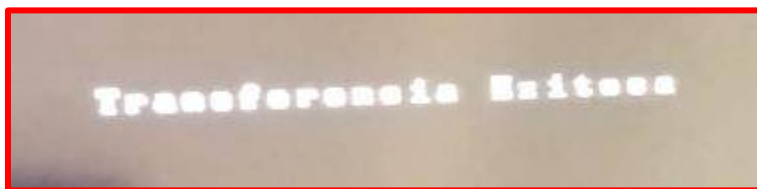
Para el proceso de transferencia de equipos es necesaria la participación de dos usuarios, para ello usuario 1 (alumno 1) debe hacer un pedido y transferirlo al usuario 2 (alumno 2), donde este último por medio del llenado de un formulario deberá poner su código de usuario y colocar su huella dactilar para que la transferencia sea exitosa, previo a este proceso ambos deben estar debidamente registrados.

Figura 4.58. Pregunta si el usuario 1 desea transferir los equipos al usuario2.



Fuente: Propia de los autores

Figura 4.59. Ventana Transferencia exitosa.



Fuente: Propia de los autores

4.3.2. Funcionamiento del Interfaz Gráfica de Usuario (PC de Control)

La interfaz gráfica de usuario es el complemento para el control de este sistema. Es una aplicación hecha en Visual Studio donde permite obtener información de los datos leídos por el hardware del sistema, las cuales el administrador a cargo puede configurar sus características de forma manual, es decir tiene varias opciones de visualización.

La interfaz gráfica de usuario consta de 3 partes importantes, las cuales son:

- Pantalla de Inicio.
- Pantalla de LOGIN.
- Pantalla de Menú:

- ✓ Usuarios {
 - Agregar usuario*
 - Modificar usuario*
 - Eliminar usuario*
 - Buscar usuario*
 - Imprimir lista*
 - Limpiar*
 - Cerrar*

- ✓ Equipos {
 - Agregar Area de Electronica*
 - Agregar Nombre de Objeto*
 - Nuevo equipo*
 - Eliminar equipo*
 - Buscar equipos*
 - Mostrar lista de equipos*
 - Imprimir lista*
 - Limpiar*
 - Equipos disponibles*

- ✓ Pedidos {
 - Mostrar pedidos*
 - Imprimir lista*
 - Limpiar*
 - Transferencia de equipos*

- ✓ Devoluciones {
 - Mostrar Devoluciones*
 - Imprimir lista*
 - Limpiar*

- ✓ Administradores {*Registrar*

- ✓ Herramientas {
 - Conectar*
 - Generar Informe*
 - Ver Modulos*
 - Resetear Pantalla*

- ✓ Conectar {*Desconectar*

- ✓ Generar Informe

- ✓ Ver Módulos {
 - Pantalla tactil*
 - Sensor de huella*
 - NFC RFID*

- ✓ Resetear Pantalla

4.3.2.1. Inicio

Es la pantalla inicial de la aplicación del sistema. Consta de una Barra progresiva de procesos que indica que se está cargando la conexión con el módulo electrónico, además de un botón Administrar que permite entrar a la base de datos del sistema sin necesidad de conectarse al Módulo Electrónico.

Figura 4.60. Pantalla de Inicio.



Fuente: Propia de los autores


4.3.2.2. Login

Esta parte de la aplicación del sistema es para la seguridad de la información almacenada en la base datos, de las cuales exige para su ingreso un nombre de usuario y una contraseña, por defecto en el sistema vacío está dada con:

- **Nombre de Usuario:** admin
- **Contraseña:** 1234

Estos datos son para el uso exclusivo del Administrador

Figura 4.61. Pantalla de Login.



Fuente: Propia de los autores

4.3.2.3. Menú

Esta pantalla de Menú está diseñada de forma interactiva para el fácil manejo del administrador del sistema. En esta pantalla se puede acceder a todas las funcionalidades del sistema tanto para el registro manual de alumnos y de equipos, como para el control automático de estos mismos; es decir ofrece opciones para verificar la conexión con el modulo electrónico y la verificación de sus dispositivos externos(sensor GT-511C3, lector NFC RFID).

Figura 4.62. Pantalla de Menú Principal.



Fuente: Propia de los autores

A continuación se describe cada uno de los elementos que se encuentran en el Menú de la aplicación.

➤ **Menú Usuarios**

En el menú usuarios encontramos todo lo necesario para agregar, modificar, eliminar, buscar usuarios, ver lista de detalles de todos los usuarios registrados de manera manual y automática, etc.

Figura 4.63. Menú Usuarios.



Fuente: Propia de los autores

AGREGAR USUARIO

En este ítem se puede ingresar el registro manual de un nuevo alumno. En la pantalla del ítem podemos ingresar los datos más importantes del nuevo usuario.

- **Código:** Permite ingresar el código único de cada alumno.
- **Nombres:** Permite ingresar el nombre completo del alumno.

- **Apellidos:** Permite ingresar los apellidos completos del alumno.

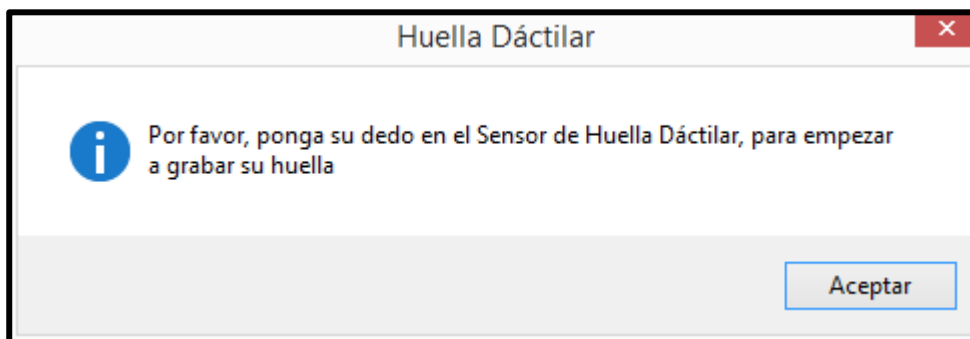
Figura 4.64. Ventana Agregar Usuarios.



Fuente: Propia de los autores

Cuando el administrador haya registrado al nuevo usuario con sus datos importantes, el sistema pedirá el registro de su huella dactilar, el cual se le pedirá al usuario poner su huella sobre el sensor GT-511C3 por tres veces para tener un registro exitoso.

Figura 4.65. Mensaje para grabar huella.

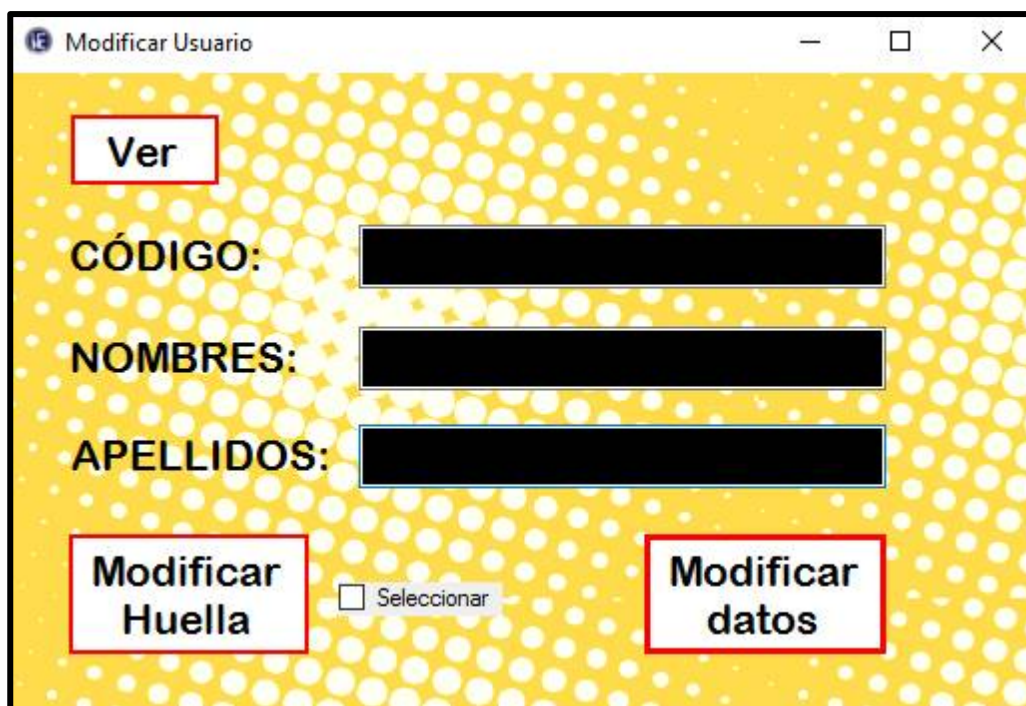


Fuente: Propia de los autores

MODIFICAR USUARIO

En este ítem podemos modificar la información del usuario.

Figura 4.66. Ventana Modificar Usuario.



Fuente: Propia de los autores

Como podemos ver en la figura 4.62 se puede modificar los datos más importantes del usuario, con el botón Ver podemos buscar el usuario que se desea modificar y editando las casillas de ingreso más el botón Modificar datos, la información del usuario queda modificada, además con la opción Modificar huella antes seleccionada con un check, se puede modificar el registro de la huella digital poniendo el dedo que se desea registrar por tres veces según los requerimientos del sistema.

ELIMINAR USUARIO

En este ítem se elimina los usuarios registrados en el sistema.

Figura 4.67. Ventana Eliminar Usuario.



The screenshot shows a web application window titled "Eliminar Usuario". The window has a yellow background with a white polka-dot pattern. At the top left, there is a button labeled "Ver". Below it, there are three input fields with labels "CÓDIGO:", "NOMBRES:", and "APELLIDOS:". At the bottom right, there is a button labeled "Eliminar".

Fuente: Propia de los autores

Con este ítem pulsando sobre la opción Ver podemos buscar el usuario que se desea eliminar, o simplemente escribir los datos en las casillas de ingreso para eliminarlos del sistema.

BUSCAR USUARIO

Este ítem permite buscar los usuarios en el sistema y mostrarlos.

Permite escribir cualquier dato del usuario y buscarlo en la base de datos del sistema, o también permite mostrar todos los usuarios registrados y buscarlos en la lista.

Figura 4.68. Ventana Buscar Usuarios.

BUSCAR USUARIOS

CÓDIGO: 081697H

NOMBRES:

APELLIDOS:

Buscar

ID	CODIGO	NOMBRES	APELLIDOS
0	081697H	Jesus Francisco	Custodic Pisfi

Mostrar todos

Aceptar

Fuente: Propia de los autores

IMPRIMIR LISTA

Este ítem muestra un informe completo con la lista de usuarios registrados en la base de datos del sistema, sucede lo mismo para los demás ítems de los otros menús.

LIMPIAR

El ítem Limpiar borra toda la información de los usuarios registrados en la base de datos del sistema, sucede lo mismo para los demás ítems de los otros menús.

➤ *Menú Equipos*

En el menú equipos encontramos todo lo necesario para agregar, modificar, eliminar, buscar equipos, ver lista de detalles de todos los equipos registrados de manera manual y automática; etc. Podemos agregar un Área de estudio donde almacenar y organizar los equipos, y asignarle un nombre de objeto a estos mismos.

Figura 4.69. Menú Equipos.



Fuente: Propia de los autores

AGREGAR AREA DE ELECTRONICA

Este ítem permite categorizar los equipos en áreas de estudio, es decir separa los equipos en las áreas en las que se divide la Ingeniería Electrónica. Por ejemplo si agregamos un PLC, como la mayoría de estudiantes sabe este equipo pertenece al área de Control, e igual si agregamos un Multímetro este iría en el área de la Electrónica Digital, etc.

Figura 4.70. Ventana Agregar Áreas.



Fuente: Propia de los autores

AGREGAR NOMBRE DE OBJETOS

Este ítem permite darle un nombre de objeto a cada equipo; es decir categoriza a cada equipo en un nombre característico. Como por ejemplo si se desea agregar un osciloscopio; podemos encontrar cantidad de osciloscopios de diferentes tamaños modelos, marcas, series, etc.; a ello se debe la categorización en agregarle un nombre característico a cada equipo. Además de esto se le agrega un ID único a cada equipo a través del NFC/RFID.

Figura 4.71. Ventana Agregar Nombre de Objeto.



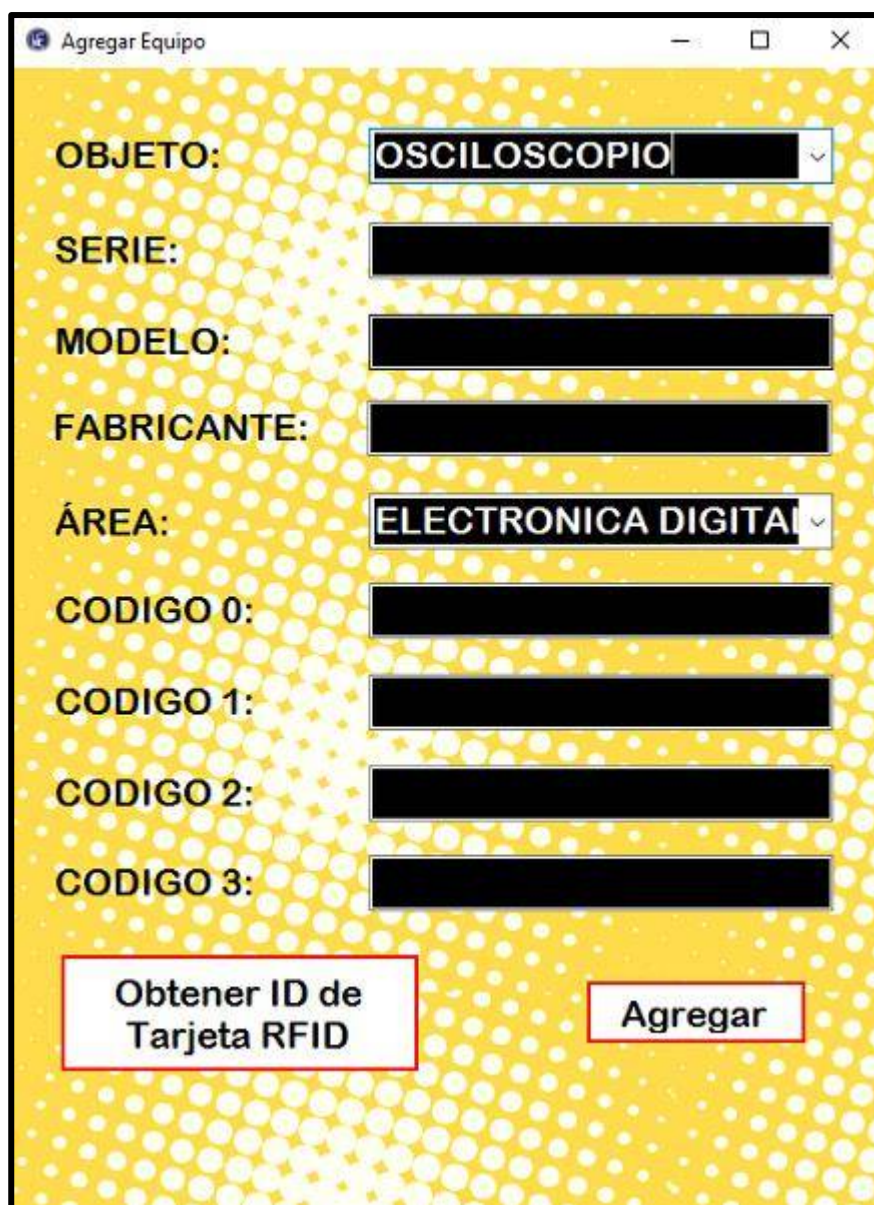
Fuente: Propia de los autores

NUEVO EQUIPO

Este es el ítem para añadir un nuevo equipo con sus características más importantes.

- **Objeto:** Permite la selección del objeto.
- **Serie:** Permite ingresar el número de serie del equipo.
- **Modelo:** Permite ingresar el modelo del equipo.
- **Fabricante:** Permite ingresar el nombre del fabricante.
- **Área:** Permite seleccionar el área en el que se desea categorizar al equipo.
- **Código:** Permite el ingreso manual del código de identificación del equipo.
- **Obtener ID de la tarjeta RFID:** esta opción permite el ingreso automático del código de identificación del equipo pasando el equipo el lector NFC/RFID.

Figura 4.72. Ventana Buscar Usuarios.



The image shows a software window titled "Agregar Equipo" with a yellow background and a white polka-dot pattern. The window contains a form with the following fields and controls:

- OBJETO:** A dropdown menu with "OSCILOSCOPIO" selected.
- SERIE:** A text input field.
- MODELO:** A text input field.
- FABRICANTE:** A text input field.
- ÁREA:** A dropdown menu with "ELECTRONICA DIGITAL" selected.
- CODIGO 0:** A text input field.
- CODIGO 1:** A text input field.
- CODIGO 2:** A text input field.
- CODIGO 3:** A text input field.

At the bottom of the form, there are two buttons:

- Obtener ID de Tarjeta RFID** (highlighted with a red border)
- Agregar** (highlighted with a red border)

Fuente: Propia de los autores

En la figura 4.69 se puede ver un mensaje en la que se pide pasar un Tag sobre la lectora NFC/RFID para poder completar el formulario de registro de equipos.

Figura 4.73. Mensaje para pasar el Tag en la lectora NFC/RFID.



Fuente: Propia de los autores

MODIFICAR EQUIPO

Este ítem permite editar las características del equipo.

Con la opción Ver podemos buscar el equipo que se desea modificar y con el botón Obtener ID de Tarjeta RFID se colocará el código único de identificación de los equipos para ser editados automáticamente pasando el equipo por el lector NFC/RFID.

Figura 4.74. Ventana Modificar Equipos.

Modificar Equipo

Ver

OBJETO: [Redacted]

SERIE: [Redacted]

MODELO: [Redacted]

FABRICANTE: [Redacted]

ÁREA: [Redacted] ▼

CODIGO 0: [Redacted]

CODIGO 1: [Redacted]

CODIGO 2: [Redacted]

CODIGO 3: [Redacted]

Obtener ID de Tarjeta RFID

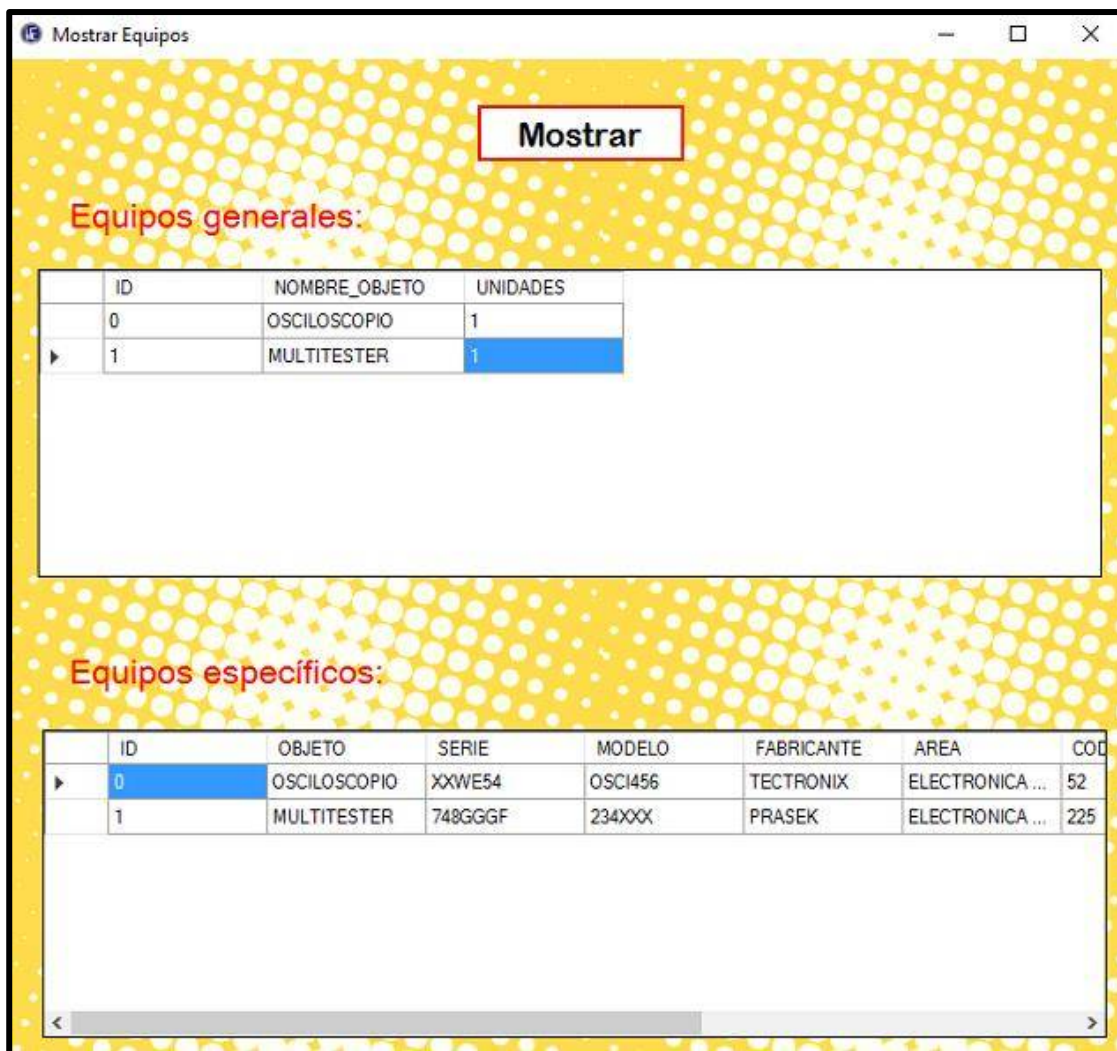
Modificar

Fuente: Propia de los autores

MOSTRAR LISTA DE EQUIPOS

Este ítem permite la visualización de todos los equipos registrados en la base de datos del sistema.

Figura 4.75. Ventana Mostrar Lista de Equipos.



Fuente: Propia de los autores

En Equipos generales muestra la lista de equipos con sus nombres y la cantidad disponibles. En cuanto en Equipos específicos muestra en detalles todas las características del equipo.

EQUIPOS DISPONIBLES

En este ítem muestra la cantidad de equipos disponibles para ser pedido por los usuarios.

Figura 4.76. Ventana Mostrar Equipos Disponibles.



Fuente: Propia de los autores

➤ Pedidos

En el menú Pedidos encontramos los eventos de pedidos hechos por los usuarios en tiempo real (hora y fecha).

Figura 4.77. Menú Pedidos.



Fuente: Propia de los autores

MOSTRAR PEDIDOS

En este ítem se aprecia todos los pedidos hechos por cada uno de los usuarios con las características de los equipos, nombre del usuario que los retiró e incluso con hora y fecha de salida del almacén del laboratorio. También se muestra la lista de pedidos hechos por el usuario cuando este no encuentre los equipos en el listado de la pantalla táctil; será definido en la lista de “Otros Pedidos”.

Figura 4.78. Ventana Mostrar Pedidos.

Mostrar

Pedidos:

ID	OBJETO	NOMBRES	APELLIDOS	FECHA
----	--------	---------	-----------	-------

Otros Pedidos:

ID	OBJETO	NOMBRES	APELLIDOS	FECHA
----	--------	---------	-----------	-------

Fuente: Propia de los autores

TRANSFERENCIA DE EQUIPOS

Para la transferencia de equipos a través de la aplicación es necesaria la participación de dos usuarios, como se puede ver la siguiente figura, se pide ingresar el código de cada uno de los dos usuarios.

Figura 4.79. Ventana Transferencia de equipos.

The image shows a software window titled "Transferencia". The main content area has a yellow background with a pattern of small white circles. At the top, the text "Transferencia de Equipos entre Usuarios" is displayed in blue. Below this, there are two input fields. The first is labeled "CÓDIGO 1" in a red-bordered box, followed by a white input field with a small downward arrow on the right. The second is labeled "CÓDIGO 2" in a red-bordered box, followed by another white input field with a small downward arrow on the right. In the bottom right corner, there is a green button with a red border labeled "TRANSFERIR".

Fuente: Propia de los autores

➤ Menú Devoluciones

En el menú Devoluciones encontramos los eventos de devolución hechos por los usuarios en tiempo real (hora y fecha).

Figura 4.80. Menú Devoluciones.



Fuente: Propia de los autores

MOSTRAR DEVOLUCIONES

En este ítem se aprecia todas las devoluciones hechos por cada uno de los usuarios con las características de los equipos, nombre del usuario que los retiró e incluso con hora y fecha en las que el usuario las devolvió al laboratorio.

Figura 4.81. Ventana Mostrar Devoluciones.

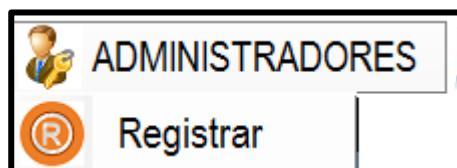


Fuente: propia de los autores

➤ Menú Administradores

En el Menú administradores se hace un registro privilegiado a los encargados del laboratorio.

Figura 4.82. Menú Administradores.



Fuente: Propia de los autores

REGISTRAR

En este ítem se hace el registro del encargado del laboratorio.

- **Administrador:** Es el nombre del administrador del laboratorio.
- **Contraseña:** Por seguridad se le pide ingresar con una contraseña única por cada administrador

Figura 4.83. Ventana Registro de Administradores.



The image shows a software window titled 'Registro_Usuarios'. The main content area has a yellow background with a pattern of small white dots. The title 'Registro de Administradores' is centered at the top. Below the title, there are three text input fields. The first is labeled 'Administrador:', the second 'Contraseña:', and the third 'Confirme Contraseña:'. Each field is currently empty. Below the 'Confirme Contraseña:' field, there is a checkbox with the label 'Mostrar Caracteres'. At the bottom of the window, there are three buttons: 'Limpiar', 'Registrar', and 'Cancelar', each with a red border.

Fuente: Propia de los autores

➤ *Menú Herramientas*

En el Menú Herramientas encontramos opciones extras en el sistema, como la conexión manual entre el interfaz de control y el modulo electrónico, la verificación de la conexión de los dispositivos externos en el sistema, también un informe completo de cada usuario; un reseteo de la pantalla táctil para en caso haya pérdida de comunicación con el sistema.

Hay dos maneras de encontrar estas opciones; la primera la encontramos en el Menú Principal y la segunda en el Menú Herramientas.

Figura 4.84. Menú Herramientas.



Fuente: Propia de los autores

CONEXION

Este ítem permite la conexión manual entre la PC de control y el módulo electrónico.

GENERAR INFORME

Este ítem genera un informe completo de cada uno de los usuarios registrados en la base de datos del sistema.

VER MODULOS

Este ítem permite verificar la comunicación de los dispositivos externos con el sistema.

Figura 4.85. Ventana para ver la conexión de los Módulos.



Fuente: Propia de los autores

- **Pantalla Táctil:** Permite cambiar el fondo de la pantalla táctil.
- **Sensor De Huella:** Permite verificar la comunicación del sensor de huella GT-511C3 con el sistema.
- **NFC RFID:** Permite verificar la comunicación del lector NFC/RFID con el sistema.
- **RESETEAR PANTALLA:** Este ítem permite el reseteo de la comunicación entre el modulo electrónico y la PC de control.

➤ *Habilitar Registro de Usuarios*

Permite habilitar el registro de los usuarios desde el modulo electrónico.

Figura 4.86. Habilitar registro de usuarios.



Fuente: Propia de los autores

4.4. Pruebas de demostración del funcionamiento del prototipo.

Para las pruebas de demostración del funcionamiento del prototipo se pidió a tres alumnos de la escuela profesional de Ingeniería Electrónica hacer la simulación del proceso de registro en la base de datos y el proceso de pedido y devolución de algunos equipos. Los alumnos se acercaron al prototipo (módulo electrónico) y realizaron los procesos. Los alumnos al finalizar las pruebas dieron sus puntos de crítica acerca del prototipo, todos favorables al buen funcionamiento del prototipo, de las cuales coincidieron en los siguientes puntos:

- Fácil manipulación de la pantalla interactiva del módulo electrónico.
- Mayor orden y conformidad al momento del pedido y la devolución de equipos.
- Adecuado funcionamiento del prototipo.
- Fácil manejo de los elementos que conforma el prototipo.

Figura 4.87. Prueba de funcionamiento del Prototipo.



Fuente: Propia de los autores

4.5. Costos del prototipo

Teniendo en cuenta todos los gastos que se han originado para realizar este Prototipo de un Sistema de Control para el Préstamo de Equipos del Laboratorio, se hace necesario desglosar ordenadamente todos los recursos empleados en este prototipo, es así que se tiene lo siguiente:

4.5.1. Costos de Componentes Electrónicos.

Tabla 4.6. Costos de Componentes Electrónicos.

PRODUCTOS	CANTIDAD	COSTO UNIDAD (S/)	COSTO TOTAL(S/)
Pantalla Táctil LCD/TFT de 7" (SSD1963)	1	245.00	245.00
Placa de Arduino Mega 2560	1	80.00	80.00
Shield LCD/TFT para Arduino Mega 2560	1	41.00	41.00
Placa de Arduino Uno	1	65.00	65.00
Shield ArduinoEthernet W5100	1	52.00	52.00
Sensor de huella digital (GT-511C3)	1	170.00	170.00
Conector de Sensor de Huella Digital	1	3.00	3.00
NFC / RFID Shield - 13,56 MHz	1	80.00	80.00
Tarjeta RFID (Tag)	6	1.5	9.00
Fuente de Alimentación de 2.5A– 5v	1	20.00	20.00
Conector RJ45	4	1.00	4.00
Pulsador	1	1.00	1.00
Interruptor unipolar luminoso rojo	1	6.20	6.20
Base enchufe Macho	1	1.00	1.00
Cordon de Alimentación AC	1	3.00	3.00
Transistor BC548	1	1.00	1.00
Placa de fibra de vidrio (7x7cm)	1	3.00	3.00
Soldadura de estaño	5	0.5	2.5
Cable Ethernet UTP	4	1.00	4.00
Cables Jumper	40	0.25	10
			810.50

Fuente: Propia de los autores

4.5.2. Costo de Materiales para Ensamblar el prototipo.

Tabla 4.7. Costos de Materiales para ensamblar el prototipo.

MATERIALES	CANTIDAD	COSTO UNIDAD (S/.)	COSTO TOTAL (S/.)
Módulo de madera para Ensamblaje del prototipo	1	150	150
Plataforma de Acrílico para ensamblar componentes	1	30	30
Accessories de Acrílico	-	10	10
Tornillos	36	0.10	3.60
Silicona	1	1.5	1.5
Soldimix	1	5.0	5.0
Pintura	1	7.0	7.0
Broca para madera	3	2.0	6
			213.10

Fuente: Propia de los autores

4.5.3. Costos de Investigación.

Tabla 4.8. Costos de Investigación.

	DETALLE	COSTO(S/)
Consultas	Internet, Biblioteca	50.00
Transporte	Movilización fuera y dentro de la Ciudad de Chiclayo	100.00
		150.00

Fuente: Propia de los autores

4.5.4. Costo de Diseño y Construcción del prototipo.

Tabla 4.9. Costos de Diseño y Construcción del Proyecto.

	ELEMENTOS	DETALLE	COSTO(S/)
Diseño y desarrollo de Software	SQL	Software para la Base de Datos	600
	Visual Basic	Software para la interacción de la PC de control con el Arduino y sus terminales.	1000
Diseño y Construcción de Hardware	Arduino Mega Arduino Uno	Programación de los controladores del Módulo Electrónico para que interactúen con la PC de Control.	1000
	Diseño de Circuitos	Conexión de los dispositivos del prototipo para que conformen el módulo electrónico	900
Implementación del Módulo	Pantalla LCD Hardware Otros	Acoplamiento de todos los dispositivos electrónicos en el Módulo de madera para su funcionamiento.	100
			3600

Fuente: Propia de los autores

4.5.5. Costos Administrativo.

Tabla 4.10. Costos Administrativos.

	DETALLE	COSTO(S/)
Tesis	Copias, Empastados, Hojas e Impresión	180
		180.00

Fuente: Propia de los autores

4.5.6. Costos del prototipo general.

Tabla 4.11. Costos del Prototipo.

ELEMENTOS	COSTOS TOTAL (S/)
Costos de componentes electrónicos	820.50
Costos de Materiales para ensamblar el prototipo	213.10
Costos de investigación	150.00
Costo de diseño y construcción del prototipo	3000
Costos administrativos	180.00
	S/4963.6

Fuente: Propia de los autores

CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Los sistemas biométricos y sistemas RFID no reemplazan a cualquiera de los sistemas actuales de identificación, sino más bien son un complemento a estos sistemas en las que se pretende mejorar los procesos de identificación, tales como seguridad, rapidez, y facilidad de gestión.

El uso de la plataforma de Arduino para la construcción del hardware, abarata los costos de implementación del sistema, puesto que es open source, y están disponibles en el mercado de manera global.

El uso de un sistema de huella dactilar en nuestro proyecto permitirá tener una mayor seguridad en cuanto al registro de los alumnos puesto que la huella dactilar es única para cada individuo y las características que se usan no cambian con el paso de la edad.

Al usar la tecnología RFID como sistema de identificación automática para el registro de equipos en el laboratorio de la EPIE, se obtendrá un mayor control de estos, puesto que esta tecnología es una mejora al código de barras, tanto en el método de lectura, velocidad de lectura, durabilidad, capacidad de información, etc.

Se diseñó un prototipo de sistema de control para el registro de préstamos de equipos, para los estudiantes de la EPIE- UNPRG, que cuenta con las siguientes características:

- ✓ Sistema con comunicación Ethernet con la PC de Control con reconocimiento automático.
- ✓ Sistema con reconocimiento de huella dactilar para el registro de alumnos.
- ✓ Sistema con tecnología RFID para el registro de equipos del laboratorio.
- ✓ El sistema cuenta con un software de aplicación (interfaz gráfico de usuario) para que el administrador tenga un mejor control y manipulación de los datos recogidos de los sensores.
- ✓ El prototipo cuenta con una base de datos incorporado en el software de aplicación para almacenar los datos obtenidos de los sensores (sensor de huella dactilar GT-511C3, shield NFC/RFID).
- ✓ Este prototipo es de código abierto puesto que tiene un puerto serial conectado con el Arduino Mega para modificar la programación de la pantalla LCD/TFT, Sensor de huella y shield NFC/RFID.
- ✓ Cuenta con una pantalla táctil interactiva en que los usuarios puedan hacer uso de él de manera fácil y amigable.

5.2. Recomendaciones

En este sistema como hemos mencionado es un prototipo de un sistema de control para el registro de préstamos de equipos para el laboratorio de Ingeniería Electrónica, las cuales hacemos las siguientes recomendaciones para los posibles cambios que se le pudiera dar:

- Este sistema está basado en la identificación de personas y de equipos, por lo que sería necesario agregar a nuestro sistema una cerradura electrónica para tener la seguridad en las puertas del laboratorio.
- En el mundo de la identificación automática, podemos encontrar muchos sistemas sofisticados al control de la identificación, se podría agregar una tecnología más a este sistema para seguir aumentando el nivel de automatización de este ambiente, como por ejemplo identificación por voz, por iris, por rostro, etc.
- En este prototipo se utilizó el shield NFC/RFID que tiene aproximado un rango de lectura de 1 pulgada, podría utilizarse un lector RFID con una cobertura más grande para obtener la lectura de los equipos a larga distancia y darle otras aplicaciones a este prototipo.
- Implementar un interfaz web para el control remoto y permitir al administrador ingresar desde cualquier host de la red y hacer el monitoreo de la información que se obtiene de los sensores. Al igual que el alumno pueda hacer sus pedidos desde la interfaz web para agilizar aún más el proceso de pedido de equipos.
- Implementar un sistema de dispensación automático de equipos, de manera que cuando los alumnos hagan sus pedidos, este seleccione el equipo y el mismo sistema se lo entregue.

REFERENCIAS BIBLIOGRÁFICAS

- [1] C. Medina Salgado, "Los Sistemas Automáticos de Identificación," [Online], *En Línea2*, Septiembre 1996. Disponible en:
<http://www.azc.uam.mx/publicaciones/enlinea2/indice.htm>
- [2] L. Hoyo Cordero and E. A. Arteaga Rico, "El código de Barras," [online], México. Centro Universitario Anglo Mexicano, s.f. Disponible en:
www.acmor.org.mx/sites/default/files/105.%20El%20codigo%20de%20barras.pdf
- [3]. R. G. Guzmán Martínez, "Implantación de código de barras en un almacén de electrodomésticos," Tesis M.S., Universidad Nacional Autónoma de México, México, 2008.
- [4] RRP Noticias, Radio Programas del Perú, (2012, octubre 26), "El código de barras cumplió 60 años de su aplicación en el comercio," [Emisora Radial], [Online], Perú. Disponible en: <http://rpp.pe/lima/actualidad/el-codigo-de-barras-cumplio-60-anos-de-su-aplicacion-en-el-comercio-noticia-534622>
- [5] Sistematic. "Historia del código de barras," Visitada marzo 2016 [Online]. Disponible en: <http://sistematic.pe/tienda/barcode.htm>
- [6] R. Carro Paz and D. Gonzáles Gómez, "Identificación Automática," Administración de las operaciones. Visitado marzo 2016. [Online]. Disponible en:
<http://nulan.mdp.edu.ar/id/eprint/1612>
- [7] Servicios informáticos KIFER S.L., "Introducción a los sistemas R.F.I.D.," Visitado enero 2016. [Online]. Disponible en: www.kifer.es/Recursos/Pdf/RFID.pdf

- [8] P. E. De la Cruz Vélez de Villa, M. Reyes Huamán and D. Bravo Loayza, “Radiofrecuencia de Identificación (RFID): microtecnología de gran impacto,” *Revista de Investigación de Sistemas e Informática* vol. 7, no. 2, pp. 77-86, diciembre 2010.
- [9] D. Kushner (2011 octubre 26), “The making of Arduino,” *IEEE Spectrum*. [Online]. Disponible en: <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino/0>
- [10] “¿Qué es Arduino?,” Visitado Noviembre 2015. [Online]. Disponible en: <http://arduino.cl/que-es-arduino/>
- [11] “Plataforma biométrica homíní,” Visitado en Diciembre 2015. [Online]. Disponible en: http://www.homini.com/new_page_5.htm
- [12] F. Serratosa, “La biometría para la identificación de las personas,” *Universitat Oberta de Catalunya*. Visitado marzo 2016. [Online]. Disponible en:
- [13] BBVA INNOVATION CENTER, “Tecnología biométrica,” *FINTECH SERIE BY innovation edge*, febrero 2016.
- [14] J.C. Santamaría Olivares, “Reconocimiento y validación de huellas dactilares utilizando una Red Neuronal,” *Revista de Tecnología*, vol. 7, no. 1, pp 55-60, enero – julio 2008.
- [15] M. Martínez Díaz, “Vulnerabilidades en Sistemas de Reconocimiento basados en huella Dactilar: Ataques HILL – CLIMBING,” Proyecto fin de carrera, Dep. Ingeniería Informática, Universidad Autónoma de Madrid, septiembre 2006.

[16] J. I. Portillo García, A. B. Bermejo Nieto and A. M. Bernardos Barbolla, “Tecnología e identificación por radiofrecuencia (RFID): aplicaciones en el ámbito de la salud,” Centro de Difusión de Tecnologías de la ETSIT-UPM, Círculo de Innovación en las Tecnologías de la Información.

[17] INTERMEC, “conceptos básicos de RFID: conocimiento y uso de la identificación por radio frecuencia,” Visitado en Enero 2016. [Online]. Disponible en:
www.intermec.com.mx/learning/content_library/white_papers/.../wpABC_MX.pdf

[18] Aprenderaprogramar.com, “Aprender Visual Basic desde cero,” Visitado en Enero 2016. [Online]. Disponible en:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=37&Itemid=61

[19] Kioskea.net (junio, 2014), “Introducción - Bases de datos,” Visitado en Marzo 2016. [Online]. Disponible en: <http://es.ccm.net/contents/66-introduccion-bases-de-datos>

[20] Microsoft Developer Network, “Microsoft SQL Server,” Visitado en Marzo 2016. [Online]. Disponible en: <https://msdn.microsoft.com/es-es/library/bb545450.aspx>

[21] “Lenguaje SQL,” Visitado en Abril 2016. [Online]. Disponible en: www.v-espino.com/~informatica/asi2/sghd/sql/SQL1.pdf

[22] “Arduino UNO & Genuino UNO,” Visitado en Noviembre 2015. [Online]. Disponible en: <https://www.arduino.cc/en/main/arduinoBoardUno#>

- [23] “Arduino MEGA 2560 & Genuino MEGA 2560,” Visitado en Noviembre 2015. [Online]. Disponible en: <https://www.arduino.cc/en/Main/arduinoBoardMega2560>
- [24] O. Torrente Artero, ARDUINO Curso práctico de formación, 1st ed. México: Ed. Alfaomega, febrero 2013.
- [25] “GT- 511C5 datasheet,” ADH Technology Co. Ltd.
- [26] “makerlab-electronics,” Visitado en Noviembre 2015. [Online]. Disponible en: <https://www.makerlab-electronics.com/>
- [27] “Posperú.com,” Visitado en Noviembre 2015. [Online]. Disponible en: <http://posperu.com/>
- [28] “Tiendas de computo,” Visitado en enero 2016. [Online]. Disponible en: <http://www.tiendadecomputoperu.com/>
- [29] “ds3 comunicaciones,” Visitado en enero 2016. [Online]. Disponible en: <http://ds3comunicaciones.com/cisco/WS-C2960C-12PC-L.html>
- [30] “GT- 511C3 datasheet,” ADH Technology Co. Ltd.

ANEXOS

ANEXO 1

PROGRAMACION USADA PARA EL ARDUINO UNO

```
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(169,254,239,75);
EthernetServer server(12);

void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  digitalWrite(2,HIGH); //PIN_AVISO_DE_TRANSIMISI3N(IRQ)
  digitalWrite(3,HIGH); //PIN_RESET
  delayMicroseconds(200);
  digitalWrite(3,LOW);
  Serial.begin(115200);
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("Server is at: ");
  Serial.println(Ethernet.localIP());
  delay(1000);
  Serial.println("Esperando Cliente...");
  delay(1000);
}
EthernetClient client;

void loop() {
  client = server.available();
  if (client) {
    while (client.connected())
    {
      if(Serial.available())
```

```

{
  String mje=Serial.readStringUntil('\n');
  mje.remove(mje.length()-1);
  client.print(mje);
}
int i=0;
String recibe="";
while(client.available())
{
  recibe = client.readStringUntil('\n');
  break;
}
if(recibe.length()>0)
{
  if(recibe == "INTERRUMPIR PANTALLA")
  {
    digitalWrite(3,HIGH); //RESET
    delayMicroseconds(100);
    digitalWrite(2,LOW); //PIN AVISO
    digitalWrite(3,LOW);
    delay(1000);
    digitalWrite(2,HIGH);
  }else Serial.println(recibe);
}
}
}

```

ANEXO 2

PROGRAMACION USADA PARA EL ARDUINO MEGA 2560

➤ **Programación usada para la pantalla táctil.**

```
#include <UTFT.h>
#include <UTFT_Buttons.h>
#include <UTouch.h>
#include <tinyFAT.h>
#include <UTFT_tinyFAT.h>
#include <SD_modificada.h>
#include <SPI.h>
extern uint8_t SmallFont[];
extern uint8_t BigFont[];
extern uint8_t Dingbats1_XL[];
UTFT      myGLCD(CTE70,38,39,40,41);
UTouch    myTouch(6,5,4,3,2);
UTFT_Buttons myButtons(&myGLCD, &myTouch);
UTFT_tinyFAT myFiles(&myGLCD);
char* files800[]={ "unprg.RAW","RFID.RAW"};
char* files[2];
#define imagedatatype unsigned int
extern imagedatatype unprg[];
int ID;
String NOMBRES;
String APELLIDOS;
String reconocimiento;
unsigned int num_lineas_texto=0;
int led_RFID=A10;
int led_SENSOR_HUELLA=A11;
int PIN_AVISO = HIGH;
unsigned int valor_color = 0x0000;
HardwareSerial &VISUAL_BASIC = Serial1;
HardwareSerial &SENSOR_HUELLA = Serial2;
#include "Sensor_Huella_Dactilar_Sparkfun.h"
SENSOR_HUELLA_SPARKFUN HUELLA;
int secuencia;
```

```

String habilitar_sensor_huella="TRUE";

void setup() {
  pinMode(A9,INPUT);
  PIN_AVISO = digitalRead(A9);
  Serial.begin(115200);
  VISUAL_BASIC.begin(115200);
  SENSOR_HUELLA.begin(9600);
  pinMode(led_RFID,OUTPUT);
  pinMode(led_SENSOR_HUELLA,OUTPUT);
  digitalWrite(led_RFID,LOW);
  digitalWrite(led_SENSOR_HUELLA,LOW);

  if(PIN_AVISO==0) //AVISO
  {
    while(true)
    {
      String msje=Visual_readString();
      if(msje=="CAMBIAR FONDO")
      {
        String msje2 = Visual_readString();
        valor_color = msje2.toInt();
        break;
      }
      if(msje=="RESETEAR")
      {
        break;
      }
      if(msje == "IDENTIFICAR SENSOR DE HUELLA")
      {
        Visual_print("IDENTIFICAR SENSOR DE HUELLA");
        HUELLA.OPEN(VISUAL_BASIC);
        break;
      }
      if(msje == "BORRAR HUELLAS")
      {
        Visual_print("BORRAR HUELLAS");
        HUELLA.DELETEALL(VISUAL_BASIC);
        break;
      }
    }
  }
}

```

```

if(msje==comando(0x03)) //"GRABAR HUELLA"
{
  Visual_print(0x03); //"GRABAR HUELLA"
  String msje2= Visual_readString(); //La PC me envía el ID de la persona.
  HUELLA.OPEN(Serial);
  HUELLA.CmosLed(1);
  HUELLA.DELETEID(msje2.toInt());
  HUELLA.EnrollID(msje2.toInt(),VISUAL_BASIC,0);
  HUELLA.CmosLed(0);
  HUELLA.CLOSE();
  break;
}
if(msje=="IDENTIFICAR R F I D")
{
  Visual_print("IDENTIFICAR R F I D");
  verificar_conexion_RFID(VISUAL_BASIC);
  break;
}
if(msje=="LEER ID TARJETA")
{
  Visual_print("LEER ID TARJETA");
  obtener_id_targeta_rfid();
  break;
}
}
}

myGLCD.InitLCD(LANDSCAPE);
myGLCD.fillScr(valor_color);
myGLCD.setFont(BigFont);
myTouch.InitTouch(LANDSCAPE);
myTouch.setPrecision(PREC_EXTREME);
myButtons.setTextFont(BigFont);
myButtons.setSymbolFont(Dingbats1_XL);
file.initFAT();
inicializar_SD();
files[0] = files800[0];
files[1] = files800[1];
myGLCD.setColor(VGA_RED);
myGLCD.print("ESPERANDO CONEXION CON LA PC",CENTER,200);

```

```

myGLCD.print("ELECTRONICA UNPRG",CENTER,250);
while(true)
{
    String inicio = Visual_readString();
    int t_inicio=(inicio.length()*300/21;
    myGLCD.setFont(BigFont);
    myGLCD.setColor(0,255,0);
    myGLCD.print("                                ",(800-600)/2, 420);
    myGLCD.print(inicio,(800-t_inicio)/2, 420);
    if(inicio == "Cliente Conectado!") {delay(500); break;}
}
habilitar_sensor_huella = Visual_readString();
myGLCD.fillScr(valor_color);
}

void loop() {
    myGLCD.setBackColor(VGA_TRANSPARENT);
    myGLCD.setColor(VGA_WHITE);
    myGLCD.print("BIENVENIDOS AL SISTEMA DE CONTROL",CENTER,80);
    myGLCD.print("PARA EL REGISTRO DE PRESTAMOS DE
EQUIPOS",CENTER,120);
    myGLCD.print("A LOS ESTUDIANTES, PARA EL LABORATORIO
DE",CENTER,160);
    myGLCD.print("INGENIERIA ELECTRONICA.",100,200);
    myGLCD.setColor(VGA_YELLOW);
    myGLCD.print("ALUMNOS:",50,240);
    myGLCD.setColor(VGA_AQUA);
    myGLCD.print("*JESUS F. CUSTODIO PISFIL.",50,280);
    myGLCD.print("*MICHEL VIDARTE COTRINA.",50,320);
    figurita_rfid(550,200);
    int btn_presionado,siguiente;
    siguiente=myButtons.addButton(100,400,200,40,"Siguiete");
    myButtons.setButtonColors(VGA_WHITE,VGA_GRAY,VGA_WHITE,VGA_BLUE,V
GA_RED); //ESTABLECE EL COLOR ROJO AL BOTON
    myButtons.drawButton(siguiente);
    Visual_print(0x00);
    Visual_print(0x01);
    Visual_print(0x02);
    while(true)
    {

```

```

    if(myTouch.dataAvailable() == true)
    {
        btn_presionado=myButtons.checkButtons();
        if(btn_presionado==siguiente) menu_registro_logueo();
    }
}

word res;
void encabezado(int x,int y)
{
    myGLCD.setFont(BigFont);
    myGLCD.setColor(VGA_RED);
    myGLCD.setBackColor(VGA_TRANSPARENT);
    myGLCD.print("LABORATORIO DE ING. ELECTRONICA", 150, 20);
    res=myFiles.loadBitmap(x, y, 200, 200, files[0]);//800X480
    if (res!=0)
    {
        if (res==0x10)
        {
            myGLCD.print("File not found...", 0, 0);
            myGLCD.print(files[0], 0, 14);
        }
        else
        {
            myGLCD.print("ERROR: ", 0, 0);
            myGLCD.printNumI(res, 56, 0);
        }
        delay(1500);
    }
    myGLCD.setFont(SmallFont);
}

word res2;
void figurita_rfid(int x,int y)
{
    myGLCD.setFont(BigFont);
    myGLCD.setColor(VGA_RED);
    myGLCD.setBackColor(VGA_TRANSPARENT);
    myGLCD.print("LABORATORIO DE ING. ELECTRONICA", 150, 20);
    res2=myFiles.loadBitmap(x,y,200,200,files[1]);//800X480

```

```

if (res2!=0)
{
  if (res2==0x10)
  {
    myGLCD.print("File not found...", 0, 0);
    myGLCD.print(files[1], 0, 14);
  }
  else
  {
    myGLCD.print("ERROR: ", 0, 0);
    myGLCD.printNumI(res2, 56, 0);
  }
  delay(1500);
}
myGLCD.setFont(SmallFont);
}
bool activar_sensor_huella=true;
void menu_registro_logueo()
{
  int btn_presionado;
  int btn_registro;
  int btn_logueo;
  int btn_sensor_huella;
  myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
  myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA DE COLOR
AMARILLO.
  int posH_y=90;
  encabezado(800-200-40,90);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(255, 0, 0);
  myGLCD.print("USUARIO: ", 30, posH_y+15);
  myGLCD.setColor(255, 255, 255);
  myGLCD.setFont(BigFont);
  myButtons.setButtonColors(VGA_WHITE,VGA_WHITE,VGA_WHITE,VGA_YELLOW,
W,VGA_RED);
  btn_registro=myButtons.addButton( 200, 200, 250, 30,"REGISTRO");
  btn_logueo=myButtons.addButton( 200, 300, 250, 30,"LOGUEO");
  if(habilitar_sensor_huella=="TRUE")
  btn_sensor_huella=myButtons.addButton(100,400,500,30,"ACTIVAR SENSOR DE
HUELLA");

```



```

myButtons.drawButtons();
while(true)
{
    if(myTouch.dataAvailable() == true)
    {
        do{
            btn_presionado = myButtons.checkButtons();
        }while(!(btn_presionado != -1));
        if(btn_presionado == btn_registro)
        {
            Visual_print("HABILITAR REGISTRO");
            String habilitacion = Visual_readString();
            if(habilitacion=="HABILITADO")
            {
                String codigo="",nombres="",apellidos="";
                secuencia=1;
                condiciones: switch(secuencia)
                {
                    case 0: menu_registro_logueo(); break;
                    case 1: codigo=ingresa("CODIGO:",codigo); secuencia++; break;
                    case 2: nombres=ingresa("NOMBRES:",nombres); secuencia++; break;
                    case 3: apellidos=ingresa("APELLIDOS:",apellidos); secuencia++; break;
                    case 4: goto REGISTRO;
                    default: ;
                }
                goto condiciones;
            REGISTRO: Visual_print("REGISTRO");
            Visual_print(codigo);
            Visual_print(nombres);
            Visual_print(apellidos);
            int ID=Visual_readString().toInt();
            if(habilitar_sensor_huella=="TRUE")
            {
                HUELLA.OPEN(Serial);
                HUELLA.CmosLed(1);
                HUELLA.registrar_huella(ID);
                HUELLA.CmosLed(0);
                HUELLA.CLOSE();
            }
            delay(2000);
        }
    }
}

```

```

        menu_registro_logueo();
    }else{
        myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
        myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA DE
COLOR AMARILLO.
        myGLCD.setColor(VGA_GREEN);
        myGLCD.setBackColor(VGA_TRANSPARENT);
        myGLCD.print("Por favor acerquece con su DNI",CENTER,200);
        myGLCD.print("al Administrador",CENTER,250);
        myGLCD.print("ELECTRONICA UNPRG",CENTER,300);
        delay(5000);
        menu_registro_logueo();
    }
}
if(btn_presionado == btn_logueo)
{
    logueo();
}
if(btn_presionado == btn_sensor_huella)
{
    if(activar_sensor_huella == true)
    {
        HUELLA.OPEN(Serial);
        HUELLA.CmosLed(1);
        myButtons.relabelButton(btn_sensor_huella,"DESACTIVAR SENSOR DE
HUELLA");
        myButtons.drawButton(btn_sensor_huella);
        activar_sensor_huella = false;
    }else{
        HUELLA.CmosLed(0);
        HUELLA.CLOSE();
        myButtons.relabelButton(btn_sensor_huella,"ACTIVAR SENSOR DE
HUELLA");
        myButtons.drawButton(btn_sensor_huella);
        activar_sensor_huella = true;
    }
}
}
}
}
}

```

```

void Visual_print(String msje)
{
    VISUAL_BASIC.println(msje);
    delay(100);
}
void Visual_print(char msje)
{
    VISUAL_BASIC.println(msje);
    delay(100);
}
String Visual_readString()
{
    String msje;
    while(true)
    {
        if(VISUAL_BASIC.available()>0)
        {
            msje=VISUAL_BASIC.readStringUntil('\n');
            msje.remove(msje.length()-1);
            break;
        }
    }
    return msje;
}
String comando(char valor_hexadecimal)
{
    return String(valor_hexadecimal);
}
boolean comparar_vectores(byte vector1[],byte vector2[],int inicio,int LEN)
{
    for(int m=0; m<LEN; m++)
    {
        if(vector1[m]!=vector2[m+inicio]) return false;
    }
    return true;
}
void tren_pulsos_cuadrados(int pin)
{
    for(int k=0; k<40; k++)
    {

```

```

    digitalWrite(pin,HIGH);
    delay(50);
    digitalWrite(pin,LOW);
    delay(50);
  }
}
void logueo()
{
    iniciar_de_nuevo:
    secuencia=1;
    String codigo=ingresa("CODIGO:", ""); secuencia++;
    if(secuencia==0) menu_registro_logueo();
    Visual_print("CODIGO USUARIO");
    Visual_print(codigo);
    if(Visual_readString() == "CODIGO USUARIO")
    {
        reconocimiento = Visual_readString();
        if(reconocimiento == "USUARIO ENCONTRADO")
        {
            myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
            myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA
DE COLOR AMARILLO.
            myGLCD.setColor(VGA_RED);
            myGLCD.setBackColor(VGA_TRANSPARENT);

            if(habilitar_sensor_huella=="TRUE")
            {
                myGLCD.print("Por favor pon tu huella dactilar",CENTER,200);
                myGLCD.print("... Identificando Usuario ...",CENTER,250);
                myGLCD.print("ELECTRONICA UNPRG",CENTER,300);
                HUELLA.OPEN(Serial);
                HUELLA.CmosLed(1);
                ID = Visual_readString().toInt();
                int id_sensor = HUELLA.IdentificarHuella();
                HUELLA.CmosLed(0);
                HUELLA.CLOSE();
                if(id_sensor != -1)
                {
                    if(ID == id_sensor)
                    {

```

```

        Visual_print("USUARIO IDENTIFICADO");
        tren_pulsos_cuadrados(led_SENSOR_HUELLA);
        NOMBRES = Visual_readString();
        APELLIDOS = Visual_readString();
        cabecera();
        myGLCD.print("BIENVENIDO: ",CENTER,200);
        myGLCD.print(NOMBRES+" "+APELLIDOS,CENTER,250);
        delay(2000);
        menu_pedidos_devoluciones();
        while(true);
    }else{
        Visual_print("USUARIO NO IDENTIFICADO, ACCESO
DENEGADO");
        goto iniciar_de_nuevo;
    }
}else{
    Visual_print("USUARIO NO IDENTIFICADO, ACCESO
DENEGADO");
    goto iniciar_de_nuevo;
}
}else{
    ID = Visual_readString().toInt();
    Visual_print("USUARIO IDENTIFICADO");
    tren_pulsos_cuadrados(led_SENSOR_HUELLA);
    NOMBRES = Visual_readString();
    APELLIDOS = Visual_readString();
    cabecera();
    myGLCD.print("BIENVENIDO: ",CENTER,200);
    myGLCD.print(NOMBRES+" "+APELLIDOS,CENTER,250);
    delay(2000);
    menu_pedidos_devoluciones();
    while(true);
}
}else goto iniciar_de_nuevo;
}else{
    HUELLA.CmosLed(1);
    goto iniciar_de_nuevo;
}
}

```

➤ **Programación usada para el pedido, devolución y transferencia de equipos.**

```
String lista_objetos[50];
String lista_id_objetos[50];
void menu_pedidos_devoluciones()
{
    int btn_presionado;
    int btn_pedidos;
    int btn_devol;
    int btn_transferencia;
    int btn_salir;
    myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
    myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA DE COLOR
AMARILLO.
    int posH_y=90;
    encabezado(800-200-40,90);
    myGLCD.setFont(BigFont);
    myGLCD.setColor(255, 0, 0);
    myGLCD.print("USUARIO: ", 30, posH_y+15);
    myGLCD.setColor(VGA_BLUE);
    myGLCD.print(NOMBRES+" "+APELLIDOS,30,posH_y+60);
    myGLCD.setColor(255, 255, 255);
    myGLCD.setFont(BigFont);
myButtons.setButtonColors(VGA_WHITE,VGA_WHITE,VGA_WHITE,VGA_YELLOW,VGA
A_RED);
    btn_pedidos=myButtons.addButton( 200, 220, 250, 30,"PEDIDOS");
    btn_devol=myButtons.addButton( 200, 320, 250, 30,"DEVOLUCIONES");
    btn_transferencia=myButtons.addButton(200,420,250,30,"TRANSFERENCIA");
    btn_salir=myButtons.addButton(500,420,250,30,"SALIR");
    myButtons.drawButtons();
    while(true)
    {
        if(myTouch.dataAvailable() == true)
        {
            do{
                btn_presionado = myButtons.checkButtons();
            }while(!(btn_presionado != -1));
            if(btn_presionado == btn_pedidos)
            {
```

```

        OBTENER_LISTA_MATERIALES();
    }
    if(btn_presionado == btn_devol)
    {
        //Visual_print(ID);
        OBTENER_LISTA_PENDIENTES();
    }
    if(btn_presionado == btn_transferencia)
    {
        menu_transferencia();
    }
    if(btn_presionado == btn_salir)
    {
        menu_registro_logueo();
    }
}
}
}
void menu_transferencia()
{
    int posH_y=90;
    int btn_presionado;
    myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
    myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA DE
COLOR AMARILLO.
    myGLCD.setBackColor(valor_color);
    myGLCD.setColor(VGA_WHITE);
    myGLCD.print("Desea transferir sus equipos a otro usuario: ", 30, posH_y+15);
    int btn_Si=myButtons.addButton(100,200,100,30,"SI");
    int btn_No=myButtons.addButton(600,200,100,30,"NO");
    myButtons.drawButtons();
    while(true)
    {
        if(myTouch.dataAvailable() == true)
        {
            do{
                btn_presionado = myButtons.checkButtons();
            }while(!(btn_presionado != -1));
            if(btn_presionado==btn_Si)
            {

```

```

myGLCD.print("IngreseCodigo del Usuario Destino", 30, posH_y+200);
delay(2000);
String codigo_nuevo="";
secuencia=1;
condiciones: switch(secuencia)
{
    case 0: menu_pedidos_devoluciones(); break;
    case 1: codigo_nuevo=ingresa("CODIGO:",codigo_nuevo); secuencia++;
break;

    case 2: goto salir;
    default: ;
}
goto condiciones;
salir: Visual_print("TRANSFERENCIA");
Visual_print(codigo_nuevo);
identificacion();
menu_pedidos_devoluciones();
}
if(btn_presionado==btn_No)
{
    menu_pedidos_devoluciones();
}
}
}
void cabecera()
{
    myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
    myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA DE COLOR
AMARILLO.
    myGLCD.setColor(VGA_RED);
    myGLCD.setBackColor(VGA_TRANSPARENT);
    myGLCD.print("LABORATORIO DE ING. ELECTRONICA", 150, 20);
    myGLCD.setColor(VGA_BLUE);
myButtons.setButtonColors(VGA_WHITE,VGA_GRAY,VGA_WHITE,VGA_BLUE,VGA_RE
D); //ESTABLECE EL COLOR ROJO AL BOTON
}
int valor=0;
int k=0;
int num_elementos=0;

```



```

void OBTENER_LISTA_PENDIENTES()
{
    Visual_print("LISTA PENDIENTES");
    while(!(Visual_readString()=="LISTA PENDIENTES"));
    valor=0;
    k=0;
    num_elementos=0;
    cabecera();
    char* lb[6]={"-","-","-","-","-","-"};
    int btn_item[6];
    boolean validar_devoluciones[6]={false,false,false,false,false,false};
    const int pos_x=100;
    const int pos_y=140;
    const int btn_x=600;
    int k=0;
    int btn_presionado;
    num_lineas_texto=0;
    myGLCD.setColor(VGA_GREEN);
    myGLCD.print("NOMBRES:",pos_x,pos_y-45);
    myGLCD.setColor(VGA_BLUE);

    while(true)
    {
        String mensaje = Visual_readString();
        if(mensaje == "Fin") break;
        String id_objeto = Visual_readString();
        if(k<6)
        {
            myGLCD.print(mensaje,pos_x,pos_y+45*k);
            btn_item[k]=myButtons.addButton(pos_x+btn_x,pos_y-5+45*k,30,20,lb[k]);
        }
        lista_objetos[k]=mensaje;
        lista_id_objetos[k]=id_objeto;
        num_lineas_texto++;
        k++;
    }
    int btn_devolver = myButtons.addButton(600, 420, 150, 30,"DEVOLVER");
    int btn_atras = myButtons.addButton(400, 420, 150, 30,"ATRAS");
    int arriba=myButtons.addButton(20,100,40,40,"<=");
    int abajo=myButtons.addButton(20,420,40,40,">=");

```

```

num_elementos=k-6;
if(k==0) myButtons.disableButton(btn_devolver);
k=0;
myButtons.disableButton(arriba);
if(num_elementos<=0) myButtons.disableButton(abajo);
myButtons.drawButtons();
while(true)
{
    if(myTouch.dataAvailable() == true)
    {
        do{
            btn_presionado = myButtons.checkButtons();
        }while(!(btn_presionado != -1));
        for(int m=0;m<num_lineas_texto && m<6;m++)
        {
            if(btn_presionado == btn_item[m])
            {
                if(lb[m]=="-")
                {
                    lb[m]="X";
                    validar_devoluciones[m]=true;
                }else{
                    lb[m]="-";
                    validar_devoluciones[m]=false;
                }
                myButtons.relabelButton(btn_item[m],lb[m],true);
            }
        }
        if(btn_presionado == btn_devolver)
        {
            for(int mtu=0;mtu<num_lineas_texto && mtu<6;mtu++)
            {
                if(validar_devoluciones[mtu]==true)
                {
                    myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
                    myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA DE
COLOR AMARILLO.
                    myGLCD.setColor(VGA_GREEN);
                    myGLCD.setBackColor(valor_color);
                    myGLCD.print("PASE LA TARJETA DEL EQUIPO",CENTER,200);

```

```

myGLCD.print(lista_objetos[mtu],CENTER,250);
myGLCD.print("POR EL LECTOR RFID",CENTER,300);
myGLCD.print("ELECTRONICA UNPRG",CENTER,350);
Visual_print("CLAVE R F I D");
String id_objeto_laboratorio;
id_objeto_laboratorio = lista_id_objetos[mtu+valor];
Visual_print(id_objeto_laboratorio);
OBTENER_CLAVE_RFID();
int jkl=0;
tuv;;
if(lector_RFID()==true)
{
    Visual_print("ELIMINAR OBJETO DE LA LISTA");
    Visual_print(id_objeto_laboratorio);
    myGLCD.print("OBJETO DEVUELTO",CENTER,400);
    delay(1000);
}
else{
    myGLCD.print("OBJETO NO RECONOCIDO",CENTER,400);
    delay(1000);
    myGLCD.print("          ",CENTER,400);
    jkl++;
    if(jkl!=3) goto tuv;
}
}
}
OBTENER_LISTA_PENDIENTES();
}
if(btn_presionado == btn_atras)
{
    menu_pedidos_devoluciones();
}
if(btn_presionado==arriba) { valor--;flechas();}
if(btn_presionado==abajo) { valor++;flechas();}
if(valor==0) { myButtons.disableButton(arriba);myButtons.drawButton(arriba);} else
{ myButtons.enableButton(arriba);myButtons.drawButton(arriba);}
    if(valor==num_elementos)
{ myButtons.disableButton(abajo);myButtons.drawButton(abajo);} else
{ myButtons.enableButton(abajo);myButtons.drawButton(abajo);}
}
}

```

```

    }
    void OBTENER_LISTA_MATERIALES()
    {
        myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
        myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA DE COLOR
AMARILLO.
        Visual_print("LISTA MATERIALES");
        while(!(Visual_readString()=="LISTA MATERIALES"));
        String msje1 = Visual_readString();
        if(msje1!="TIEMPO ACEPTABLE")
        {
            cabecera();
            myGLCD.print("SE HA VENCIDO EL PERIODO DE TIEMPO DE 12
HORAS",CENTER,200);
            myGLCD.print("POR FAVOR DEVUELVA LOS OBJETOS
PEDIDOS",CENTER,250);
            myGLCD.print("PARA QUE PUEDA HACER UN NUEVO
PEDIDO",CENTER,300);
            int boton_atras = myButtons.addButton( 500, 420, 150, 30,"ATRAS");
            myButtons.drawButtons();
            while(true)
            {
                if(myTouch.dataAvailable() == true)
                {
                    int boton_presionado = myButtons.checkButtons();
                    if(boton_presionado==boton_atras) {myButtons.deleteAllButtons();
menu_pedidos_devoluciones();}
                }
            }
            valor=0;
            k=0;
            num_elementos=0;
            cabecera();
            char* lb[6]={"-","-","-","-","-","-"};
            int btn_item[6];
            boolean validar_pedidos[6]={ false,false,false,false,false,false};
            const int pos_x=100;
            const int pos_y=140;
            const int btn_x=525;

```

```

int btn_presionado;
num_lineas_texto=0;
myGLCD.setColor(VGA_GREEN);
myGLCD.print("NOMBRES:",pos_x,pos_y-45);
myGLCD.print("UNIDADES:",pos_x+375,pos_y-45);
myGLCD.setColor(VGA_BLUE);
while(true)
{
    String mensaje = Visual_readString();
    if(mensaje == "Fin") break;
    String Cantidades = Visual_readString();
    if(k<6)
    {
        myGLCD.print(mensaje,pos_x,pos_y+45*k);
        myGLCD.print(Cantidades,pos_x+425,pos_y+45*k);
        btn_item[k]=myButtons.addButton(pos_x+btn_x,pos_y-5+45*k,30,20,lb[k]);
    }
    lista_objetos[k]=mensaje;
    num_lineas_texto++;
    k++;
}
int btn_pedir = myButtons.addButton(600, 420, 150, 30,"PEDIR");
int btn_otros = myButtons.addButton(400, 420, 150, 30,"OTROS");
int btn_atras = myButtons.addButton(200, 420, 150, 30,"ATRAS");
int arriba=myButtons.addButton(20,100,40,40,"<=");
int abajo=myButtons.addButton(20,420,40,40,">");
num_elementos=k-6;
if(k==0) myButtons.disableButton(btn_pedir);
k=0;
myButtons.disableButton(arriba);
if(num_elementos<=0) myButtons.disableButton(abajo);
myButtons.drawButtons();
while(true)
{
    if(myTouch.dataAvailable() == true)
    {
        do{
            btn_presionado = myButtons.checkButtons();
        }while(!(btn_presionado != -1));
    }
}

```

```

for(int m=0;m<num_lineas_texto && m<6;m++)
{
  if(btn_presionado == btn_item[m])
  {
    if(lb[m]=="-")
    {
      lb[m]="X";
      validar_pedidos[m]=true;
    }else{
      lb[m]="-";
      validar_pedidos[m]=false;
    }
    myButtons.relabelButton(btn_item[m],lb[m],true);
  }
  if(btn_presionado == btn_pedir)
  {
    for(int mtu=0;mtu<num_lineas_texto && mtu<6;mtu++)
    {
      if(validar_pedidos[mtu]==true)
      {
        myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
        myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA DE
COLOR AMARILLO.
        myGLCD.setColor(VGA_RED);
        myGLCD.setBackColor(valor_color);
        String objeto_laboratorio = lista_objetos[mtu+valor];
        myGLCD.print("PASE LA TARJETA RFID DE:",100,100);
        myGLCD.print(objeto_laboratorio,100,150);
        Visual_print("AGREGAR OBJETO A LA LISTA");
        Visual_print(objeto_laboratorio);
        abcd: obtener_id_targeta_rfid();
        String verificar = Visual_readString();
        if(verificar=="INCORRECTO")
        {
          myGLCD.setColor(VGA_RED);
          myGLCD.setBackColor(valor_color);
          myGLCD.print(" TARJETA INCORRECTA ",100,200); //200,380
          delay(500);
          myGLCD.print("          ",100,200);
          goto abcd;
        }
      }
    }
  }
}

```

```

        }else{
            myGLCD.setColor(VGA_RED);
            myGLCD.setBackColor(valor_color);
            myGLCD.print(" TARJETA CORRECTA ",100,200);
            Visual_print("SONIDO2");
            delay(500);
            myGLCD.print("          ",100,200);
        }
    }
    }
    OBTENER_LISTA_MATERIALES();
}
}
if(btn_presionado == btn_atras)
{
    menu_pedidos_devoluciones();
}
if(btn_presionado == btn_otros)
{
    String equipo=ingresa("EQUIPO:", "");
    Visual_print("OTRO EQUIPO");
    Visual_print(equipo);
    OBTENER_LISTA_MATERIALES();
}
if(btn_presionado==arriba) { valor--;flechas();}
if(btn_presionado==abajo) { valor++;flechas();}
if(valor==0) { myButtons.disableButton(arriba);myButtons.drawButton(arriba);} else
{ myButtons.enableButton(arriba);myButtons.drawButton(arriba);}
    if(valor==num_elementos)
{ myButtons.disableButton(abajo);myButtons.drawButton(abajo);} else
{ myButtons.enableButton(abajo);myButtons.drawButton(abajo);}
}
}
}
void flechas()
{
    int pos_x=100;
    int pos_y=100;
    for(int gtu=0; gtu<6 ; gtu++)
    {

```

```

myGLCD.setBackColor(VGA_BLACK);
myGLCD.print("                ",pos_x,pos_y+50*gtu);
}
for(int gtu=0; gtu<6; gtu++)
{
    if(k<6)
    {
        if(gtu>k-1) goto salir;
    }
    myGLCD.print(lista_objetos[gtu+valor],pos_x,pos_y+50*gtu);
}
salir;;
}
void identificacion()
{
    if(Visual_readString() == "CODIGO USUARIO")
    {
        reconocimiento = Visual_readString();
        if(reconocimiento == "USUARIO ENCONTRADO")
        {
            myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
            myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA
DE COLOR AMARILLO.
            myGLCD.setColor(VGA_RED);
            myGLCD.setBackColor(VGA_TRANSPARENT);

            if(habilitar_sensor_huella=="TRUE")
            {
                myGLCD.print("Por favor pon tu huella dactilar",CENTER,200);
                myGLCD.print("... Identificando Usuario de Destino ...",CENTER,250);
                myGLCD.print("ELECTRONICA UNPRG",CENTER,300);
                HUELLA.OPEN(Serial);
                HUELLA.CmosLed(1);
                ID = Visual_readString().toInt();
                int id_sensor = HUELLA.IdentificarHuella();
                HUELLA.CmosLed(0);
                HUELLA.CLOSE();
                myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
                myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA
DE COLOR AMARILLO.

```



```

myGLCD.setBackColor(valor_color);
myGLCD.setColor(VGA_WHITE);
if(id_sensor != -1)
{
    if(ID == id_sensor)
    {
        Visual_print("USUARIO IDENTIFICADO");
        tren_pulsos_cuadrados(led_SENSOR_HUELLA);
        myGLCD.print("Transferencia Exitosa", 400, 200);
        Visual_print("SONIDO3");
        delay(2000);
    }else{
        Visual_print("USUARIO NO IDENTIFICADO, ACCESO
DENEGADO");

        myGLCD.print("Transferencia No Exitosa", 400, 200);
        delay(2000);
    }
}else{
    Visual_print("USUARIO NO IDENTIFICADO, ACCESO
DENEGADO");

    myGLCD.print("Transferencia No Exitosa", 400, 200);
    delay(2000);
}
}else{
    Visual_print("USUARIO IDENTIFICADO");
    tren_pulsos_cuadrados(led_SENSOR_HUELLA);
    myGLCD.print("Transferencia Exitosa", 400, 200);
    Visual_print("SONIDO3");
    delay(2000);
}
}
}

```

➤ **Programación usada para el control del teclado de la pantalla.**

```

int btn_caja_texto;
void caja_texto(char* mostrar_texto)
{
    myButtons.relabelButton(btn_caja_texto,mostrar_texto);
myButtons.setButtonColors(VGA_BLUE,VGA_BLUE,VGA_MAROON,VGA_BLUE,VGA_W
HITE); //ESTABLECE EL COLOR BLANCO AL BOTON
    myButtons.drawButton(btn_caja_texto);
}
String letra_m="ABC";
String teclado(String mostrar)
{
char*mayusculas[]={ "1","2","3","4","5","6","7","8","9","0","Q","W","E","R","T","Y","U","I","
O","P","A","S","D","F","G","H","J","K","L","~","Z","X","C","V","B","N","M"};
char*minusculas[]={ "1","2","3","4","5","6","7","8","9","0","q","w","e","r","t","y","u","i","o","p"
,"a","s","d","f","g","h","j","k","l","~","z","x","c","v","b","n","m"};
    #define cant_teclas 37 //Definimos la cantidad de teclas a utilizar para las letras y
números.
    char *caracteres[cant_teclas];
    for(int posicion=0;posicion<cant_teclas;posicion++)
    {
        caracteres[posicion] = mayusculas[posicion];
    }
    #define cant 10 //Definimos la cantidad de teclas por fila.
    int btn_teclas[cant_teclas+4],btn_anterior=-2,btn_presionado,n=0;
    String mensaje;
    mensaje = mostrar+"";
    for(int i=0;i<cant;i++) //Agremos todos los botones con los números y letras.
    {
        btn_teclas[i]=myButtons.addButton(30+50*i,150,40,40,caracteres[i]);
        btn_teclas[i+cant]=myButtons.addButton(30+50*i,200,40,40,caracteres[i+cant]);
        btn_teclas[i+2*cant]=myButtons.addButton(30+50*i,250,40,40,caracteres[i+2*cant]);
        if(i<7)
        {
            btn_teclas[i+3*cant]=myButtons.addButton(30+50*i,300,40,40,caracteres[i+3*cant]);
        }
    }
}

```



```

        char mensajeria[mensaje.length()+1];
        mensaje.toCharArray(mensajeria,mensaje.length()+1);
        caja_texto(mensajeria);
    }
}
if(btn_presionado == btn_teclas[cant_teclas+1]) // TECLA CON LA FIGURA DE LA
TIJERA
{
    if(btn_anterior != -2)
    {
myButtons.setButtonColors(VGA_WHITE,VGA_GRAY,VGA_WHITE,VGA_BLUE,VGA_RE
D); //ESTABLECE EL COLOR ROJO AL BOTON
myButtons.drawButton(btn_anterior); //Dibuja al Boton Seleccionado.
    }
myButtons.setButtonColors(VGA_WHITE,VGA_GRAY,VGA_WHITE,VGA_BLUE,VGA_GR
EEN); //ESTABLECE EL COLOR VERDE AL BOTON
myButtons.drawButton(btn_teclas[cant_teclas+1]); //Dibuja al Boton Seleccionado.
    mensaje.remove(mensaje.length()-1); //Borra un caracter del mensaje.
    char mensajeria[mensaje.length()+1];
    mensaje.toCharArray(mensajeria,mensaje.length()+1);
    caja_texto(mensajeria);
}
if(btn_presionado == btn_teclas[cant_teclas]) //TECLA CON LA FIGURA DEL
LÁPIZ. (SIGUIENTE)
{
myButtons.setButtonColors(VGA_WHITE,VGA_GRAY,VGA_WHITE,VGA_BLUE,VGA_GR
EEN); //ESTABLECE EL COLOR VERDE AL BOTON
myButtons.drawButton(btn_teclas[cant_teclas]); //Dibuja al Boton Seleccionado.
    break;
}
if(btn_presionado == btn_teclas[cant_teclas+2]) //TECLA PARA LAS
MAYÚSCULAS Y MINÚSCULAS
{
    if(btn_anterior != -2)
    {
myButtons.setButtonColors(VGA_WHITE,VGA_GRAY,VGA_WHITE,VGA_BLUE,VGA_RE
D); //ESTABLECE EL COLOR ROJO AL BOTON
myButtons.drawButton(btn_anterior); //Dibuja al Boton Seleccionado.
    }
myButtons.setButtonColors(VGA_WHITE,VGA_GRAY,VGA_WHITE,VGA_BLUE,VGA_RE

```

```

D); //ESTABLECE EL COLOR ROJO AL BOTON
//myButtons.drawButton(btn_teclas[cant_teclas+2]); //Dibuja al Boton Seleccionado.
    if(letra_m=="abc")
    {
        letra_m="ABC";
        myButtons.relabelButton(btn_teclas[cant_teclas+2],"ABC");
        myButtons.drawButton(btn_teclas[cant_teclas+2]);
        for(int posicion=0;posicion<cant_teclas;posicion++)
        {
            caracteres[posicion] = mayusculas[posicion];
            myButtons.relabelButton(btn_teclas[posicion],mayusculas[posicion]);
            myButtons.drawButton(btn_teclas[posicion]);
        }
    }else{
        letra_m="abc";
        myButtons.relabelButton(btn_teclas[cant_teclas+2],"abc");
        myButtons.drawButton(btn_teclas[cant_teclas+2]);
        for(int posicion=0;posicion<cant_teclas;posicion++)
        {
            caracteres[posicion] = minusculas[posicion];
            myButtons.relabelButton(btn_teclas[posicion],minusculas[posicion]);
            myButtons.drawButton(btn_teclas[posicion]);
        }
    }
}

if(btn_presionado == btn_teclas[cant_teclas+3]) //Tecla con el Botón Espacio.
{
    if(btn_anterior != -2)
    {
myButtons.setButtonColors(VGA_WHITE,VGA_GRAY,VGA_WHITE,VGA_BLUE,VGA_RE
D); //ESTABLECE EL COLOR ROJO AL BOTON
myButtons.drawButton(btn_anterior); //Dibuja al Boton Seleccionado.
    }
myButtons.setButtonColors(VGA_WHITE,VGA_GRAY,VGA_WHITE,VGA_BLUE,VGA_GR
EEN); //ESTABLECE EL COLOR VERDE AL BOTON
myButtons.drawButton(btn_teclas[cant_teclas+3]); //Dibuja al Boton Seleccionado.
    mensaje = mensaje + " ";
    char mensajeria[mensaje.length()+1];
    mensaje.toCharArray(mensajeria,mensaje.length()+1);

```

```

        caja_texto(mensajeria);
    }
    if(btn_presionado == btn_teclas[cant_teclas+4])
    {
        secuencia=secuencia-2;
        break;
    }

    btn_anterior = btn_presionado;

}
}
myButtons.deleteAllButtons();
return mensaje;
}

String ingresa(char* poner_info,String mostrar)
{
    myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
    myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA DE COLOR
AMARILLO.
    encabezado(800-200-40,150);
    myGLCD.setFont(BigFont);
    myGLCD.setBackgroundColor(0, 0, 255);
    int btn_cod=myButtons.addButton(50,68,200,30,poner_info,BUTTON_DISABLED);
    myButtons.drawButton(btn_cod);
    String texto_informacion = teclado(mostrar);
    return texto_informacion;
}

```

➤ **Programación usada para el control del NFC/RFID.**

```
#include <Wire.h>
int datosRFID[4];
int claveRFID[4]; // CÓDIGOS DE LA TARJETA RFID O TAG AUTORIZADO
byte ACK[]={ 1,0x00,0x00,0xFF,0x00,0xFF,0x00};
byte SECUENCIA[]={ 1,0x00,0x00,0xFF};
byte DATA1[20];
const int IRQ = A8;
boolean lector_RFID()
{
    unsigned int conteo = 0;
    boolean retorno = false;
    verificar_conexion_RFID(Serial);
    DATA1[0] = 0x14; // SAMConfiguration
    DATA1[1] = 0x01; // Normal Mode
    DATA1[2] = 0x01; // Timeout 50ms * DATA1[2] = 50 ms.
    DATA1[3] = 0x01; // IRQ Activado
    enviar_data(DATA1,4);
    while(digitalRead(IRQ));
    recibir_data(0x03,1,DATA1);
    while(true)
    {
        DATA1[0] = 0x4A;
        DATA1[1] = 0x01;
        DATA1[2] = 0x00;
        enviar_data(DATA1,3);
        while(digitalRead(IRQ));
        if(recibir_data(0x03,11,DATA1)==1)
        {
            datosRFID[0]=DATA1[6];
            datosRFID[1]=DATA1[7];
            datosRFID[2]=DATA1[8];
            datosRFID[3]=DATA1[9];
            ///// comparamos todos los datos para dar permiso de acceso.
            if(comparando_clave()) // comparamos los datos de los dos vectores.
            {
                Serial.println("LOS CODIGOS HAN SIDO VERIFICADOS");
                Visual_print("LOS CODIGOS HAN SIDO VERIFICADOS");
            }
        }
    }
}
```

```

    Serial.println("OBJETO RECONOCIDO"); // IMPRIME MENSAJE.
    Visual_print("OBJETO RECONOCIDO");
    tren_pulsos_cuadrados(led_RFID);
    datosRFID[4] = 0; //COLOCA A CERO TODA LA MATRIZ;
    retorno = true;
    break;
} else {
    Serial.println("TARJETA INCORRECTA, ACCESO DENEGADO");
    Visual_print("TARJETA INCORRECTA, ACCESO DENEGADO");
    datosRFID[4] = 0; //COLOCA A CERO TODA LA MATRIZ;
    conteo++;
    if(conteo==5) {retorno=false; break;}
    }
}
delay(1000);
}
return retorno;
}

void verificar_conexion_RFID(HardwareSerial &comunicacion)
{
    Wire.begin();
    pinMode(IRQ,INPUT);
    nuevo:DATA1[0] = 0x02;
    enviar_data(DATA1,1);
    while(digitalRead(IRQ));
    boolean htu=recibir_data(0x03,5,DATA1);
    if(htu == 0) goto nuevo;
    comunicacion.print("Se encontro el Circuito Integrado
PN5");comunicacion.println(DATA1[0],HEX);delay(100);
    comunicacion.print("Version del Firmware: ");
comunicacion.print(DATA1[1],HEX);comunicacion.print('.');comunicacion.println(DATA1[2],H
EX);delay(100);
    comunicacion.print("Support: ");comunicacion.println(DATA1[3]);delay(100);
}

void enviar_data(byte packet_data[],int LEN)
{
    LEN++;
    Wire.beginTransaction(0x24);
    Wire.write(0x00); //PREAMBLE
    Wire.write(0x00); // START CODE 1

```



```

Wire.write(0xFF); // START CODE 2
Wire.write(LEN); //LEN
Wire.write(~LEN+1); //LCS
Wire.write(0xD4); //TFI
int checksum = 0xD4;
for(int i=0;i<LEN-1;i++)
{
    Wire.write(packet_data[i]);
    checksum+=packet_data[i];
}
Wire.write(~checksum+1); //DCS
Wire.write(0x00); //POSTAMBLE
Wire.endTransmission();
}
boolean recibir_data(byte command_code,int tam,byte VECTOR_DATA[])
{
    guardar_data(DATA1,7);
    boolean htu=comparar_vectores(ACK,DATA1,0,7);
    if(htu!=true) return 0;
    while(digitalRead(IRQ));
    tam+=9;
    guardar_data(DATA1,tam);
    htu=comparar_vectores(SECUENCIA,DATA1,0,4);
    if(htu!=true) return 0;
    byte LEN = DATA1[4]; byte LCS = DATA1[5];
    byte suma = LEN+LCS;
    if(suma!=0x00) return 0;
    for(int m=6;m<tam-1;m++)
    {
        suma+=DATA1[m];
    }
    if(suma!=0x00) return 0;
    for(int m=8;m<tam-2;m++)
    {
        VECTOR_DATA[m-8]=DATA1[m];
    }
    return 1;
}
void guardar_data(byte ABC[],int num)
{

```

```

Wire.requestFrom(0x24,num);
int m=0;
while(Wire.available())
{
    byte ct = Wire.read();
    ABC[m]=ct;
    m++;
}
}
int comparando_clave()
{
    for(int m=0; m<4; m++)
    {
        if(datosRFID[m]!=claveRFID[m]) return false;
    }
    return true;
}
void OBTENER_CLAVE_RFID()
{
    while(true)
    {
        if(VISUAL_BASIC.available()>0)
        {
            String msje=VISUAL_BASIC.readStringUntil('\n');
            msje.remove(msje.length()-1);
            Serial.println(msje);
            if(msje == "CLAVE R F I D")
            {
                int k=0;
                while(true)
                {
                    if(VISUAL_BASIC.available()>0)
                    {
                        String msje=VISUAL_BASIC.readStringUntil('\n');
                        msje.remove(msje.length()-1);
                        claveRFID[k]=msje.toInt(); //Convertimos el mensaje de String a Int y lo
guardamos en el vector claveRFID.
                        Serial.println(claveRFID[k]);
                        k++;
                        if(k==4) break;

```

```

        }
    }
    break;
}
}
}
}
void obtener_id_targeta_rfid()
{
    Wire.begin();
    pinMode(IRQ,INPUT);
    DATA1[0] = 0x14; // SAMConfiguration
    DATA1[1] = 0x01; // Normal Mode
    DATA1[2] = 0x01; // Timeout 50ms * DATA1[2] = 50 ms.
    DATA1[3] = 0x01; // IRQ Activado
    enviar_data(DATA1,4);
    while(digitalRead(IRQ));
    recibir_data(0x03,1,DATA1);
    while(true)
    {
        DATA1[0] = 0x4A;
        DATA1[1] = 0x01;
        DATA1[2] = 0x00;
        enviar_data(DATA1,3);
        while(digitalRead(IRQ));
        boolean htu = recibir_data(0x03,11,DATA1);
        if(htu)
        {
            VISUAL_BASIC.println(DATA1[6]);
            delay(100);
            VISUAL_BASIC.println(DATA1[7]);
            delay(100);
            VISUAL_BASIC.println(DATA1[8]);
            delay(100);
            VISUAL_BASIC.println(DATA1[9]);
            delay(100);
            break;
        }
    }
}
}

```

➤ **Programación usada para el control del sensor de huella GT-511C3.**

```

class SENSOR_HUELLA_SPARKFUN
{
    private:
        byte PARAMETRO[4];
        byte
RESPONSE_PACKET[12]={0x55,0xAA,0x01,0x00,0x00,0x00,0x00,0x00,0x30,0x00,0x30,0x0
1};

        byte IDENTIFICADOR[4]={0x5A,0xA5,0x01,0x00};
        byte DATA2[14];
        int posH_y=60;
        int gh=1;
    public:
        void OPEN(HardwareSerial &comunicacion)
        {
            PARAMETRO[0]=1;
            PARAMETRO[1]=0;
            PARAMETRO[2]=0;
            PARAMETRO[3]=0;
            enviar_comando(PARAMETRO,0x01);// Command: Open
            guardar_data_serial(DATA2,12);
            boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
            if(htu == true)
            {
                guardar_data_serial(DATA2,4);
                boolean htu2 = comparar_vectores(IDENTIFICADOR,DATA2,0,4);
                if(htu2 == true)
                {
                    byte FirmwareVersion[4];
                    guardar_data_serial(FirmwareVersion,4);
                    byte IsoAreaMaxSize[4];
                    guardar_data_serial(IsoAreaMaxSize,4);
                    byte DeviceSerialNumber[16];
                    guardar_data_serial(DeviceSerialNumber,16);
                    unsigned int checkSum =
sum(IDENTIFICADOR,4)+sum(FirmwareVersion,4)+sum(IsoAreaMaxSize,4)+sum(DeviceSeri
alNumber,16);
                    byte check[2];

```

```

    guardar_data_serial(check,2);
    if(lowByte(checkSum)==check[0] && highByte(checkSum)==check[1])
    {
        comunicacion.print("Firmware Version: ");
    imprimir_vector(FirmwareVersion,4,comunicacion);delay(100);
        comunicacion.print("Maximum size of ISO CD image: ");
    imprimir_vector(IsoAreaMaxSize,4,comunicacion);delay(100);
        comunicacion.print("Unique serial number: ");
    imprimir_vector(DeviceSerialNumber,16,comunicacion);delay(100);
    }
}
}
}
void CLOSE()
{
    PARAMETRO[0]=0;
    PARAMETRO[1]=0;
    PARAMETRO[2]=0;
    PARAMETRO[3]=0;
    enviar_comando(PARAMETRO,0x02);// Command: Close
    guardar_data_serial(DATA2,12);
    boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
    if(htu==true) Serial.print("SENSOR DE HUELLA DESACTIVADO");
}
void CmosLed(boolean estado)
{
    PARAMETRO[0]=estado;
    PARAMETRO[1]=0;
    PARAMETRO[2]=0;
    PARAMETRO[3]=0;
    enviar_comando(PARAMETRO,0x12);// Command: CmosLed
    guardar_data_serial(DATA2,12);
    boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
    //if(htu==true) {Serial.print("LED: "); Serial.println(estado);}
}
void DELETEID(int ID)
{
    PARAMETRO[0]=ID;
    PARAMETRO[1]=0;
    PARAMETRO[2]=0;

```

```

PARAMETRO[3]=0;
enviar_comando(PARAMETRO,0x40);// Command: DeleteID
guardar_data_serial(DATA2,12);
boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
}
void DELETEALL(HardwareSerial &comunicacion)
{
PARAMETRO[0]=0;
PARAMETRO[1]=0;
PARAMETRO[2]=0;
PARAMETRO[3]=0;
enviar_comando(PARAMETRO,0x41);// Command: DeleteAll
guardar_data_serial(DATA2,12);
boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
if(htu==true)
{
comunicacion.println("BASE DE DATOS DE HUELLAS BORRADA");
}else{
comunicacion.println("NO SE PUDO BORRAR BASE DE DATOS DE
HUELLAS");
}
}
boolean ISPRESSFINGER()
{
PARAMETRO[0]=0;
PARAMETRO[1]=0;
PARAMETRO[2]=0;
PARAMETRO[3]=0;
enviar_comando(PARAMETRO,0x26);// Command: IsPressFinger
guardar_data_serial(DATA2,12);
boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
return htu;
}
void GetRawImage()
{
boolean htu = false;
PARAMETRO[0]=0;
PARAMETRO[1]=0;
PARAMETRO[2]=0;
PARAMETRO[3]=0;

```

```

    enviar_comando(PARAMETRO,0x63);// Command: GetRawImage
    guardar_data_serial(DATA2,12);
    htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
    guardar_data_serial(DATA2,4);
}

void EnrollID(int ID,HardwareSerial &comunicacion, boolean estado)
{
    if(EnrollStart(ID)==true)
    {
        comunicacion.println("Presione el dedo para empezar a Enrolar #"); delay(100);
        while(!ISPRESSFINGER()); delay(100);
        if(CaptureFinger(1)==true)
        {
            if(estado==1) myGLCD.print("      *Vez #1: OK", 30, posH_y+(60*gh)+30);
            if(Enroll1()==true)
            {
                comunicacion.println("Remueva su dedo"); delay(100);
                if(estado==1) presione_dedo(1);
                while(ISPRESSFINGER()); delay(100);
                comunicacion.println("Presione el dedo de nuevo"); delay(100);
                if(estado==1) presione_dedo(0);
                while(!ISPRESSFINGER()); delay(100);
                if(CaptureFinger(1)==true)
                {
                    if(estado==1) myGLCD.print("      *Vez #2: OK", 30, posH_y+(60*gh)+60);
                    if(Enroll2()==true)
                    {
                        comunicacion.println("Remueva su dedo"); delay(100);
                        if(estado==1) presione_dedo(1);
                        while(ISPRESSFINGER()); delay(100);
                        comunicacion.println("Presione el dedo por ultima vez"); delay(100);
                        if(estado==1) presione_dedo(0);
                        while(!ISPRESSFINGER()); delay(100);
                        if(CaptureFinger(1)==true)
                        {
                            if(estado==1) myGLCD.print("      *Vez #3: OK", 30,
posH_y+(60*gh)+90);
                            if(Enroll3()==true)
                            {

```

```

        comunicacion.println("Remueva su dedo"); delay(100);
        if(estado==1) presione_dedo(1);
        while(ISPRESSFINGER()); delay(100);
        comunicacion.println("La Enrolacion ha sucedido"); delay(100);
        if(estado==1)
        {
            myGLCD.setColor(0, 255, 0);
            myGLCD.print("!Huella Registrada!", 440, 420);
            myGLCD.setColor(255, 255, 255);
        }
    }
}
}
}
}
}else{
    comunicacion.println("No se puede grabar huella"); delay(100);
}
}

int Identify()
{
    int ID=-1;
    PARAMETRO[0]=0; // Si estado = 1 entonces Captura Imagen de Alta Resolucion. Si
estado = 0 entonces Captura Imagen de Baja Resolucion
    PARAMETRO[1]=0;
    PARAMETRO[2]=0;
    PARAMETRO[3]=0;
    enviar_comando(PARAMETRO,0x51);// Command: Identify
    guardar_data_serial(DATA2,12);
    boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,4);
    if(htu==true)
    {
        if(DATA2[8]==0x30 && DATA2[9]==0x00)
        {
            unsigned int checkSum = sum(DATA2,10);
            if(lowByte(checkSum)==DATA2[10] && highByte(checkSum)==DATA2[11])
            {
                ID=DATA2[4];

```



```

    }
  }
}
return ID;
}

int IdentificarHuella()
{
  int ID;
  int retorno=-1;
  int mnp=0;
  while(!ISPRESSFINGER()); delay(100);
  while(true)
  {
    if(CaptureFinger(1)==true)
    {
      ID=Identify();
      if(ID<200 && ID!=-1)
      {
        retorno=ID;
        break;
      } else
      {
        mnp++;
        if(mnp==3) break;
      }
    }
  }
  return retorno;
}

void registrar_huella(int id)
{
  myButtons.deleteAllButtons(); //Elimina todos los botones existentes.
  myGLCD.fillScr(valor_color); //PONE EL FONDO DE LA PANTALLA DE COLOR
AMARILLO.
  encabezado(800-200-40,90);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(255, 0, 0);
  myGLCD.setBackgroundColor(VGA_BLACK);
  myGLCD.print("*ACERQUESE AL SENSOR DE HUELLA", 30, posH_y+15);

```

```

    myGLCD.setColor(VGA_BLUE);
    DELETEID(id);
    EnrollID(id,Serial,1);
}
void presione_dedo(boolean t)
{
    if(t==0)
    {
        myGLCD.setColor(255, 0, 0);
        myGLCD.print("          ", 440, 320);
        myGLCD.print("          ", 440, 350);
        myGLCD.print("!Presione el mismo", 440, 320);
        myGLCD.print("dedo!", 440, 350);
        myGLCD.setColor(VGA_BLUE);
    }else
    {
        myGLCD.setColor(255, 0, 0);
        myGLCD.print("          ", 440, 320);
        myGLCD.print("          ", 440, 350);
        myGLCD.print("!Remueva su dedo!", 440, 320);
        myGLCD.setColor(VGA_BLUE);
    }
}
boolean CaptureFinger(boolean estado)
{
    PARAMETRO[0]=estado; // Si estado = 1 entonces Captura Imagen de Alta
Resolucion. Si estado = 0 entonces Captura Imagen de Baja Resolucion
    PARAMETRO[1]=0;
    PARAMETRO[2]=0;
    PARAMETRO[3]=0;
    enviar_comando(PARAMETRO,0x60);// Command: CaptureFinger
    guardar_data_serial(DATA2,12);
    boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
    return htu;
}
boolean EnrollStart(int ID)
{
    PARAMETRO[0]=ID;
    PARAMETRO[1]=0;
    PARAMETRO[2]=0;

```

```

PARAMETRO[3]=0;
enviar_comando(PARAMETRO,0x22);// Command: EnrollStart
guardar_data_serial(DATA2,12);
boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
return htu;
}
boolean Enroll1()
{
PARAMETRO[0]=0;
PARAMETRO[1]=0;
PARAMETRO[2]=0;
PARAMETRO[3]=0;
enviar_comando(PARAMETRO,0x23);// Command: Enroll1
guardar_data_serial(DATA2,12);
boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
return htu;
}
boolean Enroll2()
{
PARAMETRO[0]=0;
PARAMETRO[1]=0;
PARAMETRO[2]=0;
PARAMETRO[3]=0;
enviar_comando(PARAMETRO,0x24);// Command: Enroll2
guardar_data_serial(DATA2,12);
boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
return htu;
}
boolean Enroll3()
{
PARAMETRO[0]=0;
PARAMETRO[1]=0;
PARAMETRO[2]=0;
PARAMETRO[3]=0;
enviar_comando(PARAMETRO,0x25);// Command: Enroll3
guardar_data_serial(DATA2,12);
boolean htu = comparar_vectores(RESPONSE_PACKET,DATA2,0,12);
return htu;
}
void enviar_comando(byte Parameter[4],byte Command)

```

```

{
    SENSOR_HUELLA.write(0x55); //Command start code1
    SENSOR_HUELLA.write(0xAA); //Command start code2
    SENSOR_HUELLA.write(0x01); // Device ID LSB
    SENSOR_HUELLA.write(0x00); // Device ID MSB
    SENSOR_HUELLA.write(Parameter[0]); // Parameter 0
    SENSOR_HUELLA.write(Parameter[1]); // Parameter 1
    SENSOR_HUELLA.write(Parameter[2]); // Parameter 2
    SENSOR_HUELLA.write(Parameter[3]); // Parameter 3
    SENSOR_HUELLA.write(Command); // Command Code LSB
    SENSOR_HUELLA.write(0x00); // Command Code MSB
    unsigned int checkSum = 0x55 + 0xAA + 0x01 + 0x00 + Parameter[0] + Parameter[1]
+ Parameter[2] + Parameter[3] + Command + 0x00;
    SENSOR_HUELLA.write(lowByte(checkSum));
    SENSOR_HUELLA.write(highByte(checkSum));
}
void guardar_data_serial(byte ABC[],unsigned int num)
{
    int m=0;
    while(true)
    {
        if(SENSOR_HUELLA.available())
        {
            byte ct = SENSOR_HUELLA.read();
            ABC[m]=ct;
            m++;
            if(m==num) break;
        }
    }
}

unsigned int sum(byte vector[],unsigned int LEN)
{
    unsigned int suma=0;
    for(unsigned int m=0;m<LEN;m++)
    {
        suma=suma+vector[m];
    }
    return suma;
}

```

```
void imprimir_vector(byte vector[],int LEN,HardwareSerial &Comunicacion)
{
  for(int m=0;m<LEN;m++)
  {
    Comunicacion.print(vector[m],HEX);
  }
  Comunicacion.println("");
}
};
```

ANEXO 3

COMANDOS PARA LA PROGRAMACIÓN DEL SENSOR GT-511C3

2. Protocol: Packet Structure

(Multi-byte item is represented as Little Endian.)

Command Packet (Command)

OFFSET	ITEM	TYPE	DESCRIPTION
0	0x55	BYTE	Command start code1
1	0xAA	BYTE	Command start code2
2	<i>Device ID</i>	WORD	Device ID: default is 0x0001, always fixed
4	<i>Parameter</i>	DWORD	Input parameter
8	<i>Command</i>	WORD	Command code
10	<i>Check Sum</i>	WORD	Check Sum (byte addition) OFFSET[0]+...+OFFSET[9]=Check Sum

Response Packet (Acknowledge)

OFFSET	ITEM	TYPE	DESCRIPTION
0	0x55	BYTE	Response start code1
1	0xAA	BYTE	Response start code2
2	<i>Device ID</i>	WORD	Device ID: default is 0x0001, always fixed
4	<i>Parameter</i>	DWORD	Response == 0x30: (ACK) Output Parameter Response == 0x31: (NACK) Error code
8	<i>Response</i>	WORD	0x30: Acknowledge (ACK). 0x31: Non-acknowledge (NACK).
10	<i>Check Sum</i>	WORD	Check Sum (byte addition) OFFSET[0]+...+OFFSET[9]=Check Sum

Data Packet (Data)

OFFSET	ITEM	TYPE	DESCRIPTION
0	0x5A	BYTE	Data start code1
1	0xA5	BYTE	Data start code2
2	<i>Device ID</i>	WORD	Device ID: default is 0x0001, always fixed
4	<i>Data</i>	N BYTES	N bytes Data The size is pre-defined per protocol stage
4+N	<i>Check Sum</i>	WORD	Check Sum (byte addition) OFFSET[0]+...+OFFSET[4+N-1]=Check Sum

3. Protocol: Commands Summary

In a command packet *Command* can be one of below.

Number (HEX)	Alias	Description
01	<i>Open</i>	Initialization
02	<i>Close</i>	Termination
03	<i>UsbInternalCheck</i>	Check if the connected USB device is valid
04	<i>ChangeBaudrate</i>	Change UART baud rate
05	<i>SetIAPMode</i>	Enter IAP Mode In this mode, FW Upgrade is available
12	<i>CmosLed</i>	Control CMOS LED
20	<i>GetEnrollCount</i>	Get enrolled fingerprint count
21	<i>CheckEnrolled</i>	Check whether the specified ID is already enrolled
22	<i>EnrollStart</i>	Start an enrollment
23	<i>Enroll1</i>	Make 1 st template for an enrollment
24	<i>Enroll2</i>	Make 2 nd template for an enrollment
25	<i>Enroll3</i>	Make 3 rd template for an enrollment, merge three templates into one template, save merged template to the database
26	<i>IsPressFinger</i>	Check if a finger is placed on the sensor
40	<i>DeleteID</i>	Delete the fingerprint with the specified ID
41	<i>DeleteAll</i>	Delete all fingerprints from the database
50	<i>Verify</i>	1:1 Verification of the capture fingerprint image with the specified ID
51	<i>Identify</i>	1:N Identification of the capture fingerprint image with the database
52	<i>VerifyTemplate</i>	1:1 Verification of a fingerprint template with the specified ID
53	<i>IdentifyTemplate</i>	1:N Identification of a fingerprint template with the database
60	<i>CaptureFinger</i>	Capture a fingerprint image(256x256) from the sensor
61	<i>MakeTemplate</i>	Make template for transmission

Number (HEX)	Alias	Description
62	<i>GetImage</i>	Download the captured fingerprint image(256x256)
63	<i>GetRawImage</i>	Capture & Download raw fingerprint image(320x240)
70	<i>GetTemplate</i>	Download the template of the specified ID
71	<i>SetTemplate</i>	Upload the template of the specified ID
72	<i>GetDatabaseStart</i>	Start database download, obsolete
73	<i>GetDatabaseEnd</i>	End database download, obsolete
80	<i>UpgradeFirmware</i>	Not supported
81	<i>UpgradeISOCImage</i>	Not supported
30	<i>Ack</i>	Acknowledge.
31	<i>Nack</i>	Non-acknowledge.

4. Protocol: Error Codes

When response packet is Non-acknowledge, *Parameter* represents an error code as below.

NACK Parameter	Value	Description
NACK_TIMEOUT	0x1001	Obsolete , capture timeout
NACK_INVALID_BAUDRATE	0x1002	Obsolete , Invalid serial baud rate
NACK_INVALID_POS	0x1003	The specified ID is not between 0~199
NACK_IS_NOT_USED	0x1004	The specified ID is not used
NACK_IS_ALREADY_USED	0x1005	The specified ID is already used
NACK_COMM_ERR	0x1006	Communication Error
NACK_VERIFY_FAILED	0x1007	1:1 Verification Failure
NACK_IDENTIFY_FAILED	0x1008	1:N Identification Failure
NACK_DB_IS_FULL	0x1009	The database is full
NACK_DB_IS_EMPTY	0x100A	The database is empty
NACK_TURN_ERR	0x100B	Obsolete , Invalid order of the enrollment (The order was not as: EnrollStart -> Enroll1 -> Enroll2 -> Enroll3)
NACK_BAD_FINGER	0x100C	Too bad fingerprint
NACK_ENROLL_FAILED	0x100D	Enrollment Failure
NACK_IS_NOT_SUPPORTED	0x100E	The specified command is not supported
NACK_DEV_ERR	0x100F	Device Error, especially if Crypto-Chip is trouble
NACK_CAPTURE_CANCELED	0x1010	Obsolete , The capturing is canceled
NACK_INVALID_PARAM	0x1011	Invalid parameter
NACK_FINGER_IS_NOT_PRESSED	0x1012	Finger is not pressed
Duplicated ID	0 - 199	There is duplicated fingerprint (while enrollment or setting template), This error describes just duplicated ID

ANEXO 4

COMANDOS PARA LA PROGRAMACION DEL CHIPSET 532 NFC/RFID

13.56MHz

7. Commands supported

The following description of the commands details:

- The frame structure⁵ including the type and amount of data:
 - That the host controller has to deliver to the PN532 (*Input*).
 - That the PN532 returns to the host controller (*Output*).
- When existing, the possible causes of syntax error (*Syntax Error Conditions*).
- A description of the process attached to the command (*Description*).

For Input and Output data, optional bytes are written into square brackets ([]).

List of commands:

A cross (X) in the PN532 column indicates if the command may be used with the PN532 configured as initiator or/and with the PN532 configured as target.

The "Command Code" column gives the value of the command code (CC in the two represented frames below) that is used in the frame from the host controller to the PN532.

00	00	FF	LEN	LCS	D4	CC	Optional Input Data	DCS	00
----	----	----	-----	-----	----	----	---------------------	-----	----

00	00	FF	LEN	LCS	D5	CC+1	Optional Output Data	DCS	00
----	----	----	-----	-----	----	------	----------------------	-----	----

For the "RF Communication" commands, when commands are dedicated to the PN532 as initiator (respectively Target) a **In** prefix has been added (respectively **Tg** prefix)

Table 12. Command set

Command	PN532 as Initiator	PN532 as Target	Command Code	Page
Miscellaneous				
Diagnose	X	X	0x00	69
GetFirmwareVersion	X	X	0x02	73
GetGeneralStatus	X	X	0x04	74
ReadRegister	X	X	0x06	76
WriteRegister	X	X	0x08	78
ReadGPIO	X	X	0x0C	79
WriteGPIO	X	X	0x0E	81
SetSerialBaudRate	X	X	0x10	83
SetParameters	X	X	0x12	85
SAMConfiguration	X	X	0x14	89
PowerDown	X	X	0x16	98

⁵ The frame representation does not include the complete frame, but only the following field:

- TFI,
- Command Code,
- When needed the input or output data.

Thus, the Preamble, Start of Packet, LEN, LCS, DCS and Postamble are omitted in this description.

Command	PN532 as Initiator	PN532 as Target	Command Code	Page
R F c o m m u n i c a t i o n				
RFCConfiguration	X	X	0x32	101
RFRRegulationTest	X	X	0x58	107
I n i t i a t o r				
InJumpForDEF	X		0x56	108
InJumpForPSL	X		0x46	113
InListPassiveTarget	X		0x4A	115
InATR	X		0x50	122
InPSL	X		0x4E	125
InDataExchange	X		0x40	127
InCommunicateThru	X		0x42	136
InDeselect	X		0x44	139
InRelease	X		0x52	140
InSelect	X		0x54	141
InAutoPoll	X		0x60	144
T a r g e t				
TgInitAsTarget		X	0x8C	151
TgSetGeneralBytes		X	0x92	158
TgGetData		X	0x86	160
TgSetData		X	0x8E	164
TgSetMetaData		X	0x94	166
TgGetInitiatorCommand		X	0x88	168
TgResponseToInitiator		X	0x90	170
TgGetTargetStatus		X	0x8A	172