

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA



Propuestas de aplicaciones de RFID en seguridad de instalaciones.

PRESENTADO POR:

RONALD SALVADOR ESCOBAR RAMÍREZ

EDDYS ELÍAS LACAYO LÓPEZ

SAÚL ROSA CHAVARRIA

PARA OPTAR AL TÍTULO DE:

INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, ENERO DE 2016

UNIVERSIDAD DE EL SALVADOR

RECTOR INTERINO :

LIC. JOSÉ LUIS ARGUETA ANTILLÒN

SECRETARIA GENERAL :

DRA. ANA LETICIA ZAVALA DE AMAYA

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO :

ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL

SECRETARIO :

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERÍA ELÉCTRICA

DIRECTOR :

ING. ARMANDO MARTÍNEZ CALDERÓN

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO ELECTRICISTA

Título :

Propuestas de aplicaciones de RFID en seguridad de instalaciones.

Presentado por :

RONALD SALVADOR ESCOBAR RAMÍREZ

EDDYS ELÍAS LACAYO LÓPEZ

SAÚL ROSA CHAVARRIA

Trabajo de Graduación Aprobado por:

Docente Asesor :

ING. WERNER DAVID MELÉNDEZ VALLE

San Salvador, Enero de 2016

Trabajo de Graduación Aprobado por:

Docente Asesor :

ING. WERNER DAVID MELÉNDEZ VALLE

AGRADECIMIENTOS.

Primero quisiera agradecer a Dios por darme la oportunidad de haber logrado obtener este triunfo tan importante en mi vida; luego agradecer a mis padres por todo su apoyo y consejos que me ayudaron como motivación para seguir siempre adelante a pesar de todas las adversidad es.

También quiero agradecer a todas las personas que estuvieron siempre a mi lado apoyándome a mis compañeros de estudio principalmente, con los cuales compartimos muchos momentos alegres así como también momentos difíciles pero que con la ayuda de Dios y con el esfuerzo de todos logramos salir adelante para lograr nuestro objetivo.

SAÚL ROSA CHAVARRIA.

AGRADECIMIENTOS.

Agradezco a Dios por otorgarme la fortaleza, la sabiduría, inteligencia y cada virtud para el desarrollo y la bendición de finalización de esta carrera, gracias a mi madre Juana María Hernández de Lacayo López, a mi padre David Humberto Lacayo González, y a mis hermanos Duilio Martín Lacayo López, Ana María Lacayo López, Francisco Javier Reyes López; por apoyarme y darme el ánimo para levantarme y seguir adelante, así como por todos los consejos que me brindaron durante mis años en la Universidad, gracias a mis padres por cada palabra de ánimo que me dieron porque me sirvieron como guía para lograr mis objetivos.

Gracias a mi familia por estar siempre incondicionalmente a mi lado, preocupándose por mí, y por apoyarme en todas mis decisiones, estando pendiente de mí y velando por mi bienestar, Muchas Gracias. Gracias a mis amigos/as y compañeros por apoyarme, por escucharme y por las palabras de sabiduría que me brindaron, y que fueron un trampolín para continuar adelante en mi trayectoria académica.

Al Ing asesor del presente trabajo, por la orientación, el seguimiento y la supervisión continúa del mismo. A mis compañeros de tesis por todas las horas que trabajamos juntos en la elaboración del trabajo.

Gracias a todos por apoyarme en cada paso que recorrí en mi carrera, por cada palabra de aliento que me hizo más fuerte para vencer todos los obstáculos que se presentaron, gracias Dios por permitirme ver la finalización de mi carrera y te pido bendiciones y fortaleza para el futuro, en mi vida y en mi ámbito laboral, sabiduría para poder tomar buenas decisiones y salir adelante.

EDDYS ELÍAS LACAYO LÓPEZ.

AGRADECIMIENTOS.

Primeramente agradezco a Dios todopoderoso, por brindarme la oportunidad y la bendición de finalizar mis estudios, por darme la fortaleza, salud y esperanza de poder alcanzar este anhelo que hoy se vuelve una realidad.

A mis padres Elsa Rosalina Ramírez de Escobar y Rafael Escobar Menjívar, por brindarme su ayuda espiritual y económica hasta esta etapa de mi vida.

A mi hermana Patricia Lizette Escobar Ramírez por todo el apoyo que me brindó en el transcurso de mi formación académica.

A todas las personas que me ayudaron ya sea directa o indirectamente en todo este período de aprendizaje. Docentes, Compañeros, personal de laboratorio y administrativo de la escuela de ingeniería eléctrica.

Gracias y que Dios los bendiga.

RONALD SALVADOR ESCOBAR RAMÍREZ.

ACTA DE CONSTANCIA DE NOTA Y DEFENSA FINAL

En esta fecha, Jueves, 10 de diciembre de 2015, en la Sala de Reuniones de la Escuela de Ingeniería Eléctrica, a las 4:30 horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. MSc. e Ing. Salvador de Jesús Germán
Director Interino

Firma:



2. MSc. e Ing. José Wilber Calderón Urrutia
Secretario Interino

Firma:



Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

1- Ing. Werner David Melendez Valle

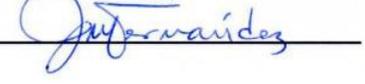
Firma:



2- Ing. Gerardo Marvin Jorge Hernández



3- Ing. José Miguel Hernández



Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

Propuestas de aplicaciones de RFID en seguridad de instalaciones.

A cargo de los Bachilleres:

- Escobar Ramírez Ronald Salvador
- Lacayo López Eddys Elías
- Rosa Chavarría Saúl

Habiendo obtenido en el presente Trabajo una nota promedio de la defensa final: 8.9

(OCHO PUNTO NUEVE)

Índice Contenido

CAPITULO I: TECNOLOGIA RFID	1
INTRODUCCIÓN.....	1
1.1 DEFINICION.....	2
1.2 RESEÑA HISTORICA	3
1.3 TEORIA DE FUNCIONAMIENTO DE SISTEMAS RFID	6
1.4 DESCRIPCION TECNICA	8
1.4.1 Lectores RFID.....	8
1.4.2 Etiquetas RFID	10
1.4.3 Antenas en Lectores RFID.....	13
1.4.4 Hardware de procesamiento (Middleware).....	13
1.4.5 Sistema en conjunto RFID en la banda UHF	14
1.5 NORMAS	25
1.5.1 Regulación y estandarización RFID.....	25
1.5.2 El código EPC	26
1.5.3 Normas ISO.....	29
1.5.4 Normas EPC.	30
1.6 ALTERNATIVAS DE MERCADO	32
1.7 SITUACIÓN ACTUAL EN EL SALVADOR Y PREVISIONES FUTURAS	33
1.8 JUSTIFICACION	37
1.9 LIMITANTES Y ALTERNATIVAS PARA SUPERARLAS	37
CAPÍTULO 2: Hardware-Software: selección y pruebas.	39
INTRODUCCIÓN.....	39
2.1 ESTUDIO DE COMPONENTES ASOCIADOS A TECNOLOGÍA RFID	40
2.2 HARDWARE	41
2.2.1 Procesador principal.....	41
2.2.2 Interface RFID.....	43
2.2.3 Interface GSM:	46
2.2.4 Antenas:	49
2.3 SOFTWARE	71
2.3.1 Descripción del Software del Sistema.	71

CAPÍTULO 3: Análisis y propuestas de diseño.	105
Introducción	105
3.1 Descripción de las Pruebas.....	106
3.1.1 Pruebas de detección de Tag	106
3.1.2 Propuesta de solución	107
3.1.3 Presupuesto.....	131
CONCLUSIONES	132
BIBLIOGRAFÍA:	133
ANEXO A: MANUAL DE USUARIO.	137
A.1 MANUAL DE USUARIO: HARDWARE	137
A.2 MANUAL DE USUARIO: SOFTWARE	139
ANEXO B: CÓDIGOS DE CONFIGURACIÓN DE LOS DIFERENTES ELEMENTOS QUE COMPONEN EL SISTEMA.	258

Índice de Figuras:

Figura 1.1: Diagrama del funcionamiento de un sistema RFID.....	2
Figura 1.2: Etiqueta RFID.....	3
Figura 1.3.....	5
Figura 1.4: Comunicación entre el módulo y el transponder.....	6
Figura 1.5: Frecuencias RFID	7
Figura 1.6: Diagrama del Lector	9
Figura 1.7: Componentes electrónicos de lectores RFID	9
Figura 1.8: Diagrama de bloques de una etiqueta escribible.....	11
Figura 1.9: Lector con antena móvil.....	13
Figura 1.10: Acoplamiento inductivo.	15
Figura 1.11: Acoplamiento capacitivo.....	16
Figura 1.12: Propagación de ondas electromagnéticas.	16
Figura 1.13: Reducción de la potencia por unidad de área recibida en función de la distancia.....	17
Figura 1.14: Área activa para antenas de 300 y 600MHz.....	17
Figura 1.15: Modulación ASK y FSK.....	19
Figura 1.16: Proceso de comunicación en un sistema RFID pasivo	20
Figura 1.17: Colisiones en sistemas RFID	21
Figura 1.18: Taxonomía de los protocolos anticollision para etiquetas RFID	22
Figura 1.19: Esquema de la propagación de una onda electromagnética y su onda reflejada.	23
Figura 1.20: “Internet de las cosas”, una nueva dimensión.....	25
Figura 1.21: Código de barras	26
Figura 1.22: Detalle del EPC en su versión de 96 bits.	28
Figura 1.23: Esquema de los diferentes principios de operación según la memoria de la etiqueta de los sistemas	29
Figura 1.24: Lectura de un código de barras.....	32
Figura 1.25: Pinzas antirrobo con chip RFID.....	33
Figura 1.26: Torres de antenas antirrobo para centros comerciales que utilizan tecnología RFID..	33
Figura 1.27	34
Figura 1.28: Diferentes aplicaciones en las que se espera gane terreno la tecnología RFID en El Salvador.....	36
Figura 2.1: Diagrama de bloques del sistema prototipo de seguridad utilizando la tecnología RFID.	40
Figura 2.2: Tarjeta Raspberry pi B+	42
Figura 2.3: Arduino Mega 2560.....	44
Figura 2.4: Módulo Lector AS3992 RFID UHF.....	45
Figura 2.5: Antena de 5dBi PCB UHF RFID 902 - 928 MHz, 5 cm X 5 cm.....	45
Figura 2.6: Etiqueta UHF RFID adhesiva, pasiva.....	46
Figura 2.7: Arduino Uno	47
Figura 2.8: Shield GPRS/GSM	48
Figura 2.9: Antena parche circular implementada.....	49

Figura 2.10: Medición de patrón de radiación circular a derechas en el plano y-z a 915 Mhz.	50
Figura 2.11: Ganancia vrs. Frecuencia.....	50
Figura 2.12: Alimentación por línea microstrip.....	51
Figura 2.13: Calculo obtenido con ADS para una línea microstrip de 50 Ω	52
Figura 2.14: Arreglo de antena parche rectangular.	53
Figura 2.15: Patrón de radiación teórico de antena parche rectangular.	54
Figura 2.16: Diseño de antena parche rectangular en HFSS.	54
Figura 2.17: Impedancia de entrada de la antena parche a 925 Mhz.....	55
Figura 2.18: Valor del coeficiente de reflexión.	55
Figura 2.19: Ganancia total de la antena parche.	56
Figura 2.20: Esquema básico de antena helicoidal.	56
Figura 2.21: Parámetros dimensionales del helicoide.	57
Figura 2.22: Adaptador de impedancia de antena helicoidal.	57
Figura 2.23: Patrón de radiación teórico de antena helicoidal.	58
Figura 2.24: Calculo de parámetros de antena helicoidal.....	58
Figura 2.25: Antena helicoidal 915 Mhz implementada.	59
Figura 2.26: Esquema para la realización de pruebas de los diferentes tipos de antenas.	59
Figura 2.27: Sintonizador de RF utilizado como analizador de espectro.	60
Figura 2.28: Interfaz gráfica de analizador de espectro RTL-SDR.	60
Figura 2.29: Lecturas experimentales con los 3 tipos de antenas (distancias en cms).....	61
Figura 2.30: Antena parche circular 915Mhz.	63
Figura 2.31.....	63
Figura 2.32.....	64
Figura 2.33: Distintas clases de etiqueta RFID, con las cuales se realizaron las pruebas.	65
Figura 2.34: Etiqueta RFID.....	69
Figura 2.35: Dimensiones física de la etiqueta Impinj H47.	70
Figura 2.36: Grafica de comparación entre distancia de lectura de la etiqueta Impinj H47 y la frecuencia a la cual se lee.	71
Figura 2.37: Diagrama de flujo del funcionamiento básico del código del sistema.....	72
Figura 2.38: Diagrama de flujo del funcionamiento de la función de autenticación de usuarios..	73
Figura 2.39: Diagrama de flujo del menú principal del sistema.....	74
Figura 2.40: Diagrama de flujo del funcionamiento del proceso de inventario sencillo.	77
Figura 2.41: Diagrama de flujo del funcionamiento del proceso de inventario con RSSI.	79
Figura 2.42: Diagrama de flujo del funcionamiento del proceso de inventario continuo.	82
Figura 2.43: Diagrama de flujo del funcionamiento del proceso de escaneo sencillo.....	83
Figura 2.44: Diagrama de flujo del funcionamiento del proceso de escaneo con RSSI.	85
Figura 2.45: Diagrama de flujo del funcionamiento del proceso de escaneo continuo.	87
Figura 2.46: Diagrama de flujo del funcionamiento del proceso de las opciones avanzadas.	91
Figura 2.47: Diagrama de flujo del funcionamiento del proceso de usuarios.	98
Figura 2.48: Diagrama de flujo del funcionamiento del proceso de usuarios SMS.	101
Figura 3.1: Primera planta del edificio de EIE.	106
Figura 3.2: Segunda planta del edificio de EIE	106

Figura 3.3.....	107
Figura 3.4.....	107
Figura 3.5.....	108
Figura 3.6.....	109
Figura 3.7.....	109
Figura 3.8.....	110
Figura 3.9.....	111
Figura 3.10.....	111
Figura 3.11.....	112
Figura 3.12.....	114
Figura 3.13: Se realizan un escaneo y no se detectó ninguna etiqueta, a) fue realizado con la interface principal, b) fue realizado con la interface WEB.....	116
Figura 3.14: Se realizó un escaneo y se detectó una etiqueta, la cual no tenía autorización para salir del lugar a controlar, a) fue realizado con la interface principal, b) fue realizado con la interface WEB.....	116
Figura 3.15: Mensaje de texto de alarmas del sistema prototipo RFID dirigidos a un usuario específico, al realizar un escaneo de seguridad.....	117
Figura 3 16.....	117
Figura 3 17.....	118
Figura 3 18.....	118
Figura 3 19.....	119
Figura 3.20: Primera planta de Edificio de EIE.....	121
Figura 3.21: Segunda planta de Edificio de EIE.....	121
Figura 3.22: Salón de bodega 2.....	122
Figura 3.23: Salón de sistemas embebidos.....	122
Figura 3.24: Laboratorio de Control Automático.....	123
Figura 3.25: Laboratorio de aplicaciones fotovoltaicas.....	123
Figura 3.26.....	124
Figura 3.27: Interface gráfica principal: a) Registro de usuario para el ingreso del mismo al sistema, b) Menú principal de opciones o funciones, c) Menú de opciones avanzadas, el cual muestra funciones para la modificación física del funcionamiento del mismo.....	125
Figura 3.28: Interface gráfica WEB: a) Registro de usuario para el ingreso del mismo al sistema, b) Menú principal de opciones o funciones, c) Menú de opciones avanzadas, el cual muestra funciones para la modificación física del funcionamiento del mismo.....	127
Figura 3.29: Ventana de interface principal de los resultados de la función de inventario sencillo, en este muestra que se encuentran todos los elementos a controlar en el sitio.....	128
Figura 3 30: Ventana de interface WEB de los resultados de la función de inventario sencillo, en este muestra que se encuentran todos los elementos a controlar en el sitio.....	129
Figura 3.31: Ventana de interface principal de los resultados de la función de inventario sencillo, en esta muestra que en el lugar falta un elemento.....	130
Figura 3.32: Ventana de interface WEB de los resultados de la función de inventario sencillo, en esta muestra que en el lugar falta un elemento.....	130

Figura 3.33: Mensaje de texto de alarmas del sistema prototipo RFID dirigidos a un usuario especifico, al realizar un inventario.	131
Figura A.1: Conexión entre el módulo lector RFID AS3992 y el arduino MEGA 2560.	137
Figura A.2: Led de encendido o power del módulo RFID AS3992.....	138
Figura A.3: Conexión entre el arduino UNO y el arduino shield SIM 900 (GPRS Shield V1.1(B))....	138
Figura A.4: Puentes en el shield SIM 900 para elegir los pines de comunicación TX y RX.....	138
Figura A.5: Vista de los puertos USB y Ethernet de Raspberry PI B+ y posición de conexión de los módulos arduino MEGA 2560 y arduino UNO.	139
Figura A.6: Conexión con la Raspberry B+ por medio de un cliente SSH llamado PUTTY.....	140
Figura A.7: Activación del servidor VNC en la Raspberry B+.....	141
Figura A.8: Conexión con el escritorio remoto o VNC, con la Raspberry B+.....	142
Figura A.9: Activación del servidor DJANGO.	143
Figura A.10: Conexión con la interfaz gráfica WEB o con el servidor DJANGO desde el Host del usuario.....	143
Figura A.11: Autenticación del usuario con el sistema de seguridad RFID, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.	144
Figura A.12: Menú principal del sistema, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.....	145
Figura A. 13: Visualizar base de datos, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.....	147
Figura A.14: Tipos de escaneos de radio frecuencia que puede realizar el sistema a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.	147
Figura A. 15: Tipos de inventarios que el sistema puede generar, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.....	148
Figura A.16: Resultado de la ejecución de la subfunción inventario sencillo, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.	150
Figura A.17: Resultado de la ejecución de la subfunción inventario con RSSI, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.	151
Figura A. 18: Resultado de la ejecución de la subfunción inventario continuo, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.	153
Figura A.19: Tipos de escaneos que el sistema puede generar, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.....	154
Figura A.20: Resultado de la ejecución de la subfunción escaneo sencillo, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.	155
Figura A.21: Resultado de la ejecución de la subfunción escaneo con RSSI, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.	156
Figura A.22: Resultado de la ejecución de la subfunción escaneo continuo, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.	157
Figura A.23: Seleccionar etiqueta específica, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.....	158
Figura A.24: Advertencia del sistema al usuario de la existencia o no de la etiqueta por ende del elemento, a) Interfaz gráfica situada en el módulo de control, b) Interfaz gráfica WEB.	159

Figura A.25: Menú de la función información de la etiqueta específica, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	160
Figura A.26: Formas es que el sistema busca la información específica del elemento, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	161
Figura A.27: Resultado de la búsqueda de la información de una etiqueta específica, a) El elemento fue encontrado, interface gráfica situada en el módulo de control, b) El elemento fue no encontrado, interface gráfica situada en el módulo de control, c) El elemento	162
Figura A.28: Ventana de la función de insertar elemento a la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	163
Figura A.29: Advertencia de que ya existe el elemento que el usuario quiere registrar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	164
Figura A.30: Resultado de la función de insertar elemento a la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	165
Figura A.31: Visualizar base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	166
Figura A.32: Menú de la función información de un préstamo específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	167
Figura A.33: Formas es que el sistema busca la información específica del préstamo, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	170
Figura A.34: Resultado de la búsqueda de la información de un préstamo específico, a) El préstamo fue encontrado, interface gráfica situada en el módulo de control, b) El préstamo fue no encontrado, interface gráfica situada en el módulo de control, c) El préstamo fue encontrado, interface WEB y d) El préstamo fue no encontrado, interface WEB.....	171
Figura A.35: Visualizar base de datos de alarmas, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	172
Figura A.36: Menú de la función información de una alarma específica, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	173
Figura A.37: Formas en que el sistema busca la información específica de una alarma, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	174
Figura A.38: Resultado de la búsqueda de la información de una alarma específica, a) El elemento fue encontrado, interface gráfica situada en el módulo de control, b) El elemento fue no encontrado, interface gráfica situada en el módulo de control, c) El elemento fue encontrado , Interface gráfica WEB, d) El elemento fue no encontrado, Interface gráfica WEB.	175
Figura A.39: Autenticación del usuario para ingresar al menú de opciones avanzadas, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	176
Figura A.40: Menú de opciones avanzadas, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.....	177
Figura A.41: Menú de la función potencia de la antena, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	178
Figura A. 42: Mensajes de información sobre el estado de la antena cuando el usuario elige sobre activar o desactivar la antena, a) ANTENA ACTIVADA Interface gráfica situada en el módulo de control, b) ANTENA DESACTIVADA Interface gráfica situada en el módulo de control, c) ANTENA ACTIVADA Interface gráfica WEB, d) ANTENA DESACTIVADA Interface gráfica WEB.	179

Figura A.43: Visualización del módulo RFID a) Antena desactivada, b) Antena activada.	180
Figura A.44: Ventana para ingresar la nueva potencia de la antena, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	180
Figura A.45: Mensaje de modificación de la potencia de la antena, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	181
Figura A.46: Menú de la función modificar la frecuencia de operación, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	182
Figura A.47: Ventana para ingresar la frecuencia de operación al sistema, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	184
Figura A.48: Mensajes de información sobre el ingreso de la frecuencia de operación, a) Modo salto activado Interface gráfica situada en el módulo de control, a) Modo salto desactivado Interface gráfica situada en el módulo de control, c) Modo salto activado Interface gráfica WEB, d) Modo salto desactivado Interface gráfica WEB.	185
Figura A.49: Ventana para ingresar el password de la etiqueta RFID para poder modificar su ID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	186
Figura A.50: Ventana para ingresar el nuevo ID de la etiqueta RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	187
Figura A.51: Advertencias al usuario sobre los errores de no haber encontrado la etiqueta a escribir y/o que el password ingresado por el usuario no es el password de la etiqueta RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	188
Figura A.52: Resultado de escribir una etiqueta RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	189
Figura A.53: Ventana para ingresar el password actual que posee la etiqueta a modificar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	190
Figura A.54: Ventana para ingresar el password nuevo que posee la etiqueta a modificar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	191
Figura A.55: Advertencias al usuario sobre los errores de no haber encontrado la etiqueta y/o que el password ingresado por el usuario no es el password de la etiqueta RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	193
Figura A.56: Resultado de modificar el password una etiqueta RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	194
Figura A.57: Menú de la función de modificar los registro de los elementos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	195
Figura A.58: Opciones de búsqueda del elemento para modificar sus registros, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	197
Figura A.59: Formas en que el sistema busca la información específica del elemento a modificar, Interface gráfica situada en el módulo de control.	197
Figura A.60: Ventanas para ingresar ya sea el nuevo ID, el nuevo NOMBRE o el nuevo DESCRIPCION del elemento, Interface gráfica situada en el módulo de control.	198
Figura A.61: Ejemplo de las ventanas que utiliza la interfaz WEB para modificar los registros de los elementos, se presenta las ventanas para modificar el nombre del elemento y ser elegido en la base de datos ya sea por su nombre actual o por su ID.	199

Figura A.62: Resultados de la modificación de los registros de un elemento en específico (en este caso se modificó su NOMBRE), a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	200
Figura A.63: Menú de la función PRESTAMOS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.....	201
Figura A.64: Ventana para ingresar los datos necesarios para un nuevo elemento en la tabla prestamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	202
Figura A.65: Ventana de advertencia sobre el préstamo de un elemento que no se encuentra registrado en la base de datos principal, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.....	203
Figura A.66: Ventanas de advertencia al usuario, informando del elemento que ya se encuentra registrado en la base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.....	204
Figura A.67: Ventanas de advertencia al usuario, informando del elemento que se encuentra registrado en la base de datos de alarmas, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB, el sistema interroga al usuario si desea continuar y eliminar la información del elemento en la base de datos de alarmas e ingresarlos en la base de datos de préstamos.	205
Figura A.68: El sistema interroga al usuario si desea continuar y eliminar la información del elemento en la base de datos de alarmas e ingresarlos en la base de datos de préstamos, a) Interface gráfica situada en el módulo de control.	206
Figura A.69: El sistema advierte al usuario que la información del elemento en la base de datos de alarmas ha sido eliminada para poder ser registrada en la base de datos de préstamos, a) Interface gráfica situada en el módulo de control.	206
Figura A.70: Resultados de la función al registrar un elemento en la base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	206
Figura A.71: Menú de la función modificar los registros de los elementos de la base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	207
Figura A.72: Opciones de búsqueda del elemento para modificar sus registros en la base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	210
Figura A.73: Formas en que el sistema busca la información específica del préstamo a modificar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	210
Figura A.74: Ventanas para ingresar ya sea la nueva DESCRIPCION del elemento o el nuevo tiempo que durara el préstamo, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	210
Figura A.75: Ejemplo de las ventanas que utiliza la interfaz WEB para modificar los registros de los elementos en la base de datos de préstamos, se presenta las ventanas para modificar la descripción del elemento y ser elegido en la base de datos ya sea por su nombre actual o por su ID.	212
Figura A.76: Ventana para ingresar la nueva información de la persona que realizara o continuara el préstamo del elemento, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	213

Figura A.77: Resultados de la modificación de los registros de un préstamo en específico (en este caso se modificó los datos personales de la persona quien realizo el préstamo), a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	214
Figura A.78: Opciones de búsqueda para poder eliminar la información de un préstamo específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	215
Figura A.79: Formas en que el sistema busca la información específica del préstamo a eliminar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	216
Figura A.80: Forma en que el sistema advierte al usuario de que se eliminara toda la información de la base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	217
Figura A.81: Menú de las funciones de ALARMAS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	218
Figura A.82: Opciones de búsqueda para poder eliminar la información de una alarma específica, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	219
Figura A.83: Formas en que el sistema busca la información específica de la alarma a eliminar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	220
Figura A.84: Forma en que el sistema advierte al usuario de que se eliminara toda la información de la base de datos de alarmas, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	221
Figura A.85: Menú de las sub funciones de la función de USUARIOS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	222
Figura A.86: Visualización de los datos de todos los USUARIOS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	223
Figura A.87: Menú de la función información de un usuario específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	225
Figura A.88: Formas es que el sistema busca la información específica de un usuario, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	227
Figura A.89: Resultado de la búsqueda de la información de un usuario específico, a) La información del usuario fue encontrada, interface gráfica situada en el módulo de control, b) La información del usuario fue no encontrada, interface gráfica situada en el módulo de control, c) La información del usuario fue encontrada, Interface gráfica WEB, d) La información del usuario fue no encontrada, Interface gráfica WEB.	228
Figura A.90: Ventana para ingresar los datos necesarios para un nuevo usuario en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	229
Figura A.91: Ventanas de advertencia de que la informando del usuario ya se encuentra registrado en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	230
Figura A.92: Resultados de la función al registrar un usuario en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	231
Figura A.93: Menú de la función modificar los registros de los usuarios de la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	232
Figura A.94: Forma en que el sistema busca la información específica del usuario a modificar, Interface gráfica situada en el módulo de control.	232

Figura A.95: Ventanas para ingresar ya sea el nuevo nombre de USUARIO o el nuevo PASSWORD, a) Interface gráfica situada en el módulo de control, b) Ejemplo de las ventanas para modificar la información específica de un usuario, Interface gráfica WEB.	233
Figura A.96: Ventana para ingresar la nueva información del usuario, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	234
Figura A.97: Resultados de la modificación de los registros de un usuario en específico (en este caso se modificó el nombre de USUARIO), a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	235
Figura A.98: Opción de búsqueda para poder eliminar la información de un usuario específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	236
Figura A.99: Forma en que el sistema advierte de que se eliminara toda la información de la base de datos de usuarios, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	237
Figura A.100: Menú de las sub funciones de la función de USUARIOS SMS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	238
Figura A.101: Visualización de los datos de todos los USUARIOS SMS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	239
Figura A.102: Menú de la función información de un usuario SMS específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	240
Figura A.103: Formas es que el sistema busca la información específica de un usuario SMS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	243
Figura A.104: Resultado de la búsqueda de la información de un usuario SMS específico, a) La información del usuario SMS fue encontrada, interface gráfica situada en el módulo de control, b) La información del usuario SMS fue no encontrada, interface gráfica situada en el módulo de control, c) La información del usuario SMS fue encontrada, Interface gráfica WEB, d) La información del usuario SMS fue no encontrada, Interface gráfica WEB.	244
Figura A. 105: Ventana para ingresar los datos necesarios para un nuevo usuario SMS en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	245
Figura A.106: Ventanas de advertencia de que la informando del usuario SMS ya se encuentra registrado en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	246
Figura A.107: Resultados de la función al registrar un usuario SMS en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	246
Figura A.108: Menú de la función modificar los registros de los usuarios SMS de la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	247
Figura A.109: Forma en que el sistema busca la información específica del usuario SMS a modificar, Interface gráfica situada en el módulo de control.	248
Figura A.110: Ventanas para ingresar ya sea el nuevo número de teléfono y el nuevo comentario, a) Interface gráfica situada en el módulo de control, b) Ejemplo de una ventana para la modificación de la información de un usuario sms específico Interface gráfica WEB.	248
Figura A.111: Ventana para ingresar la nueva información del usuario SMS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	250

Figura A.112: Resultados de la modificación de los registros de un usuario SMS en específico (en este caso se modificó el nombre de USUARIO), a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.....	251
Figura A.113: Opción de búsqueda para poder eliminar la información de un usuario SMS específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.....	252
Figura A.114: Forma en que el sistema advierte de que se eliminara toda la información de la base de datos de usuarios SMS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	253
Figura A.115: Menú de la función sobre eliminar la información de los registros de los elementos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	254
Figura A.116: Opciones de búsqueda para poder eliminar la información de un elemento específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.....	255
Figura A.117: Formas en que el sistema busca la información específica del elemento a eliminar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	256
Figura A.118: Forma en que el sistema advierte al usuario de que se eliminara toda la información de la base de datos principal, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.	257

Índice de Tablas:

Tabla 1.1: Desarrollo de los estándares por frecuencia.....	4
Tabla 1.2: Evolución histórica de la tecnología RFID.....	6
Tabla 1.3: Características generales de RFID, según la frecuencia	8
Tabla 1.4: Características específicas a cada rango de frecuencia.....	8
Tabla 1.5: Diferencias entre etiqueta activa y una pasiva.	12
Tabla 1.6.....	18
Tabla 1.7: Potencias de emisión en diversos dispositivos.....	18
Tabla 1.8.....	34
Tabla 2.1: Características técnicas de la Raspberry pi B+.	43
Tabla 2.2: Características técnicas de Arduino Mega 2560	44
Tabla 2.3: Características técnicas del módulo RFID UHF AS3992.....	46
Tabla 2.4: Características técnicas del Arduino Uno.....	47
Tabla 2.5: Características del sustrato.	52
Tabla 2.6: Valores de intensidad medidos con el analizador de espectro.....	60
Tabla 2.7.....	62
Tabla 2.8: Parámetros técnicos de antena parche circular.....	62
Tabla 2.9: Datos obtenidos con antena parche circular 915Mhz.	64
Tabla 2.10: Características técnicas de las etiquetas.....	69
Tabla 3.1.....	110
Tabla 3.2.....	112
Tabla 3.3.....	113
Tabla 3.4.....	113
Tabla 3.5.....	114
Tabla 3.6.....	115
Tabla 3.7.....	120
Tabla 3.8.....	120
Tabla 3.9.....	124
Tabla 3.10.....	131
Tabla A. 1: Tabla de funcionamiento del GPRS Shield V1.1 (B) arduino, SIM900.....	139

CAPITULO I: TECNOLOGIA RFID

INTRODUCCIÓN

En la actualidad los procesos de identificación automática (Auto-ID) vía ondas de radiofrecuencia, están expandiendo su uso en diversos campos de la industria, e incluso los encontramos frecuentemente en nuestra vida diaria en empresas de servicios, diversos almacenes, sistemas logísticos y similares, dado que brindan opciones de control de inventario y alternativas de seguridad a bajo costo, que son atractivas para el comercio e industria. Estas aplicaciones emplean ondas de radio y etiquetas codificadas para asignar identificaciones únicas a productos y con ello facilitar su ubicación; la tecnología en la cual se basan, se empezó a desarrollar durante la segunda guerra mundial, y ha continuado su mejora hasta nuestros días, de tal forma que las alternativas de mercado son diversas, a costos relativamente accesibles.

En el presente trabajo, se pretende explorar las características de esta tecnología y a partir de ello proponer un sistema de seguridad y control de inventario que se pueda utilizar en la FIA-UES. Para demostrar la viabilidad de la propuesta, se implementará un prototipo con el cual se buscará ejemplificar los beneficios de este tipo de soluciones de seguridad de bajo costo. Así mismo, se propondrán alternativas que permitan ampliar la cobertura de este tipo de sistemas, sin dejar de lado la normativa aplicable a estas aplicaciones.

Actualmente se dispone de estándares internacionalmente aceptados para el uso de estas aplicaciones, especialmente en la banda de UHF (siglas del inglés Ultra High Frequency o 'Frecuencia Ultra Alta'), es por ello que en el presente trabajo, se ha seleccionado dicha banda, para construir el prototipo previamente citado.

1.1 DEFINICION.

La tecnología RFID (siglas del inglés Radio Frequency Identification o ‘Identificación por Radio Frecuencia’) es un término en el que se incluyen aquellas tecnologías que utilizan ondas de radio para identificar de manera automática personas, animales u objetos. A nivel de bloques, sus componentes más importantes se muestran en la figura 1.1

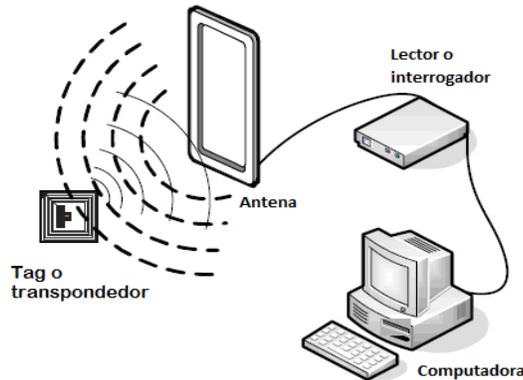


Figura 1.1: Diagrama del funcionamiento de un sistema RFID.

En donde:

- Tag o transpondedor: es un dispositivo de tamaño pequeño capaz de ser adherido a un objeto, el cual contiene una antena y un micro chip, con los que se tiene la capacidad de recibir y transmitir órdenes por radio frecuencia (o RF), el micro chip contiene un número de identificación único capaz de diferenciarlo de otros.
- Antena: es la interface entre el lector RFID y el canal inalámbrico usado para la comunicación con las Tags, mismas que pueden ser tanto de verificación de estado, como de órdenes o comandos de grabación.
Lector o interrogador: se encarga de interrogar a las etiquetas RFID por medio de órdenes transmitidas por RF, además tiene la funcionalidad de leer la información de dichas etiquetas y de procesar dicha información.
- Computadora: es la responsable de la interacción con el usuario, además de la programación o control del lector.

Por tanto RFID es un método de almacenamiento y recuperación de datos de manera inalámbrica y sin necesidad de contacto directo con el elemento a identificar. Esta propiedad de comunicación inalámbrica entre el módulo lector y las etiquetas RFID, es la que permite consolidar este tipo de sistemas como alternativas de seguridad y control.

En el mercado existen diversos sistemas de identificación automática, como por ejemplo los sistemas basados en códigos de barra, la limitante principal de estos es que se necesita que el lector y la etiqueta estén a escasos centímetros entre ellos, aumentando así el esfuerzo y el número de personas necesarios para realizar procedimientos tipo inventarios o de control, mientras que la tecnología RFID permite una mayor separación, llegando incluso a los 6 u 8 metros, dependiendo de las características del elemento radiante asociado al módulo lector.

La forma constructiva básica de una etiqueta o tag, se muestra en la figura 1.2, en donde sus componentes principales son:

- Microchip: Dispositivo capaz de almacenar información específica para la identificación individual de un artículo o producto
- Antena: Elemento radiante que va acoplado al microchip.
- Condensador: Elemento utilizado para acoplar impedancia del microchip y la antena.
- Contacto:

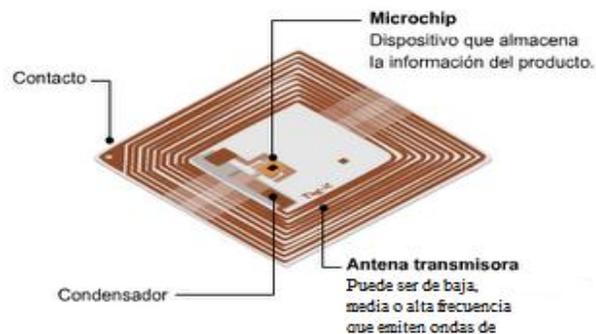


Figura 1.2: Etiqueta RFID.

1.2 RESEÑA HISTORICA

Se tiene antecedentes del uso de la RFID desde la década del 20, pero su historia documentada comienza en 1940, cuando se desarrolló como medio para la identificación de aeronaves militares (aliados y enemigos) durante la segunda guerra mundial; a dichos dispositivos se les dio el nombre de RADAR (acrónimo del inglés Radio detecting and ranging o ‘Detección y localización por radio’). El uso de tales equipos permitía la detección a kilómetros de distancia, pero no su identificación específica. El ejército alemán descubrió que, si los pilotos balanceaban sus aviones, al volver a la base cambiaría la señal de radio reflejada de vuelta. Este método permitía distinguir a los aviones alemanes de los aliados y se convirtió en el primer dispositivo de RFID.

En los años 60’s y 70’s se comenzó a desarrollar y/o concretar el uso de esta tecnología, como sistemas antirrobo que, usando ondas de radio, determinaban si un objeto había sido pagado o no; otra aplicación importante son los sistemas de control de acceso, en estos se emplea una tarjeta con una antena que emite una señal a un lector adherido a la puerta y cuando ésta se valida la cerradura se desbloquea, permitiendo el ingreso y registro de la persona que realizó este procedimiento, sin necesidad de llaves.

Las primeras patentes en relación a la tecnología RFID se realizaron en 1973.

Estados Unidos monto sistemas para el manejo de puertas en centrales nucleares, que se abrían al paso de camiones que portaban materiales, equipados con una antena. También se desarrolló un sistema para el control de ganado, distinguiendo que animales habían sido vacunados, y los que no; insertando bajo la piel de estos, una etiqueta RFID.

En un principio las etiquetas RFID eran de mayor tamaño que las actuales, y disponían de complejos sistemas con bobinas de metal, antenas y cristales, del tamaño de un teclado de

computadora; pero gracias a los avances y la miniaturización de la tecnología, se logró una reducción apreciable en sus dimensiones. En algún momento se utilizaron etiquetas acopladas capacitivamente, las cuales usaban tinta de carbono conductivo, en lugar de las grandes bobinas metálicas para la transmisión de datos, la tinta era imprimida en hojas de papel y leída por dispositivos a través de ondas de radio, también se utilizaron chips de silicón de alrededor de 3 milímetros de ancho, con capacidad para almacenar 96 bits de información¹.

Con el tiempo se han desarrollado mejoras en la capacidad de omisión y recepción, así como en la cobertura que es posible alcanzar, lo cual ha llevado a extender su uso en ámbitos que van desde un sencillo control de seguridad de una casa, hasta la seguridad nacional; como sucede con los pasaportes expedidos en la actualidad en los Estados Unidos, que lleva asociadas etiquetas RFID.

Los estándares en el campo de RFID, abordan áreas fundamentales, como lo son: Protocolo en el interfaz aéreo y contenido de los datos (prácticamente en el enlace de datos, que se da entre etiquetas y lector, como lo son anticollisión y protocolos de comunicación) certificación y aplicaciones (protocolo de datos e interfaces de programación). Estos son controlados una parte por la ISO (Organización Internacional para la Estandarización) y otra parte por AUTO-ID, como se muestra en la Tabla 1.1.

Banda de frecuencias	Nombre del estándar	Detalles
Menos de 135 KHz	ISO 18000-2	
6,765 ~ 6,795 MHz		
7,4 ~ 8,8 MHz		
13,55 ~ 13,57 MHz	ISO 18000-3	Uso común
26,96 ~ 27,78 MHz		Aplicaciones especiales
433 MHz	ISO 18000-7	Tags activos en Asia
868 ~ 870 MHz	ISO 18000-6 A/B	Europa
902 ~ 928 MHz	AutoID clase 0/1	Norteamérica
860 ~ 960 MHz	EPCGlobal Gen2	Mundial
2400 ~ 2483 MHz	ISO 18000-4	
8725 ~ 8785 MHz	ISO 18000-5	Uso raramente en RFID

Tabla 1.1: Desarrollo de los estándares por frecuencia².

Los sistemas RFID especialmente en la banda de UHF tuvieron un auge en 1999, ya que se fundó la sociedad Auto-ID Center, la cual incluye entre sus miembros, a casi 100 compañías internacionales y cinco de las principales universidades del mundo³.

La función principal de este centro es seleccionar la tecnología y crear los estándares adecuados para la identificación de objetos por medio de las frecuencias de radio, utilizando etiquetas pasivas y lectores inductivos.

¹ En el apartado 1.5.2 del documento se describen las diferentes memorias existentes para los EPC.

² Del documento: Sistema de reparto de recursos en un sistema RFID con lectores interferentes y tags móviles (Álvaro Campillo Soler, Universidad Politécnica de Cartagena, 2013)

³ Según documento: La Tecnología de identificación de objetos Auto-ID: Hacia la integración total de sistemas distribuidos (Jaime Moraleda Novo, Universidad de Castilla-La Mancha).

Auto-ID ofrece información completa, estructurada y actualizada en tiempo real; por lo que se afecta, por un lado a la etiqueta identificativa y por el otro, a la estructuración de los sistemas asociados de almacenamiento y gestión de la información.

La etiqueta identificativa contiene un número, llamado código de producto electrónico, conocido como EPC (acrónimo del inglés Electronic Product Code o código de producto electrónico)⁴; este número sirve de referencia, ya que conduce hasta el archivo almacenado en una base de datos, como se muestra en la figura 1.3.

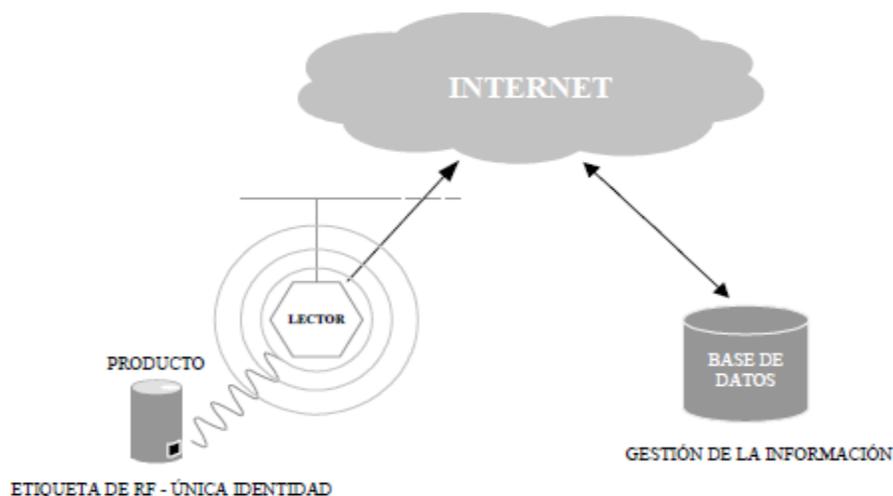


Figura 1.3⁵

El resultado de poder acceder a la base de datos, a través internet, hará posible obtener de forma instantánea, información relativa a cualquier objeto, en cualquier lugar del mundo, desde un ordenador conectado a red una vez que se haya realizado su identificación.

La tabla 1.2 muestra hitos importantes asociados a la tecnología RFID

Décadas	Eventos
1940-1950	<ul style="list-style-type: none"> Se utiliza el radar. Se desarrolló en la segunda guerra mundial RFID, se inventa alrededor del año 1948.
1950-1960	<ul style="list-style-type: none"> Primeras investigaciones sobre la tecnología RFID Experimentos en laboratorios.
1960-1970	<ul style="list-style-type: none"> Desarrollo de la teoría de RFID. Primeras pruebas de campo.
1970-1980	<ul style="list-style-type: none"> Avance de RFID. Se aceleran las pruebas con RFID. Primeras implementaciones adaptadas con RFID.

⁴ En el apartado 1.5.2 del documento se presentan las características del EPC

⁵ Según documento: La Tecnología de identificación de objetos Auto-ID: Hacia la integración total de sistemas distribuidos (Jaime Moraleda Novo, Universidad de Castilla-La Mancha).

1980-1990	<ul style="list-style-type: none"> Las aplicaciones comerciales de RFID, cobran importancia.
1990-2000	<ul style="list-style-type: none"> Surgen los estándares. RFID se despliega más ampliamente.
2000-2010	<ul style="list-style-type: none"> Aparecen aplicaciones innovadoras. Combinación de RFID con servicios móviles personales. RFID subcutáneo para animales y humanos. RFID llega a formar parte de la vida diaria

Tabla 1.2: Evolución histórica de la tecnología RFID.

1.3 TEORIA DE FUNCIONAMIENTO DE SISTEMAS RFID

El funcionamiento de este tipo de tecnología se basa en un transmisor y un receptor, en donde al transmisor se le denomina “Módulo Lector o Interrogador⁶”, básicamente éste lo que hace es producir un pulso a una frecuencia determinada, es decir con modulación ASK (Modulación por Desplazamiento de Amplitud⁷), para lo cual se vale de una antena, el pulso viaja por el aire hasta alcanzar la etiqueta, luego dicha etiqueta por medio de su microchip y su antena, genera la respuesta correspondiente hacia el lector, en la figura 1.4 se observa esta interacción. La respuesta en cuestión, contiene los datos que identifican al elemento al cual está adherida y desde el módulo interrogador, se puede utilizar esta información, para el uso que se desee, por ejemplo, para almacenarla en una base de datos o procesarla para ser enviada a otro tipo de dispositivos (mensajes de texto a dispositivos móviles, correos electrónicos, etc.).

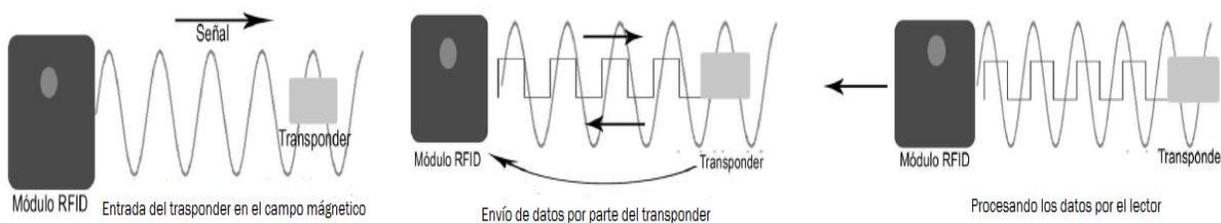


Figura 1.4: Comunicación entre el módulo y el transponder

El rango de frecuencia en los que operan los equipos RFID es muy amplio, abarca desde 50kHz y 2.5GHz, del espectro electromagnético, por lo que en tipos de sistemas RFID se tiene la clasificación, mostrada en la figura 1.5:

⁶ En el apartado 1.4.1 del documento, se presentan las características técnicas del lector

⁷ En el apartado 1.4.5.c) del documento, se presentan los protocolos de modulación

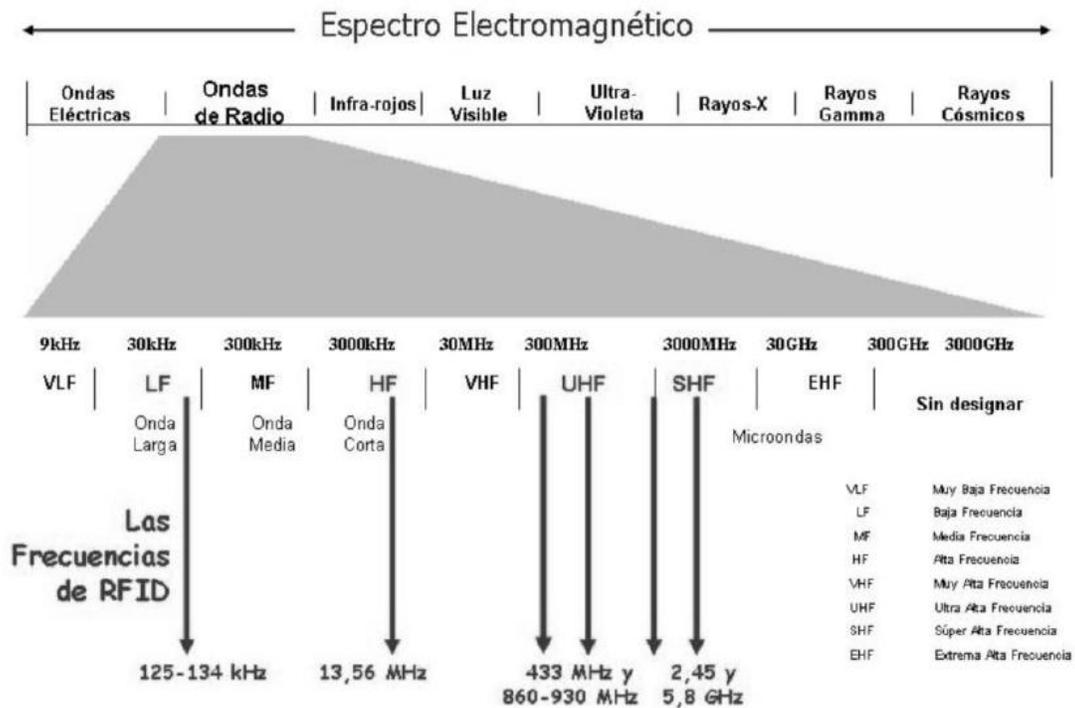


Figura 1.5: Frecuencias RFID⁸

En la Tabla 1.3, se observan características generales, de las diferentes frecuencias en sistemas RFID.

Frecuencia	LF 125 KHz	HF 13.56 MHz	UHF 868-915 MHz	Microondas 2.4, 5.8 GHz
Rango Máximo de lectura (valor típico en tags pasivos)	< 0.5 m	~ 1 m	~ 3 m	~ 1 m
Características generales	Relativamente caro, incluso en grandes cantidades. Requiere antena de cobre mayor, mas cara. Los tags inductivos son más caros que los capacitivos. Menor sensibilidad a degradaciones de rendimiento de metales y líquidos, aunque el rango de lectura es menor.	Menos caro que los tags de LF. Rango de lectura relativamente corto y menores tasas de datos comparado con frecuencias mayores. Utilización en aplicaciones que no necesitan lectura a grandes distancias o de varios tags simultáneamente.	En grandes cantidades, los tags UHF tienen el potencial de ser más baratos que los de LF y HF debido al avance del diseño de circuitos integrados. Ofrece una buena relación entre la distancia de lectura y el rendimiento, sobre todo para la lectura simultánea.	Características similares a UHF pero con mayores tasas de lectura. Más susceptible a degradaciones de rendimiento debido a metales y líquidos. Señal más direccional
Alimentación del Tag	Generalmente pasiva. Acoplamiento	Generalmente pasiva. Acoplamiento inductivo	Pasiva con acoplamiento	Pasiva con acoplamiento

^{8 10} Del documento: Análisis de antenas de UHF para aplicaciones de RFID (México, 2010)

	inductivo	o capacitivo.	electromagnético capacitivo o activo con batería integrada.	electromagnético capacitivo o activo con batería integrada.
Tasa de Datos	Menor ←————→ Mayor			
Influencia de metales ó líquidos	Mejor ←————→ Peor			
Tamaño del tag pasivo	Mayor ←————→ Menor			

Tabla 1.3: Características generales de RFID, según la frecuencia⁹

En la Tabla 1.4, se observan características más específicas, de las diferentes frecuencias en sistemas RFID.

Parámetros	Baja Frecuencia (< 135 KHz)	Alta Frecuencia (13.56 MHz)	Ultra Alta Frecuencia (433 MHz, 860 MHz, 928 MHz)	Frecuencia Microondas (2.45 GHz, 5.8 GHz)
Cobertura	Menor ←————→ Mayor			
Tamaño de la etiqueta	Mayor ←————→ Menor			
Velocidad de lectura de datos	Menor ←————→ Mayor			
Lectura en presencia de líquidos ó metales	Mejor ←————→ Peor			
Lectura en presencia de interferencias electromagnéticas	Peor ←————→ Mejor			

Tabla 1.4: Características específicas a cada rango de frecuencia¹⁰

Para el presente trabajo se utiliza un módulo Lector RFID en la banda UHF del espectro electromagnético (860 – 960 Mhz).

1.4 DESCRIPCION TECNICA

1.4.1 Lectores RFID.

^{9 10} Del documento: Sistema de reparto de recursos en un sistema RFID con lectores interferentes y tags móviles (Álvaro Campillo Soler, Universidad Politécnica de Cartagena, 2013)

10

El módulo lector, es el principal elemento del sistema RFID, este está compuesto por una antena, un módulo electrónico de radiofrecuencia y un módulo electrónico de control, en la figura 1.6, se puede observar el diagrama en bloques de un lector.



Figura 1.6: Diagrama del Lector¹¹

La operación básica del lector, es la siguiente: la antena del lector crea un campo magnético, cuyo radio de acción, varía dependiendo de la forma de la antena, de la potencia del lector y de la frecuencia en la que emita¹², en la Figura 1.7 podemos observar lo que traen los lectores en general internamente, aunque puede cambiar algún componente electrónico entre un lector y otro, dependiendo del modelo del fabricante.

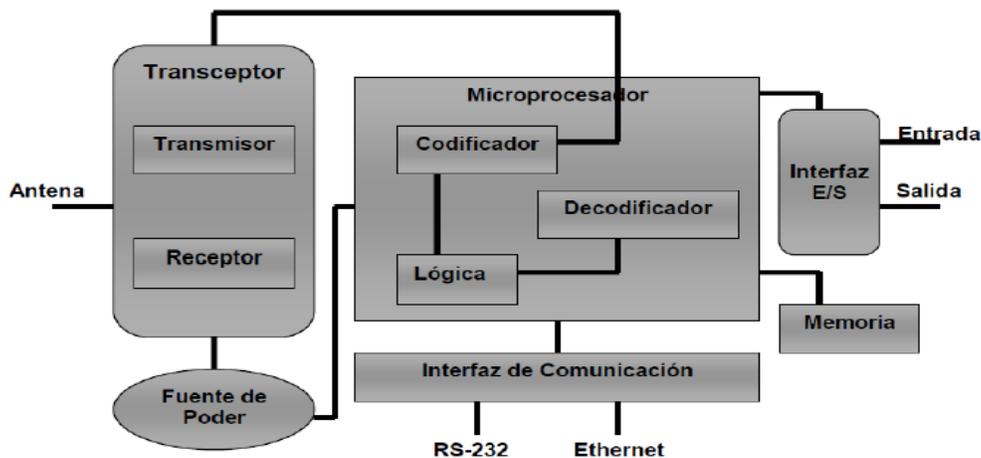


Figura 1.7: Componentes electrónicos de lectores RFID¹³

Se tienen varios modelos en el mercado, todos tienen el mismo funcionamiento básico, la principal característica de estos es la frecuencia a la que operan, por lo que podemos encontrar lectores que trabajen en las cuatro bandas del espectro electromagnético anteriormente citadas. Según la tecnología con la que ha sido construido el lector, así será la capacidad de manipular la respuesta de

11

¹² En el apartado 1.4.5.a) y 1.4.5.b) de este documento se presentan las características técnicas de esta interacción

¹³ Del documento: Análisis de antenas de UHF para aplicaciones de RFID (México, 2010)

varias etiquetas a la vez, es decir, estos pueden dividir las respuestas de las etiquetas (tratarlas como respuestas individuales) en un tiempo mínimo, no perceptible a la visión humana¹⁴.

Existen lectores de escritura/lectura¹⁵, capaces de escribir la información en las etiquetas, o de eliminarla de la orden de lectura; esto quiere decir que el lector manda la orden de escribir sobre la etiqueta; al eliminarla por ejemplo, cada vez que el lector la detecte, la etiqueta no generara la respuesta de radio frecuencia, para ese lector en específico.

La sensibilidad de un Lector, se determina por la potencia mínima de respuesta para reconocer una etiqueta, el funcionamiento en conjunto con la etiqueta, es presentado en el apartado 1.4.5.

1.4.2 Etiquetas RFID¹⁶

En la figura 1.8 se observa el diagrama de bloques de una etiqueta escribible y a continuación se mencionan todos los componentes de las etiquetas en general:

- Memoria no volátil donde se almacenan los datos.
- Una memoria ROM donde se almacenan instrucciones básicas para el funcionamiento de dicha etiqueta.
- En ocasiones también se puede incorporar una memoria RAM para almacenar datos, como un número de serie.
- Una antena o micro antena, el propósito de esta, es recibir energía procedente de la señal del lector (absorber las ondas RF) y responder a dicha señal con la transmisión de su ID (esta información está contenida en el Microchip). La longitud de la antena, es directamente proporcional a la longitud de onda, a la cual opera la etiqueta.
- MicroChip, es un minúsculo ordenador que contiene la memoria y las instrucciones para poder responder a la petición del módulo lector.
- Algunos componentes electrónicos los cuales procesan la señal de la antena y demás funciones dentro de la misma etiqueta.
- Sustrato, es el material que mantiene al chip y a la antena juntos y protegidos. En la mayoría de los tags consiste en una fina película de plástico.

¹⁴ En el apartado 1.4.5.d) se presentan las características de los sistemas Anticolisión en RFID

¹⁵ La mayoría de literatura solo los llama Módulo Lector, cuando se refieren a módulos de Lectura/escritura (graba), fijarse en la hoja técnica de datos del módulo en específico.

¹⁶ En el apartado 1.1 del documento se presenta la definición de Etiqueta y sus componentes principales

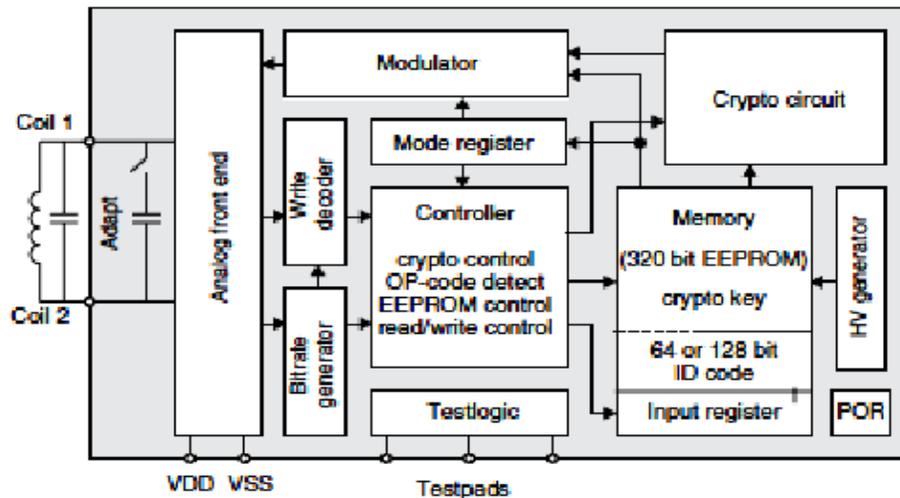


Figura 1.8: Diagrama de bloques de una etiqueta escribible.

1.4.2.1 Clasificación de las etiquetas RFID.

- Por la forma en que es suministrada la energía a la misma etiqueta.
- ✓ Etiqueta pasiva: Esta no contiene una fuente de alimentación, la energía necesaria para realizar sus procesos, se obtiene del mismo pulso de radio frecuencia generado por el lector, el modulo lector genera una señal de radio frecuencia el cual contiene la información necesaria para generar una interrogante sobre la existencia de la etiqueta, y también contiene un pequeño pulso de energía capaz de otorgarle la energía necesaria para que la etiqueta realice las funciones necesarias para responder, por lo que para que la etiqueta tenga alguna fuente de energía necesita estar cercana al campo, o en el rango de escaneo del lector, si no es así la etiqueta nunca responderá ya que no poseerá la energía necesaria para realizar la respuesta.
- ✓ Etiqueta activa: Estas poseen su propia fuente de poder, ya sea en forma de batería o en una célula solar; esto quiere decir que la etiqueta tiene un suministro de energía autónomo, que no depende de los escaneos del lector, estos tipos de etiquetas también se dividen en dos tipos: el primero son etiquetas activas que reaccionan cuando la misma etiqueta detecta el pulso generado por el lector, cuando lo hace, se energiza con su propia fuente de poder, comienza a realizar sus procesos internos y genera un pulso de respuesta hacia el lector; la etiqueta solo se activa cuando detecta el pulso de radio frecuencia generado por el lector. El segundo tipo de etiquetas activas generara el pulso de respuesta aun cuando el lector no ha realizado su escaneo; este tipo de etiquetas aumenta el alcance de detección del lector.
- ✓ Etiqueta semiactiva: Esta tiene similar funcionamiento que las etiquetas activas porque posee una fuente de alimentación propia, la diferencia es que esta fuente de alimentación es ocupada solo para alimentar las funciones del microchip y no para transmitir la señal de respuesta, cuando la etiqueta detecta la señal transmitida por el lector, dicha señal es absorbida por la antena de la etiqueta (como en la etiqueta pasiva) y el microchip se alimenta de la fuente de poder de la etiqueta para realizar todos los procesos lógicos, luego

el microchip da la orden de responder al lector y para producir dicha señal de radio frecuencia, la etiqueta obtiene la potencia de transmisión de la energía absorbida por la señal del lector.

En la Tabla 1.5 se muestran las diferencias entre las etiquetas activas y pasivas.

	RFID Activa	RFID Pasiva
Fuente de poder de las etiquetas	Interna de la tarjeta	Energía transferida desde el lector vía RF
Batería de las etiquetas	Sí	No
Disponibilidad de alimentación para etiquetas	Continua	Sólo dentro del campo del lector
Potencia de la señal requerida desde el lector a la etiqueta	Baja	Alta (puede activar la etiqueta)
Potencia de la señal disponibles desde la etiqueta al lector	Alta	Baja

Tabla 1.5: Diferencias entre etiqueta activa y una pasiva.

- Otra forma de división de los tipos de etiqueta RFID pueden ser de solo lectura o lectura/escritura, en las etiquetas de solo lectura su número de identificación ya lo trae por defecto, sin la oportunidad de poder modificarlo a las necesidades del usuario; y las etiquetas de lectura/escritura traen un número de identificación; pero se es capaz de poder modificar dicha información, cuantas veces se desee (depende de las características técnicas de la etiqueta en específico, ya que hay algunas que se pueden escribir una o pocas veces), esto es una gran ventaja ya que al vender o cambiar el elemento al cual la etiqueta está adherida, a dicha etiqueta se le puede modificar su información interna.

1.4.2.2 Capacidad de memoria de la etiqueta.

Entre mayor sea la capacidad de la memoria interna de la etiqueta, aumentara su costo, dependiendo de esta capacidad de memoria, influirá en el contenido que se le insertara o/y de la complejidad del programa, para procesar la información de dichas etiquetas. Existen diferentes capacidades de memoria, para poder diferenciarlas, se les otorga un número de identificación único, este se encuentra regido por EPC Global¹⁷.

1.4.2.3 Frecuencia de operación.

Al igual que en los lectores, en las etiquetas podemos encontrar que operan en las cuatro bandas de frecuencia del espectro electromagnético citadas en el apartado 1.3 de este documento.

¹⁷ En el apartado 1.5.2 del documento, se describen las diferentes memorias existentes para los EPC.

Una frecuencia mayor, significa una mayor velocidad a la hora de la transición de los datos entre las etiquetas y el lector, también se eleva el costo de dicha tecnología,

1.4.3 Antenas en Lectores RFID.

Las antenas son los elementos radiantes en los sistemas RFID, emiten señales de los lectores y leen las señales de respuestas de las etiquetas.

Existen lectores que poseen antenas integradas¹⁸, estos lectores son utilizados en escaneos de forma manual (en la figura 1.9, observamos un lector con una antena móvil), ya que el rango de cobertura de estos es pequeño, sus aplicaciones podrían ser, ocuparse en lugares que se encuentran muy separados, como para que una antena fija los pueda detectar.



Figura 1.9: Lector con antena móvil

Hay otros tipos de lectores que necesitan de una antena externa, estas puedan utilizarse, por ejemplo para generar un inventario en una área grande, en los que los elementos deben ser detectados sin necesidad de un acercamiento con el lector.

El diseño y la estructura de estas antenas, dependerá de la frecuencia en que opera el sistema RFID y por supuesto de la funcionalidad que se le quiera otorgar al mismo, en el apartado 1.4.5.b) se presentan las características técnicas, de las antenas RFID, como lo es, por ejemplo la densidad de potencia.

Otro punto importante a tomar en cuenta, es que la carga electromagnética, de una antena RFID, es aproximadamente menor a una quinta parte de la producida por un teléfono celular, por lo que cinco antenas activas cerca de una persona generan menos carga electromagnética que la antena de un teléfono celular, por lo que las antenas de los equipos RFID no son perjudiciales a la salud humana.

1.4.4 Hardware de procesamiento (Middleware).

¹⁸ En alguna literatura, se encuentran como lectores móviles

Es un dispositivo electrónico que enlaza todos los elementos del hardware del sistema RFID, y los dispositivos de gestión empresarial del sistema (sistemas de almacenamiento y gestión de la información¹⁹), esto lo realiza mediante una aplicación en específico. Su principal función es la de procesar toda la información relacionada a las etiquetas, obtenidas de los escaneos de los lectores (puede ser de cualquier dispositivo lector de un sistema de identificación, como por ejemplo códigos de barras, GPS, RFID, satélite, etc.). Debido a este manipuleo de la información, se es posible individualizar cada proceso, para luego incorporar, más datos sobre los elementos, a los cuales están adheridas las etiquetas, esto quiere decir, que al número de identificación (o EPC) de la etiqueta leída, se le adjuntara información importante de los elementos, como por ejemplo, el nombre del mismo, la fecha de caducidad, la parte del proceso de producción en el que se encuentra actualmente, etc., esta información adicional, se encuentra en una base de datos; el middleware se encarga de extraer los datos que han sido capturados por los lectores, se realiza un tipo de filtrado de la información obtenida de las etiquetas y en caso de ser necesario, agregar más información específica a la obtenida de los escaneos de las etiquetas.

En algunas ocasiones no es necesario el uso del middleware ya que existen lectores que no están conectados a ninguna red, y que solamente leen la información contenida en la memoria interna de las etiquetas y son presentadas al usuario.

Las funciones básicas del middleware son:

- **Procesamientos de los datos.**

Los sistemas RFID son capaces de adquirir datos en tiempo real provenientes de las etiquetas, dichas lecturas pueden ser automáticas o con órdenes manuales pudiéndose procesar y gestionar una gran cantidad de datos. El middleware realiza un filtrado de los datos obtenidos por el lector, ya que al interactuar con él, evitan varias lecturas de una sola etiqueta y así no obtener una sobrecarga de dichos datos, también es capaz de agregar un tipo de valor añadido a la información, antes de enviarla a algún sistema de gestión del usuario.

- **Gestión de dispositivos.**

El middleware también es capaz de controlar el funcionamiento de las etiquetas, lectores, impresoras de etiquetas o cualquier otro dispositivo que posea la tecnología RFID, así como también advertir de algún fallo de cualquier dispositivo, con la tecnología RFID conectado al middleware.

- **Conexión con sistemas de gestión.**

El middleware recolecta la información obtenida de cualquier dispositivo conectado que contenga la tecnología RFID y se encarga de enviarla a los sistemas que la gestionan; usuario o empresa.

1.4.5 Sistema en conjunto RFID en la banda UHF

¹⁹ Ver figura 1.1 del apartado 1.2 de este documento, para mayor claridad

1.4.5.1 Principios de operación

Ya que la comunicación entre el lector y la etiqueta es por medio de un pulso de radio frecuencia que se encuentra dentro de la banda UHF del espectro electromagnético, estos elementos poseen antenas, para dicha comunicación. El requisito indispensable para que se dé la comunicación es que tanto el lector como la etiqueta trabajen a la misma frecuencia.

Con la evolución en el diseño de la tecnología RFID, ha habido dos formas de comunicarse (es decir, dos aproximaciones diferentes en cuanto a la entrega de la energía) desde el lector hasta la etiqueta, los cuales se fundamentan en las propiedades electromagnéticas relacionadas con las antenas de RF, estas son:

- La propiedad de campo cercano (acrónimo del inglés near field), el cual el fenómeno al que se hace referencia, es que si una corriente alterna atraviesa una bobina, esta creará un campo magnético alrededor de la bobina, fenómeno conocido también como: Acoplamiento inductivo (acrónimo del inglés: Inductive Coupling).
- La propiedad de campo lejano (acrónimo del inglés far field), el funcionamiento de estos difiere del de los sistemas a bajas frecuencias que usan la inducción electromagnética, más similar a los transformadores; por lo que también al fenómeno, se le conoce como sistemas de largo alcance ó Acoplamiento capacitivo (acrónimo del inglés Passive Backscatter).

El acoplamiento inductivo se usa tanto para la comunicación de sistemas que operan en la banda de baja frecuencia (LF) como también la de alta frecuencia (HF); por debajo de 100 MHz, se consideran principios del acoplamiento en el campo cercano. el funcionamiento es el siguiente: un lector que entrega una corriente alterna, a través de una bobina, produce un campo magnético alterno, alrededor de dicha bobina. Si una etiqueta con una pequeña bobina se sitúa en ese campo, aparecerá una tensión alterna a través de esta, y si mediante un condensador, incorporado al tag, es almacenada esta tensión, se tendrá una carga almacenada de reserva, la cual podrá utilizarse para alimentar el microchip del Tag.

La corriente inducida, es proporcional a la carga aplicada a la bobina del tag, como se observa en la figura 1.10; Los tags, que ocupan este acoplamiento, utilizan modulación de carga, para enviar los datos, de vuelta al lector; nombre dado, por lo citado anteriormente.

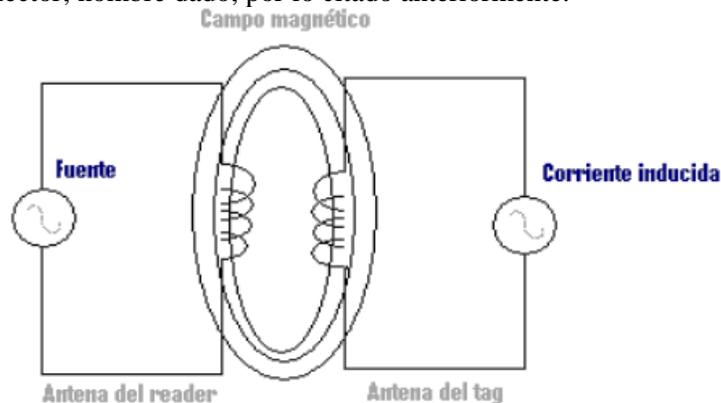


Figura 1.10: Acoplamiento inductivo.

El acoplamiento capacitivo o Backscattering, es usado para la comunicación entre el lector y la etiqueta RFID que funcionan en el rango de frecuencia de UHF y SHF, es una técnica para intercambiar datos, este consiste en que la etiqueta, refleja la onda que recibe del lector, añadiendo en ella (en la onda) la información que tiene almacenada. En este caso, el lector transmite una señal de radiofrecuencia (ver figura 1.11) que la etiqueta recibe, modula (en el apartado 1.4.5.c se explica el protocolo de modulación) y refleja de nuevo hacia el lector, con la información requerida, dependiendo de la fuente de poder de las etiquetas (ya sean pasivas o activas), éstas tomarán de la señal que les llega del lector su alimentación o no, antes de retransmitirla en respuesta.

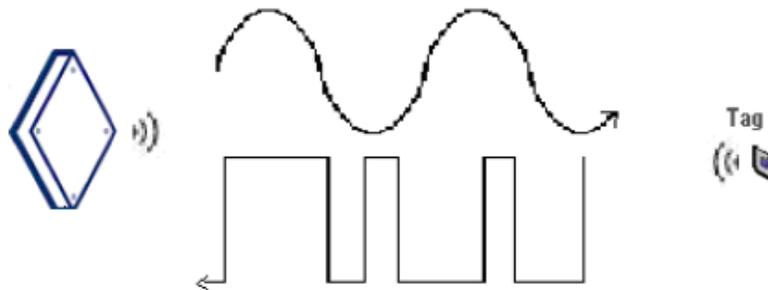


Figura 1.11: Acoplamiento capacitivo.

Es decir, a diferencia del acoplamiento inductivo, aquí no se genera una nueva onda electromagnética sino que se refleja la señal recibida por el lector (ver figura 1.12).

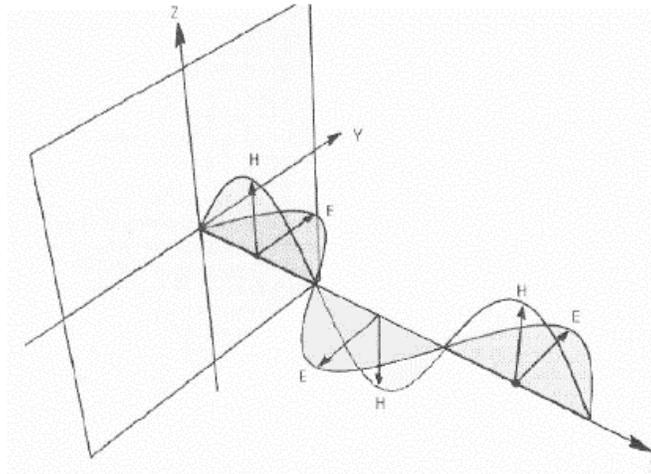


Figura 1.12: Propagación de ondas electromagnéticas.

Una gran ventaja de la utilización de operar a grandes frecuencia (UHF y SHF), es la distancia de lecturas que generan. En el rango de frecuencia UHF se puede generar entre 1 a 8 metros de distancia de lectura con etiquetas pasivas y 100 metros con etiquetas activas, mientras que en las frecuencias bajas, como las que están en el rango de LF y HF las distancias de lecturas son inferiores a un metro, entre lector y etiqueta.

1.4.5.2 Densidad de Potencia en Sistema RFID

La energía para activar el chip la colecta del campo RF recibido de la emisión del lector. Según la frecuencia de la onda, el tag recibe la energía del campo magnético en HF y del campo eléctrico en UHF. A este proceso se le conoce, como acoplamiento (acrónimo del inglés coupling). Técnicamente el acoplamiento (o coupling) se produce cuando la energía se transfiere de un sistema a otro, en nuestro caso del aire, a la antena.

La densidad de potencia (energía por unidad de tiempo) que recibe la etiqueta RFID, depende del tamaño de la antena del lector (el diseño el tamaño de esta, depende de la frecuencia de operación) por lo que a su vez se depende de la frecuencia a la cual fue generado el pulso, de escaneo del lector.

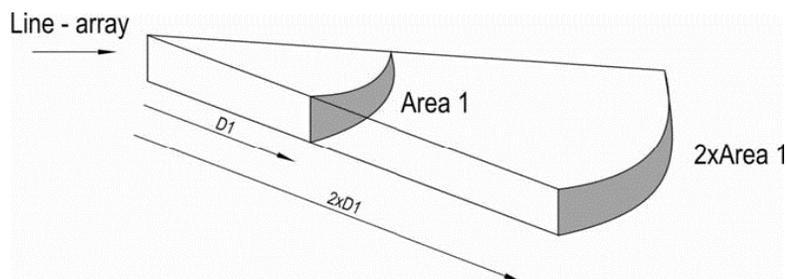


Figura 1.13: Reducción de la potencia por unidad de área recibida en función de la distancia.

La cantidad de la energía que es recibida es función de la apertura de la antena receptora, esto quiere decir que depende de la longitud de la onda del pulso de radio frecuencia de la señal recibida, por ejemplo si una antena con longitud de onda para 300MHz (0.5m) y 600MHz (0.25m), el área activa alrededor de dicha antena tiene la forma de una elipse.

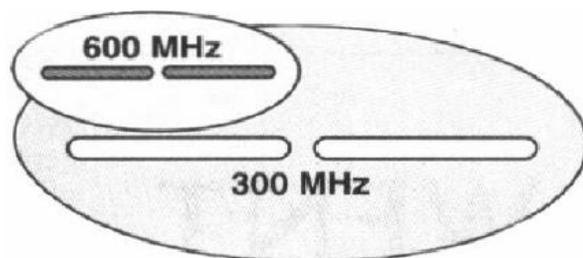


Figura 1.14: Área activa para antenas de 300 y 600MHz.

Como se observa en la figura 1.14, el área de la elipse de la antena que trabaja en la banda de 300MHz es cuatro veces más grande que la que trabaja a 600MHz, esto quiere decir que el área de recepción de la potencia de la señal de la antena de 300MHz es cuatro veces más grande que la antena de 600MHz.

La antena receptora puede ser físicamente más pequeña, aun así tener la misma apertura, ya que existe compensaciones para reducir las dimensiones de la antena, como el ancho de banda o un ajuste a la antena más fino, ya en la práctica esto depende de la energía que radiara el lector, de la frecuencia de operación y de las dimensiones de la antena de las etiquetas.

En la tabla 1.6 se presenta la potencia de emisión permitida (ó ERP del acrónimo inglés Effective Radiated Power) en zonas ya estandarizadas, por organismos internacionales.

EE.UU / Canadá	Europa	Japón	China
Banda UHF (860 – 960 MHz)			
902 – 928 MHz Pe.i.r.p = 4W	865 – 868 MHz Pe.r.p = + 20 dBm	952 – 955 MHz Pe.r.p = 1W + 6 dBantena Gain = 4W	840.5 – 844.5 MHz Pe.r.p = 2W
	865.5 – 868 MHz Pe.r.p = + 27 dBm		920.5 – 924.5 MHz Pe.r.p = 2W
	865.6 – 867.6 MHz Pe.r.p = + 33 dBm		
Banda de Microondas (2.45 GHz)			
2.400 – 2.483 GHz Pe.i.r.p = 4W	2.446 – 2.454 GHz Pe.i.r.p = 500 mW ó 4W	2.400 – 2.4835 GHz	

Tabla 1 6²⁰

Para que exista una buena comunicación entre lector y etiqueta, el lector necesita producir una potencia adecuada, capaz de energizar a las etiquetas a gran distancia, pero a la misma vez se deben atender a las regulaciones actuales, la potencia radiada está limitada a 500mW, esto quiere decir que el rango de lectura entre el lector RFID y le etiqueta es de unos 0.7m a 870MHz. En EEUU y Canadá se permite una potencia radiada de 4W, esto quiere decir que el rango de lectura entre el lector RFID y le etiqueta es de un rango de 2m.

En la Tabla 1.7 se presenta una comparación de potencias de emisión, para diversos dispositivos emisores de ondas electromagnéticas.

Fuente Electromagnética	Potencia de Emisión
Inlay o Tag RFID	10 – 200 μ W
Teléfono móvil	Inferior a 2W
Antena RFID	2W
Estación base GSM	10 – 50 W
Radio FM	300 W
Televisión UHF	5.000.000 W

Tabla 1.7: Potencias de emisión en diversos dispositivos²¹

1.4.5.3 Protocolos de Modulación en Sistemas RFID

El proceso en el cual los pulsos pueden ser usados para transportar información, se conoce como: “Modulación”. Esta, se basa en la modificación de una onda primaria, de manera que pueda seguir un patrón de pulsos, capaz de transmitir información de manera correcta.

²⁰ Según documento: Antenas de Polarización Circular para Sistema RFID (Eloy Guillén Martí, 2010)

²¹ Según documento: Análisis de antenas de UHF para aplicaciones RFID

Los datos que se envían entre la comunicación del lector y las etiquetas RFID se lleva a cabo por medio de codificación binaria y en este tipo de codificación existen dos tipos de transmisión de dichos datos en los sistemas RFID, el primero es la modulación ASK, la cual es como pensar en la señal portadora como un interruptor ON/OFF, en la señal, modulada el valor lógico 0 es representado por la ausencia de una portadora; es la más básica y fácil de implementar, pero como todas las técnicas de modulación de amplitud, esta aproximación es propensa a ser afectada por el ruido del canal de comunicación y en segundo es la modulación FSK la cual altera la frecuencia de la onda, disminuyéndola, siendo menos frecuente en los ceros y con mayor frecuencia en la representación de los unos.

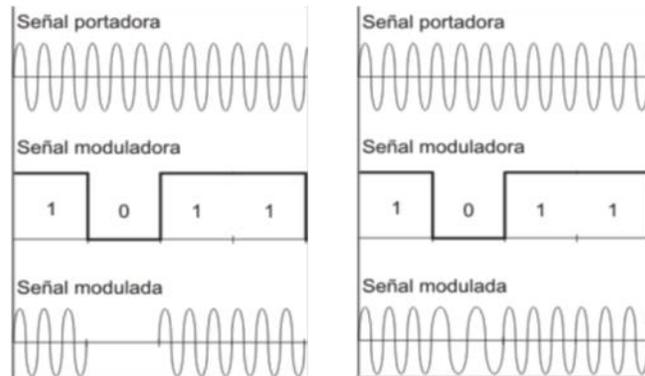


Figura 1.15: Modulación ASK y FSK.

La figura 1.16 nos muestra, como en un sistema RFID se dan los protocolos de modulación

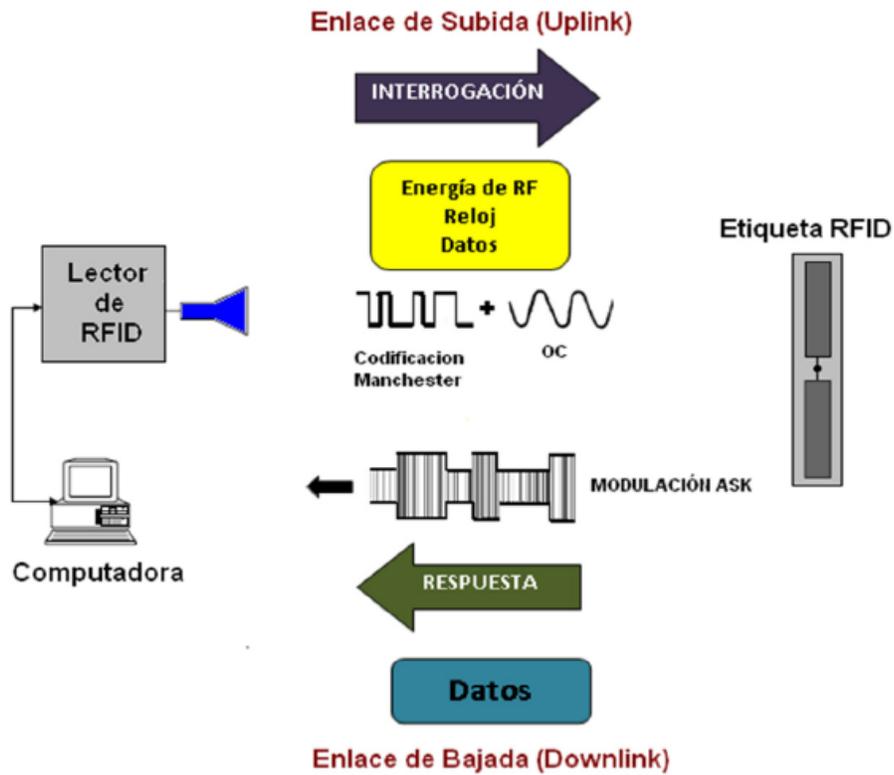


Figura 1.16: Proceso de comunicación en un sistema RFID pasivo²²

1.4.5.4 Protocolos Anticolisión con Sistema RFID

En los equipos RFID se puede encontrar incorporados sistemas anticolidión, este sistema permite al lector, como se observa en la figura 1.17, leer varias etiquetas al mismo tiempo, que se encuentran en el rango del alcance del escaneo del lector.

²² Según Tesis: Análisis de modelos de propagación e interferencia de la tecnología RFID pasiva de UHF para aplicación en la identificación vehicular

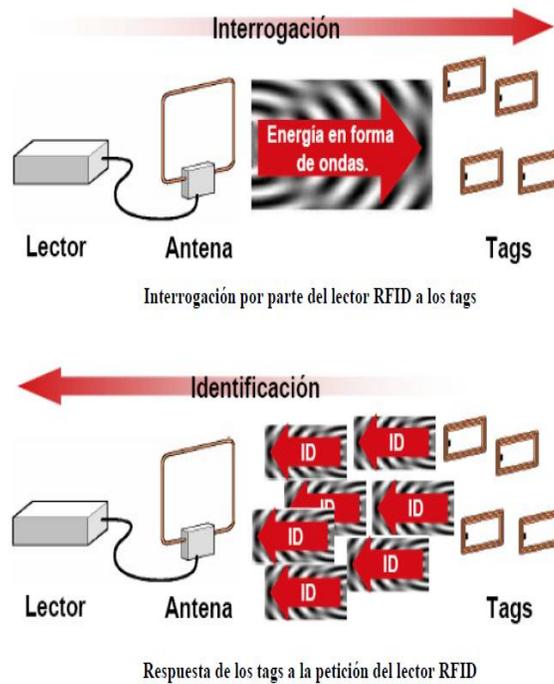


Figura 1.17: Colisiones en sistemas RFID²³

Si un grupo de etiquetas responde al mismo tiempo, el lector es capaz de detectar una colisión y detiene la transmisión de las respuestas de las tarjetas RFID durante un tiempo determinado, luego cada etiqueta responderá individualmente por separado por medio de algún tipo de algoritmo, el cual logra separar la información de respuesta de cada etiqueta, y así evitando que dicha información se corrompa gracias a las colisiones, el problema del multiacceso, ha existido desde hace mucho tiempo en la tecnología radio, en la figura 1.18, se observa los cuatro métodos desarrollados con el objetivo de separar una señal de la de otro cualquiera.

²³ Diseño de antenas UHF para aplicaciones RFID (Hernández Ripoll, Universidad autónoma de Barcelona, 2009)

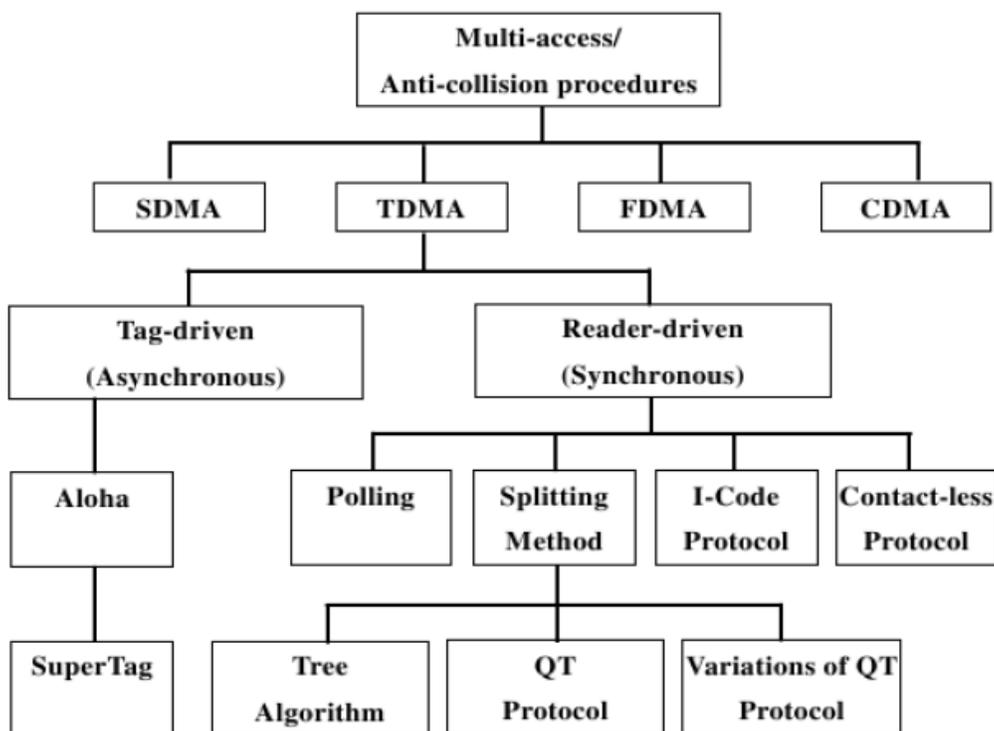


Figura 1.18: Taxonomía de los protocolos anticollisión para etiquetas RFID²⁴

Como se puede distinguir, en los sistemas RFID, las técnicas TDMA son el método más utilizado y con un número mayor de técnicas anticollisión. Los estándares ALOHA y SuperTag tienen como objetivo minimizar el tiempo de identificación de las etiquetas.

Aloha: en este protocolo, la idea básica consiste en que, en el momento en el que el lector, escanea los tags, estos envían su ID al canal independientemente de si ya había otro tag transmitiendo. Evidentemente, la transmisión de varias IDs se puede solapar en el tiempo y provocar entonces una colisión impidiendo, en este caso, realizar una recepción correcta de ninguna de las IDs transmitidas. En el Aloha original, la información que no es transmitida con éxito simplemente se descarta²⁵.

SuperTags: también opera basándose en el principio de anticollisión de Aloha, los tags retransmiten continuamente sus IDs en intervalos aleatorios hasta que el lector les reconoce sus retransmisiones. Tras la recepción de los datos de un tag (su ID), estos tags pueden ser silenciados u obligados a disminuir su velocidad de repetición de envío de datos. Este método de silenciado de tags permite el cálculo correcto de muchos tags en el mismo área. Otra variación de la aproximación conocida como SuperTags implica el silenciado de todos los tags, excepto de un tag el cual se lee asegurando, de esta manera, que no haya ninguna colisión. Tras un cierto periodo de tiempo, los tags silenciados son activados uno por uno hasta que todos son evaluados. En otros métodos derivados del

²⁴ ²⁵ ²⁶ Según documento Sistema de reparto de recursos en un sistema RFID con lectores interferentes y tags móviles

SuperTags, el lector, mediante el envío de datos en blanco o ráfagas de energía, provoca a los tags que respondan tras un retardo generado aleatoriamente²⁶.

1.4.5.5 Absorción, Reflexión, Refracción y Difracción en Sistemas RFID

El pulso generado ya sea por el lector RFID y/o la etiqueta RFID puede ser afectados por algunos de estos factores, pueden provocar interferencias a la hora de la comunicación entre lector y etiqueta.

Por ejemplo la absorción de la onda de radio frecuencia depende el material en el cual esta se propaga, la absorción de energía por el medio o canal se produce debido a que partes de dicha energía se disipa en el material del medio de transición. El fenómeno de refracción y difracción es cuando, el pulso de radio frecuencia pasa por diferentes medios o cuando inciden en el borde de un objeto a la hora de tratar de comunicarse entre lector y etiqueta. Se debe tener en mente que la comunicación entre estos dos elementos se realiza con una frecuencia elevada, dicha comunicación es más propensa a interferencias de este tipo, gracias a la refracción y a la difracción de la onda.

Ya que los pulsos que se generan, a la etiqueta o al lector, son ondas electromagnéticas que pueden ser reflejadas en una superficie conductora como metal, agua, hormigón, etc., este tipo de fenómeno puede ocasionar que la comunicación sea anulada completamente, pero existe el caso de que la misma reflexión pueda ayudar a la comunicación entre la etiqueta y el lector, todo dependerá de la superficie reflejante y las localizaciones de los tres componentes, ya sea el lector, la etiqueta o la superficie reflejante.

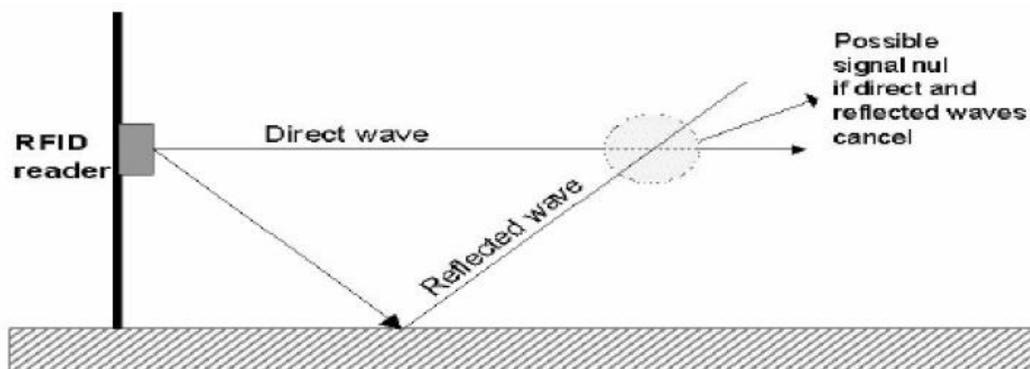


Figura 1.19: Esquema de la propagación de una onda electromagnética y su onda reflejada.

1.4.5.6 Rango de lectura.

El rango de lectura depende de la potencia de transición y en el caso de las etiquetas pasivas, de la energía absorbida de la potencia de la onda de radio frecuencia producida por el lector, el rango de lectura efectivo depende también del factor de absorción del material el cual está unida o adherida

la etiqueta, así como también de las dimensiones de la misma, esto quiere decir que si la etiqueta, es pequeña, menor será el área de absorción de la misma, pero si la etiqueta es grande, el área de absorción también es grande, por ende es proporcional el tamaño de la etiqueta con el rango de lectura del lector RFID.

1.4.5.7 Interferencias.

Cabe destacar que a mayor frecuencia opera el sistema RFID mayor será la sensibilidad a distintos tipos de materiales que pueden ocasionar interferencia a la hora de la comunicación entre el modulo lector y las etiquetas, por ejemplo si se trabaja en la banda de frecuencia HF y si la etiqueta está adherida a una superficie metálica, esta produce interferencia, en la banda de UHF y Microondas, no solo los metales producen interferencia, sino también en los líquidos y otros tipos de materiales; además de las interferencias de otras antenas de las etiquetas, cuando se encuentran muy cercanas entre sí.

Existen muchas fuentes de interferencias para la comunicación entre el lector y la etiqueta RFID, pero en la banda de frecuencia UHF el ruido eléctrico procedente de motores, luces fluorescentes, etc., es mínimo, lo que pueden generar mayor interferencia para este rango de frecuencia puede ser las bandas de frecuencia de telefonía móvil, aparatos que trabajan en la banda ISM, otros sistemas RFID a una distancia muy cercana, etc., aunque este tipo de señales son emitidas por una banda muy estrecha (En el apartado 1.7 se especifica el rango de frecuencia de toda la banda UHF a la que se puede trabajar en El Salvador).

El FHSS (Salto de Frecuencia de Espectro Ensanchado) es una de las formas más efectivas de reducir los efectos de las interferencias y de reducir las interferencias sobre otros dispositivos que comparten el espectro. De este modo la energía transmitida se distribuye a lo largo de la banda de frecuencias, reduciendo las posibles interferencias a otros sistemas y, así, como la frecuencia del receptor está continuamente cambiando, evita los efectos de otros usuarios bloqueando el receptor.

1.4.5.8 Capacidad de lectura direccional.

Los sistema RFID que trabajan en la banda de frecuencia de UHF permite el uso de utilizar antenas direccionales de tamaño pequeño, esto quiere decir que la lectura puede ser direccionada a un punto en específico o a un grupo de etiquetas específicas, y así evitar la lectura de otras etiquetas, esta capacidad es muy útil ya que se es capaz de evitar zonas con gran interferencia.

1.4.5.9 Orientación de la etiqueta.

La orientación de la etiqueta con respecto a la antena del lector RFID es muy importante ya que si el lector posee una antena con polarización lineal es muy importante que la etiqueta deba estar orientada en la dirección de la polarización de la antena del lector, esto quiere decir que si la antena del lector esta polarizada verticalmente para que detecte o lea las etiquetas RFID estas deben de

estar orientadas verticalmente con respecto a la antena del lector, si no es así el lector nunca detectará a las etiquetas aun cuando estas se encuentren a escasos centímetros de este. Pero si la antena del lector RFID está polarizada circularmente esto quiere decir que posee los dos tipos de polarización al mismo tiempo (tanto vertical como horizontal) esto es una ventaja ya que no importa la posición de la etiqueta respecto a la antena del lector, este siempre la detectará ya que si se coloca verticalmente o horizontalmente o en forma inclinada el lector siempre la detectará ya que la antena del mismo produce una onda electromagnética de forma parecida a un círculo (pero si la antena está polarizada verticalmente producirá una onda electromagnética en forma vertical).

1.5 NORMAS

1.5.1 Regulación y estandarización RFID

“RFid es, junto con los scanners, sensores y los dispositivos inteligentes, la puerta de entrada a la denominada Internet de las cosas. Probablemente, durante la próxima década, los procesos de negocio se transformarán en redes flexibles y con gran capacidad de adaptación proporcionando a las empresas acceso a datos precisos, así como conocimiento en tiempo real, que les permita reconocer y responder de manera más efectiva a los cambios de su entorno”²⁷.

En la figura 1.20 se observa una breve representación, de lo que se le llama el “internet de las cosas”.

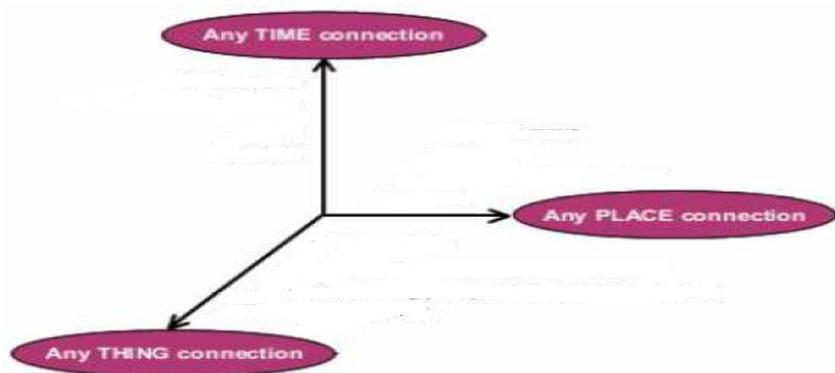


Figura 1.20: “Internet de las cosas”, una nueva dimensión.

Actualmente aún se usan códigos EAN (European Article Number ò International Article Number) es un sistema de código de barras adoptado por alrededor de 100 países y un millón de empresas (en la figura 1.21 se muestra el código de barras); pero los códigos EPC están tomando gran auge en la tecnología moderna.

²⁷ Palabras de Gérald Santucci, que es el jefe de la Unidad de “Empresas en red e identificación por radiofrecuencia RFID” de la comisión europea. Según documento RFID y sus principales aplicaciones (Junta de Castilla y León, 2007).



Figura 1.21: Código de barras

Los primeros dígitos del código de barras EAN identifican el país que otorgó el código, no el país de origen del producto.

1.5.2 El código EPC

En 2003 se creó EPC global como empresa para comercializar el EPC, ya que se abrieron laboratorios en muchos lugares del mundo. No se trata de un estándar, pero si de una organización sin fines de lucro, se encarga de la adopción y estandarización mundial del EPC. El punto central de desarrollo de ésta organización, es de crear un estándar mundial de RFID, con el uso de Internet para transmitir datos mediante una red global denominada: EPC global Network. Toda la información asociada a un código EPC se encuentra en la EPC Global Network, accesible sólo a los usuarios autorizados. Estos datos son gestionados, filtrados y consolidados por el Middleware EPC, el cual los entrega a los sistemas corporativos y a los Sistemas de Información EPC.

EPC Global Network ha tenido varios avances, el más reciente, creado en 2006 es el estándar conocido por EPC Gen2 (de clase 1 y generación 2) al cual ahora se le denomina ISO 18000-6C. Este estándar se empezó en 2004 debido a los numerosos problemas que experimentaron los protocolos de generación 1 y 0 de ésta misma clase. Los sistemas que operan bajo éste estándar son bajo el rango de frecuencias 860 MHz - 960 MHz.

Esta organización ha desarrollado varios estándares, tales como:

- Physical Markup Language (PML): lenguaje basado en XML que define el formato de la información de intercambio entre componentes dentro de la red EPC.
- Object Naming Service (ONS): para la red global, que retiene la información sobre cualquier objeto etiquetado con un tag EPC en el mundo.
- reader protocol (RP)
- Reader Management (RM)
- Low Level Reader Protocol (LLRP)

En el año 2005 la asociación EAN se ha fusionado con la UCC (Uniform code Council) para formar una única organización mundial identificada como GS1.

GS1 El Salvador es el representante exclusivo de EPC Global en el país

El código EPC, está formado por una serie de números (memoria estandarizada) que identifican unívocamente objetos; como artículos, cajas, palets, etc. A continuación se mencionan los diferentes estándares existentes:

- Un bit de información: este tipo de etiquetas son utilizadas como una medida de seguridad en pequeños almacenes o tiendas, esto quiere decir, que la información de la etiqueta

muestra una etiqueta activada (bit=1) cuando el producto o elemento al cual está adherida no sea ha pagado y la información muestra una etiqueta desactivada (bit=0) cuando el usuario de la caja la pase por un escáner y verifique el pago, realmente estas etiquetas no contiene un chip interno ya que solo necesitan un bit para mostrar o informar que si la etiqueta esta activada o desactivada.

- 64 bits de información: Este tipo de etiquetas fueron desarrolladas para la primera versión de la EPC global.
- 96 bits de información: Este tipo de etiquetas son desarrolladas para poder albergar el EPC global, esta versión es la más extendida, y son 96 bits con la intención de asegurar que todos los elementos tengan un EPC o número de identificación único y que sea una versión de bajo costo, este tipo de etiquetas pertenecen a la generación 1 de la EPC global, la cual soporta hasta 96 bits de información.
- 256 bits de información: Este tipo de etiquetas son las desarrolladoras de la generación 2 de la EPC global y aporta un EPC o número de identificación de 256 bits, esto quiere decir que contiene de la suficiente capacidad de memoria como para contener un código de 256 bits.
- Mayor a 256 bits de información: Estas etiquetas además de poder albergar un EPC o código electrónico de producto de hasta 256 bits, según la generación 2 de la EPC global, por ende almacenar más información relativa al elemento o producto la cual están adheridos, como podría ser por ejemplo el nombre del producto, fecha de caducidad, etc., o simplemente para almacenar las medidas tomadas a través de los sensores que pueden llevar las etiquetas RFID, como presión, temperatura, etc.

Los números que conforman el código EPC, identifican al fabricante y el tipo de producto, pero adicionalmente emplea una serie de dígitos extra que permiten identificar un artículo único, en la figura 1.22 se muestra el detalle del EPC en su versión de 96 bits.



Figura 1.22: Detalle del EPC en su versión de 96 bits²⁸.

Para contrastar con los transponders de un solo bit, el cual normalmente explota los efectos físicos, existen transponders que usan un microchip electrónico como sistemas portador de datos. Tienen una capacidad de almacenamiento de información mayor a pocos Kbytes. Para leer o escribir en estos sistemas de almacenamiento se realiza una transferencia de datos entre lector y transponder, esta transferencia puede seguir tres procesos: half duplex, full duplex y secuencial, como se muestra en la figura 1.23.

²⁸ Según documento: La Tecnología de identificación de objetos Auto-ID: Hacia la integración total de sistemas distribuidos. (Universidad de Castilla)

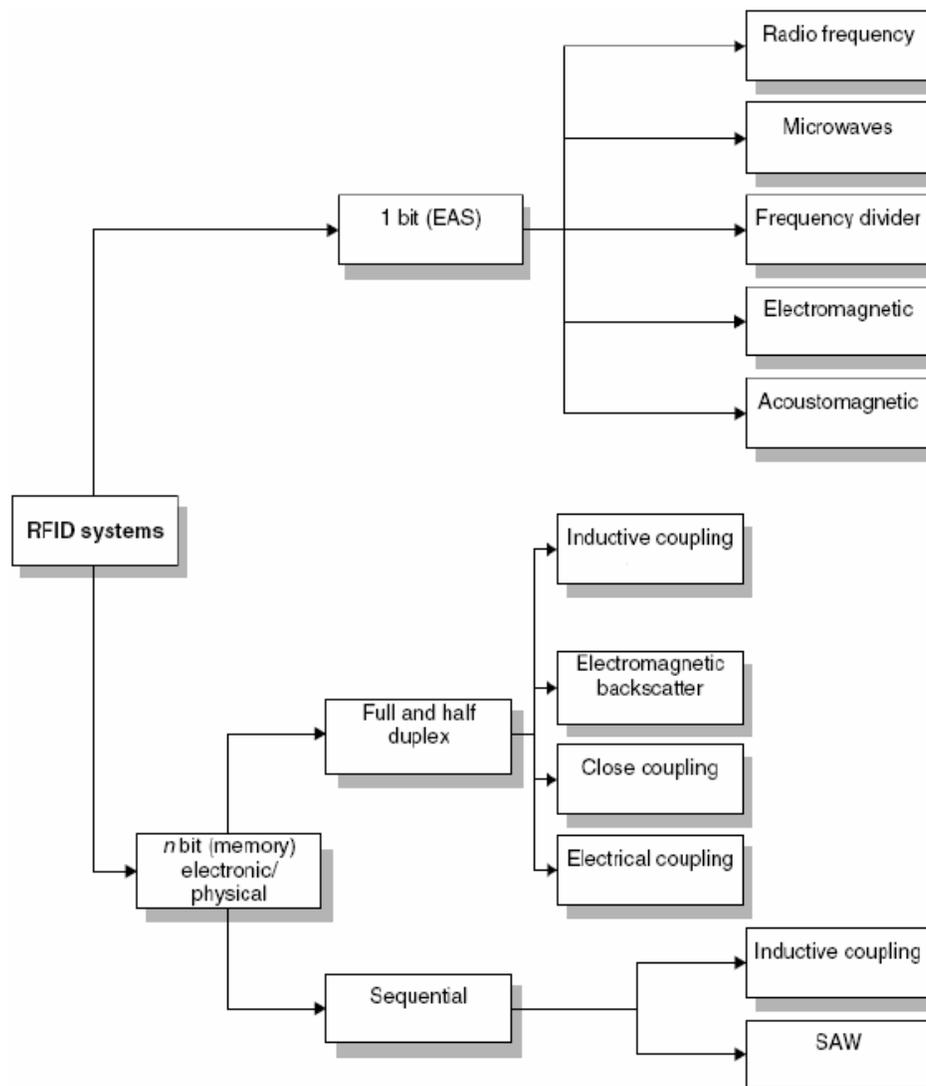


Figura 1.23: Esquema de los diferentes principios de operación según la memoria de la etiqueta de los sistemas ²⁹

1.5.3 Normas ISO.

Está diseñada la normativa para conseguir una operatividad global, en la que se define la comunicación entre las etiquetas RFID y los lectores RFID. También contiene las distintas frecuencias de trabajo utilizadas.

✓ ISO/IEC 18000-2 (LF)

- Define el interfaz aéreo para RFID en dispositivos con rango de frecuencias utilizado menor de 135 KHz.

²⁹ Según documento Estudio, diseño y simulación de un sistema RFID basado en EPC.

- Define parámetros técnicos que incluyen: la frecuencia de operación, la precisión del canal, ancho de banda de canal utilizado.
- Especifica: el protocolo de comunicación entre el interrogador y el tag, y el método de detectar y comunicarse con un tag de entre varios tags.

✓ [1] **ISO/IEC 18000-3 (HF)**

- Define el interfaz aéreo para RFID en dispositivos que operan en un rango de frecuencias de 13.56 MHz.
- Define parámetros técnicos que incluyen: la frecuencia de operación, la precisión del canal, ancho de banda del canal utilizado.
- Especifica: el protocolo de comunicación entre el interrogador y el tag, y el método de detectar y comunicarse con un tag de entre varios tags.

✓ **ISO/IEC 18000-6 (UHF)**

- Define el interfaz aéreo para RFID en dispositivos que operan en un rango de frecuencias que van desde 860 MHz a 960 MHz.
- Define parámetros técnicos que incluyen: la frecuencia de operación, la precisión del canal, ancho de banda de canal utilizado.
- Especifica: el protocolo de comunicación entre el interrogador y el tag, y el método de detectar y comunicarse con un tag de entre varios tags.

✓ **ISO/IEC 15961 RFID for item management – Data protocol: application interface:**

- Esta norma está dirigida a comandos funcionales comunes y características de sintaxis y estructura, por ejemplo, tipos de tags, formatos de almacenamiento de datos, o compresión de los datos.

✓ **ISO/IEC 15962 RFID for item management – Protocol: Data encoding rules and logical memory functions:**

- Trata de estandarizar el procedimiento que el Sistema RFID utiliza para intercambiar información de la gestión a nivel de unidad. Establece un formato de datos uniforme y correcto, una estructura de comandos y el procesamiento de los errores.

✓ **ISO/IEC 15963 for item management – Unique identification of RF tag:**

- La norma se refiere al proceso de registro y uso de la etiqueta RFID. Se diseñó para el control de calidad durante el proceso de fabricación. También está dirigido a la trazabilidad de las etiquetas RFID durante este proceso, su ciclo de vida, entre otras cosas.

1.5.4 Normas EPC.

Las normas EPC son gestionadas por la organización anteriormente mencionada denominada GS1.

✓ **EPC Gen2 (HF)**

- Define los requerimientos físicos y lógicos para un dispositivo de RFID pasivo que opera a un rango de frecuencia de 13.56 MHz
- Este estándar puede ser complementado con la ISO 18000-3
- El estándar hace referencia al sistema compuesta por los lectores y los tags.

✓ **EPC Gen2 (UHF)**

- Define los requerimientos físicos y lógicos para un dispositivo de RFID pasivo que opera a un rango de frecuencia que oscila entre 860 MHz y 960 MHz.
- Este estándar puede ser complementado con la ISO 18000-6
- El estándar hace referencia al sistema compuesta por los lectores y los tags.

EPC global como órgano de estandarización para la RFID en su uso con EPC ha organizado las etiquetas en 6 clases³⁰:

- **Clase 0:** solamente lectura (el número EPC se codifica en la etiqueta durante el proceso de fabricación).
- **Clase 1:** escritura una sola vez y lecturas indefinidas (se fabrican sin número y se incorpora a la etiqueta más tarde).
- **Clase 2:** lectura y escritura.
- **Clase 3:** capacidades de la clase 2 más la fuente de alimentación que proporciona un incremento en el rango y funcionalidades avanzadas.
- **Clase 4:** capacidades de la clase 3 más una comunicación activa con la posibilidad de comunicar con otras etiquetas activas.
- **Clase 5:** capacidades de la clase 4 más la posibilidad de poder comunicar también a etiquetas pasivas.

El nuevo estándar para UHF se ha realizado con los siguientes objetivos³¹:

- Establecer una única especificación UHF, para unificar las existentes como EPC clase 1, EPC clase 0 e ISO 18000-6, partes a y b.
- Diseño para un desarrollo mundial, dirigido a las diferentes regulaciones de diferentes regiones.
- Influenciar y mejorar las especificaciones UHF existentes, además de anticipar posibles aplicaciones futuras (como incluir funcionalidades para etiquetas que contengan sensores).

La especificación Gen2 tiene varios puntos clave³²:

- Las etiquetas RFID podrán comunicarse en cualquier frecuencia entre 860-960 MHz, este requerimiento también afecta a los lectores RFID.
- Las etiquetas (tags) serán capaces de entender tres esquemas de modulación diferentes:
- DB-ASK (Double Sideband-Amplitude Shift Keying)
- SS-ASK (Single Sideband-Amplitude Shift Keying)
- PR-ASK (Phase-Reversal Amplitude Shift Keying)

³⁰ ³¹ ³² Según documento: Diseño de antenas UHF para aplicaciones RFID

31

32

Los lectores determinarán que esquema se utiliza, teniendo en cuenta las regulaciones radio de cada gobierno y las condiciones del entorno.

Los tags pueden transmitir a diferentes velocidades, 4 en concreto: 80 Kbps, 160 Kbps; 320 Kbps o 640 Kbps. Los lectores determinan que velocidad usan. Para situarnos mejor, la Gen1 era de entre 70 y 149 Kbps.

- Los tags Gen2 aportan EPC (Electronic Product Code) de 256 bits, mientras que la Gen1 soportaba hasta 96 bits.
- La Gen2 incluye un método para soportar múltiples lectores y reducir la interferencia entre ellos (Dense-Interrogator channelized signaling). Este modo se utiliza en zonas donde múltiples lectores funcionan al mismo tiempo. Es importante saber que este modo es opcional para los lectores, según la especificación. El comportamiento en el entorno real depende de muchos factores, incluyendo interferencias externas de otros dispositivos, como teléfonos inalámbricos UHF, equipamiento industrial o equipos inalámbricos de redes LAN.
- Cabe resaltar, que solo el estándar ISO 18000-6, trata con la tecnología RFID en el rango UHF. Los estándares ISO, difieren del estándar EPC, en que el estándar ISO apunta únicamente al interfaz aéreo (es decir, a como los tags y los lectores se comunican entre sí); mientras que el estándar EPC apunta a otros componentes del sistema más allá de la interfaz aérea. El nuevo protocolo EPC Generation 2.0 recomienda un mecanismo basado en Aloha, en concreto el protocolo Slotted Aloha o Aloha ranurado.

1.6 ALTERNATIVAS DE MERCADO

La tecnología de códigos de barra, tiene todavía uso en muchos países, El Salvador no es la excepción, y como se sabe, en estos sistemas, se necesita que el usuario este interactuando con el producto u objeto a controlar, para capturar o leer el código tiene que estar a pocos centímetros del producto, como se observa en la figura 1.24:



Figura 1.24: Lectura de un código de barras.

Teniendo un sistema deficiente, ya que tiene la posibilidad de no escanear todos los códigos, ya sea por la cantidad de productos o se encuentra en una localización que es difícil de acceder, por parte del usuario.

Anteriormente en el país la tecnología RFID, se utilizaba solamente como un tipo de sistema de seguridad para algunos almacenes, este sistema se basa en colocar una pinza con un chip RFID en el elemento a controlar, por ejemplo algunos almacenes de ropa se la colocan a las prendas como se ve en la figura 1.25, el número de identificación de dichas pinzas se encuentra registrado en una base de datos, si la pinzas o el elemento pasa entre dos pequeñas torres colocadas en paralelo en cada entrada o/y salida del almacén como se observa en la figura 1.26, estas torres poseen antenas, creando un tipo de barrera invisible al ojo humano, y si una pinza pasa por esta barrera, el sistema la detectara y ara sonar una alarma audible, para que los vigilantes puedan detener dicho robo.



Figura 1.25: Pinzas antirrobo con chip RFID.



Figura 1.26: Torres de antenas antirrobo para centros comerciales que utilizan tecnología RFID.

1.7 SITUACIÓN ACTUAL EN EL SALVADOR Y PREVISIONES FUTURAS

Las tecnologías de identificación por radiofrecuencia, son recientemente cuando están teniendo una mayor aplicación y con mayor diversidad sectorial. En los últimos años las tecnologías RFID se han desarrollado y perfeccionado técnicamente, disponiendo actualmente de estándares internacionalmente aceptados para las bandas de frecuencia de trabajo más habituales, con mayor número de aplicaciones, y con la aceptación de las Administraciones responsables de la asignación

de frecuencias, que permitan el desarrollo de las tecnologías RFID, pensando además en que los recursos (por ejemplo las bandas de frecuencia) deben ser compatibles para el uso de la tecnología a nivel internacional, punto imprescindible para que se produzca el uso masivo de la tecnología considerando la globalización de productos que existen en la actualidad.

En El Salvador las empresas de telefonía celular que operan en las bandas de frecuencia de 800 MHz son Telefónica (actualmente Movistar) y Telemóvil (actualmente Tigo) y en la banda de 900 MHz opera la compañía DIGICEL, estas son las bandas de frecuencia con las que puede interferir el módulo lector que trabaja en la banda de UHF (850 a 960 MHz).

En la figura 1.20 podemos observar las frecuencias en la banda de UHF ocupadas por estas operadoras en El Salvador, y en la figura 1.21, se muestra un mejor detalle del espectro de frecuencias ocupado en la banda UHF.

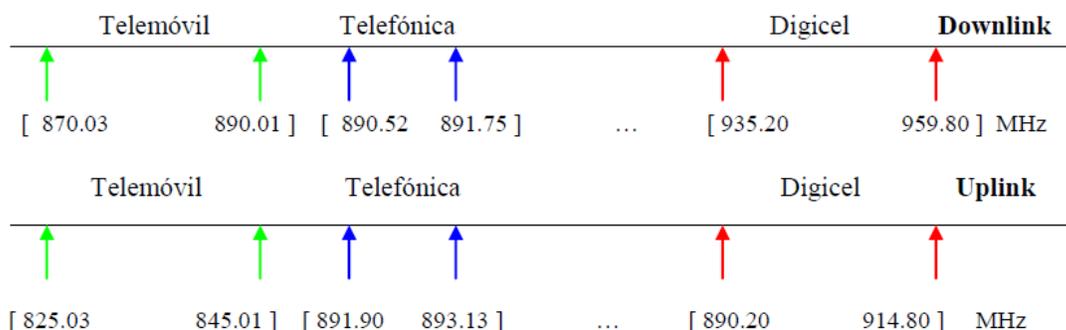


Figura 1.27³³

Sistema	Banda	Frecuencia		Asignación de canal
		Subida (MHz)	Bajada (MHz)	
T-GSM-810	810	806.0 – 821.0	851.0 – 866.0	Dinámica
GSM-850 ³⁴	850	824.0 – 849.0	869.0 – 894.0	128 – 251
P-GSM-900 ³⁵	900	890.2 – 914.8	935.2 – 959.8	1 – 124
E-GSM-900	900	880.0 - 914.8	925.0 – 959.8	975-1023, 0-124
R-GSM-900	900	876.0 – 914.8	921.0 – 959.8	955-1023, 0-124
T-GSM-900	900	870.4 – 876.0	915.4 – 921.0	Dinámica

Tabla 1.8³⁶

³³ Según Tesis: Medición de niveles de campo electromagnético producidos por los sistemas de telefonía móvil en El Salvador y su estado con respecto a normas internacionales de seguridad (UES, 2003).

³⁴ Éste término es usado para designar cualquier sistema que opere en cualquier banda de 850 MHz, excluyendo a T-GSM-810. GSM 850 es llamada, a veces, GSM-800 porque este rango de frecuencias fue conocido como la "banda de 800 MHz" cuando fue asignada por primera vez para el sistema AMPS en los Estados Unidos en 1983.

³⁵ Banda GSM-900 Extendida

³⁶ Según documento: Banda de frecuencias GSM en El Salvador.

Para el caso específico de la gestión de activos, no existen regulaciones en El Salvador según EPC global en su informe del 4 de Septiembre de 2007, “Estado de Regulación para el uso de RFID en el espectro de UHF”, respecto de la identificación de productos. Por tanto la implementación de este sistema a nivel nacional se encuentra aún virgen respecto de las aplicaciones de gestión de activos³⁷. La tecnología RFID seguirá siendo impulsada por GS1 EL SALVADOR lo cual brinda una excelente fuente de apoyo técnico local para el desarrollo de aplicaciones.

El Salvador no se queda atrás en el auge que ha tenido la tecnología RFID, en los últimos años el uso de esta tecnología ha aumentado exponencialmente, se está utilizando como un sistema de rastreabilidad o control de la ubicación del ganado en el campo, colocando un chip RFID en el arete de identificación visual que se le coloca en la oreja del ganado, eliminado así el peligro que se tenía a la hora de controlar la ubicación, ya que el granjero se tenía que acercarse a la vaca (por ejemplo) para leer su número de identificación, pero en vez de eso, los sistemas RFID le permiten el mismo control pero de forma más eficiente y sin poner en peligro al granjero; algunas empresas ocupan la tecnología RFID como un control de acceso a sus instalaciones, colocando una cerradura electrónica, en sus puertas para abrirlas, cuando un usuario coloca una tarjeta de identificación, con un chip a unos centímetros cerca de una antena o lector, y si el usuario o el número de identificación asociado al chip tiene permisos de ingresar la cerradura abrirá la puerta y dejara ingresar a la persona.

Uno de los usos más memorables de la tecnología RFID en el país, aunque no sea muy reciente, es la implementación de dicha tecnología, en las elecciones de presidente de la república de El Salvador, el Tribunal Supremo Electoral del país, requería una solución al controlar las cajas electorales, dichas cajas contienen distintos elementos, una computadora, una impresora, actas electorales, papel, etc., en total abarcaba aproximadamente 20 elementos distintos, individualmente identificada por etiquetas de códigos de barra, en su totalidad eran aproximadamente 1700 cajas electorales, tomado en cuenta que se colocaban una caja por cada sitio electoral en todo el país; además se necesitaba una persona con un escáner en mano, para leer los códigos de barra, de cada objeto, contenido en la caja electoral, le tomaría aproximadamente cinco minutos por caja, esto era un aproximado de 141 horas que equivalen a 11 días en jornadas de 12 horas, para realizar las verificaciones de las cajas, antes de ser enviadas a los sitios de destino, el problema era que solo se tenían dos días para realizarlo puesto que el tiempo de prueba y la entrega de las cajas hacia los distintos departamentos del país, por lo tanto se necesitaban incrementar la cantidad de personal y equipo necesario para la lectura y registros de los códigos de barra de los elementos contenidos en la caja y aun así se tomaría aproximadamente tres días para realizar la labor.

A través de antenas de RFID en los puestos de trabajo, y el cambio de las etiquetas de código de barras por las etiquetas RFID, el tiempo de registros de los 20 elementos en cada caja disminuyó considerablemente en 30 segundos, ya que esta tecnología realiza dicha identificación por radio frecuencia, teniendo un control más eficiente de los artículos presentes y los que faltaban en cada caja, reduciendo así el margen de error, ya que era muy importante que cada caja llegara a su destino con todos los elementos presentes; disminuyendo el personal necesario para realizar el registro de los elementos, de 50 personas necesarias con la tecnología de código de barras a necesitar 10 personas, con la tecnología RFID, ahorrando así también el tiempo de realización de proceso de verificación del contenido de las cajas a dos días de 12 horas, con un grado de certidumbre del 99%; así también se le adhirió una etiqueta RFID a la caja electoral para tener el

³⁷ Según tesis investigativa sobre RFID (Universidad Don Bosco)

control o registro de la ruta de entrega que le correspondía, si la caja no pertenecía a esa localidad el sistema mostraba una alerta, de que dicha caja no pertenecía, a esa ruta o si la cantidad de cajas cumplía la cantidad total requerida por la localidad.

En el futuro se espera un auge en el uso de la tecnología RFID en El Salvador, se espera que más y más empresas tanto públicas y privadas utilicen estos sistemas, para el control de acceso a cada parte de sus instalaciones, que más empresas realicen sus inventarios, tanto de producción, como también de control de ventas por medio de escaneos a radio frecuencia, también se espera en el control y rastreo o seguimiento en tiempo real de mercancía desde el almacén de materia prima, hasta el punto de venta del producto; utilizar los sistemas RFID como sistemas de seguridad para el control de la localización de los elementos apreciados por el usuario, así como también de la ejecución de alarmas que avisen al usuario cuando se extrajo algún elemento sin permisos.

Se espera que el departamento vehicular en el país introduzca tecnología RFID en los vehículos que entran del extranjero, para evitar fraudes, así también previniendo la comercialización de vehículos robados, ya que en el vehículo, se tendría un número de identificación único gracias al chip RFID, también se piensa en agregar chip en las tarjetas de licencias de conducir para evitar falsificaciones y tener un mejor control en los registros de los vehículos de las personas.

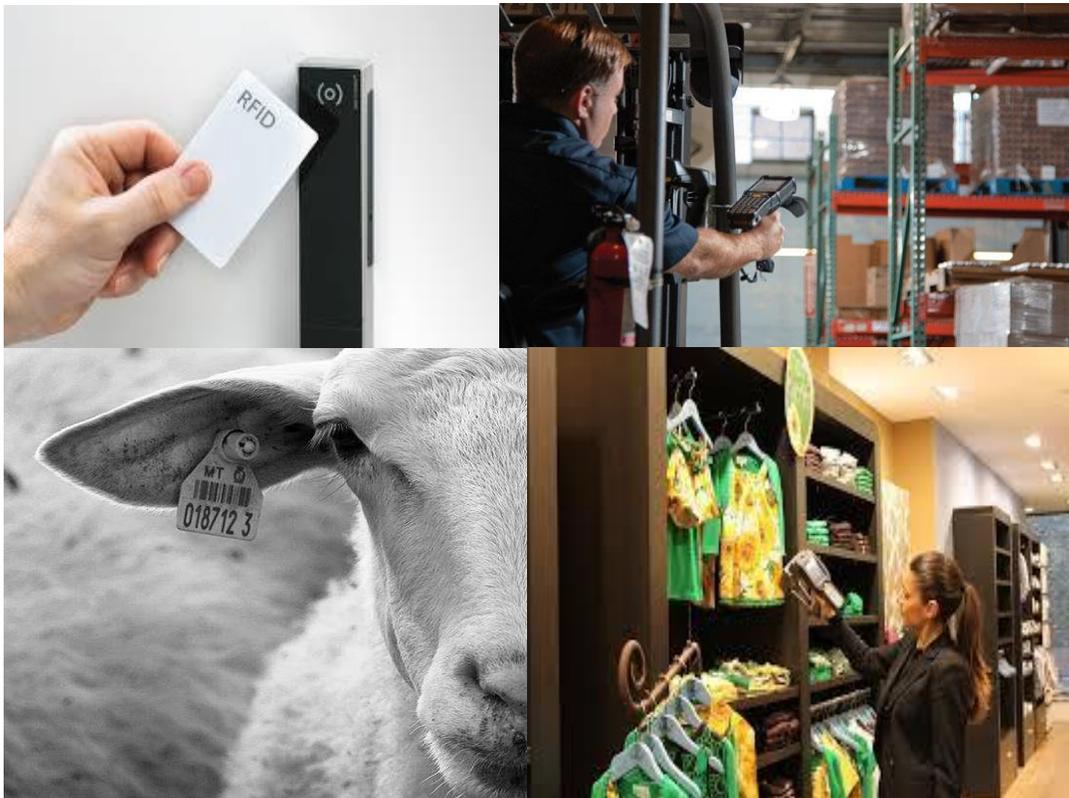


Figura 1.28: Diferentes aplicaciones en las que se espera gane terreno la tecnología RFID en El Salvador.

Existe una economía que ciertamente muestra un crecimiento lento en el país pero la inversión extranjera podrá favorecer la incorporación de nuevas tecnologías, lo que podrían incrementar las importaciones de componentes de hardware en el país, los cuales podrían constituirse en insumos

críticos que afecten el negocio, sin embargo en la medida en este se implemente de manera masiva tal y como el código de barras, el costo del hardware debería tender a la baja³⁸.

Muchas más aplicaciones en El Salvador de la tecnología RFID, son esperadas para las diferentes ramas, tanto públicas como privadas, visualizando que en el futuro, todos los productos o elementos de los hogares contengan chip, desde las cosas más complejas como televisores, refrigeradoras; así como también, de cosas menos complejas como envases de plástico, zapatos, etc.

1.8 JUSTIFICACION

El desarrollo del presente trabajo, permitirá proponer una solución alternativa para los proceso de control de movimiento de equipos, dicha solución es un sistema prototipo basado en la tecnología RFID, dicha tecnología ofrece una propuesta de sistema prototipo de localización de los elementos a controlar, en tiempo real, además permite monitorear cualquier parámetro adicional referente al elemento al cual esta adherido una etiqueta RFID, dicha etiqueta le permite tener al elemento un número de identificación único, parecido al número de inventario o al numero de un código de barras, por ende el sistema prototipo es capaz de identificar a dicho elemento individualmente, controlando así la existencia del mismo y si no ha salido del rango de control sin permiso.

Gracias a este sistema prototipo se puede llegar a tener un control más detallado y preciso de la cantidad y ubicación de los elementos en el espacio o lugar a controlar, además se tiene un sistema prototipo capaz de llevar un inventario de los elementos en forma documental más precisa y ordenada, ya que anteriormente los inventarios se realizaban en forma manual y visual por el usuario, así también se tendrá un control más preciso de los préstamos de dichos elementos.

Creando así una solución eficiente a la hora de tener un control del inventario de los elementos, ya que dicho sistema prototipo se comunica con las etiquetas adheridas a los elementos a controlar por medio de pulsos de radio frecuencias, sin el miedo de dañar el funcionamiento o físicamente al elemento, y sin la necesidad que el usuario necesite tener a la vista el número de identificación y anotarlo en su registro manualmente, esta acción lo realiza el sistema prototipo automáticamente, reduciendo el tiempo de implementación y la mano de obra.

1.9 LIMITANTES Y ALTERNATIVAS PARA SUPERARLAS

Algunas limitaciones que se encontraron a la hora de la realización del presente trabajo, primero el aumentar el rango o distancia de lectura con la construcción de algunas antenas en la frecuencia de operación de UHF con polarización circular, con el poco conocimiento que se poseía de las mismas, ya que el sistema prototipo fuera aumentado su eficiencia a la hora de realizar una inventario si se aumentara la distancia de lectura.

Se detectó la limitante de la interferencia de la banda de telefonía celular, ya que al realizar pruebas de alcance de lectura del módulo RFID en una habitación determinada pero con un número de personas presentes se alcanzó una distancia de 1.7m, ya que se detectó que cada persona poseía un teléfono celular, y por ende dichos teléfonos generaban interferencia en el momento en que el modulo realizaba los escaneos o lecturas de las etiquetas RFID, luego se realizó la misma prueba pero solamente con el usuario el cual estaba operando el sistema y su teléfono celular apagado se logró una distancia de 2.4m, por lo que el sistema prototipo es más eficiente el lugar esta solo.

³⁸ Según Tesis investigativa de universidad Don bosco (El Salvador).

Otra limitante fue el aspecto económico, ya que se podría haber adquirido elementos capaces de eliminar algunas interferencias o ayudar al módulo RFID a generar un pulso de mayor potencia, ya que se utilizaron etiquetas pasivas.

Una de las alternativas o soluciones que se analizaron fue el estudio de la implementación del prototipo del sistema RFID en la banda de microondas (2.5GHz) ya que en esta banda de frecuencias el rango de lectura es mayor y la realización o construcción de las antenas es sencilla. Así analizar la propuesta de realizar o construir un arreglo de antenas para alcanzar una mayor distancia de lectura, así como también colocar varias antenas con gran ganancia en lugares estratégicos de la habitación para abarcar completamente cada rincón de la misma, creando así un sistema prototipo muy eficiente.

Otra alternativa es la utilización de etiquetas activas, ya que estas su fuente de poder no depende de la potencia del pulso generado por el lector RFID, teniendo en mente que esta solución es más eficiente cuando se aplica para sistemas de RFID en la banda de frecuencias de microondas, ya que en la banda de frecuencias UHF las etiquetas activas tienen un costo elevado.

CAPÍTULO 2: Hardware-Software: selección y pruebas.

INTRODUCCIÓN.

En el presente capítulo se abordará el funcionamiento de los sub módulos que conforman el sistema o prototipo propuesto, siendo sus dos bloques principales el hardware y el software asociado a dicho equipo.

En el desarrollo de cada uno de los bloques anteriormente mencionados, se tomaron criterios de selección que ayudaron a depurar y mejorar ciertas variables que afectan el correcto funcionamiento del sistema; todo esto apoyado con pruebas de laboratorio realizadas en los laboratorios de la escuela de ingeniería eléctrica de la FIA.

Parte de hardware utilizado, fue construido localmente, mientras que otros elementos fueron adquisiciones de mercado, así mismo fue necesario utilizar dispositivos de medición improvisados dada la carencia de equipos idóneos en los laboratorios de la EIE. Por otra parte, el sistema está compuesto por módulos que dan la libertad de poder programarlo a la conveniencia del usuario es por eso que el software del sistema está desarrollado para las necesidades del usuario.

2.1 ESTUDIO DE COMPONENTES ASOCIADOS A TECNOLOGÍA RFID.

En esta sección se presentarán los resultados de las pruebas realizadas en laboratorio, que permitieron la selección de los diversos componentes asociados a la solución sugerida en el presente trabajo; lo cual incluye antenas, módulo lector, etiquetas y el desarrollo del software correspondiente. Todo lo anterior dio como resultado la propuesta de un prototipo para el control de movimientos de elementos o dispositivos en los laboratorios de la Escuela de Ingeniería Eléctrica de la FIA. Las opciones exploradas operan en el rango de frecuencias de UHF (868-930 MHz), dado que dicho rango es uno de los más utilizado en este tipo de aplicaciones. A nivel de diagrama de bloques, el sistema propuesto es el mostrado en la figura 2.1.

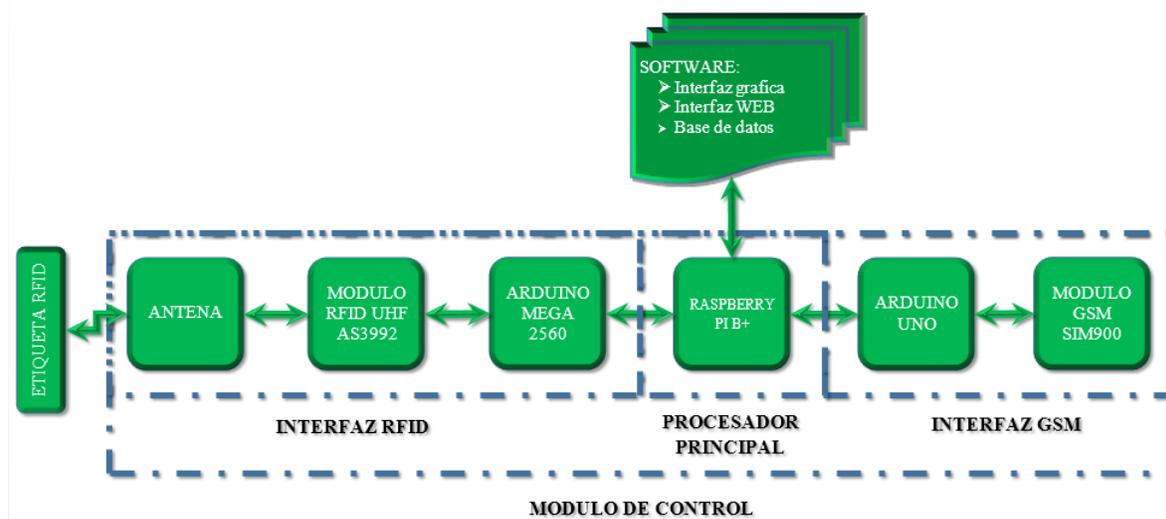


Figura 2.1: Diagrama de bloques del sistema prototipo de seguridad utilizando la tecnología RFID.

De la figura anterior, el módulo de control produce un pulso de radiofrecuencia, el cual alcanza a una etiqueta o tag a partir de lo cual se establece el intercambio de información.

Gracias a este intercambio de información entre el módulo lector y las etiquetas, y que esta comunicación es en forma inalámbrica, sin dañar o interferir con el funcionamiento de los elementos a los cuales está adherida, se es capaz de consolidar este sistema como una solución a las dificultades de seguridad o control de los elementos en una zona específica; una descripción básica del funcionamiento del sistema prototipo de seguridad RFID es:

El sistema tiene la capacidad de realizar escaneos a una frecuencia fija en el rango de 800-990 MHz (en la banda UHF), se le adhiere una etiqueta al elemento, luego se ingresa la información pertinente en la base de datos, cuando el usuario ejecute un inventario se producirá un pulso de radiofrecuencia, y cuando es alcanzado por las etiquetas responderán con su número de identificación o número EPC del elemento, luego el procesador principal interrogará a la base de datos por la existencia de cada uno, este comparará cada número EPC proveniente de la lectura del módulo con la información contenida en la base de datos, cuando termine de comparar y se detecte que falte algún artículo, el sistema mandará una orden a la interfaz GSM de mandar mensajes de textos de alarmas al usuario conteniendo los números ID de los elemento no detectadas por el lector.

Cuyas características más importantes y pruebas desarrolladas se explican a continuación:

2.2 HARDWARE.

Módulo de Control: Se le llama así a la totalidad del hardware utilizado, mismo que está conformado por diferentes tarjetas, cada una con una función específica dentro de la solución propuesta; éste módulo está conformado por:

- Procesador principal
- Interface RFID
- Interface GSM
- Antena
- Y de manera remota las etiquetas

2.2.1 Procesador principal.

Es el cerebro del sistema, donde se llevan a cabo todas las funciones y/o órdenes del módulo completo; acá también se almacenan las bases de datos utilizadas; físicamente es una Raspberry pi B+. Los comandos ejecutados por el usuario a través de la interfaz gráfica respectiva, serán procesados con esta tarjeta.

La figura 2.2 muestra el módulo mencionado anteriormente.

Tarjeta Raspberry pi B+

En la actualidad existen varios modelos tarjetas de este tipo, las que permiten el desarrollo de proyectos de aplicación tecnológicas. La Raspberry pi fue (inicialmente) desarrollada en Inglaterra por una fundación que lleva el mismo nombre; básicamente un ordenador creado con el objetivo de estimular la enseñanza de ciencias computacionales en las escuelas.

La tarjeta Raspberry Pi Modelo B+, es uno de los modelos más recientes y ofrece una serie de mejoras y ventajosas características, como por ejemplo mejora el consumo de energía, amplía las opciones de conectividad, mayor velocidad de procesamiento, etc.

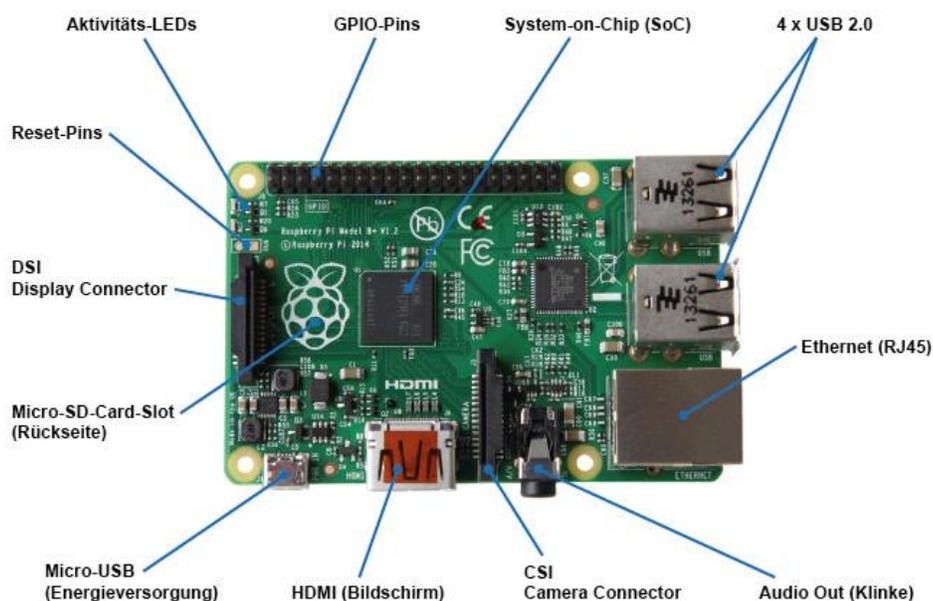


Figura 2.2: Tarjeta Raspberry pi B+

Las características técnicas de dicha tarjeta son descritas en la tabla 2.1:

Chip	Broadcom BCM2835 SoC
Arquitectura Core	ARM11
CPU	700 MHz de baja potencia ARM1176JZFS Procesador de aplicaciones
GPU	Dual Core VideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memoria	512MB SDRAM
Sistema de operación	Ranura de tarjeta Micro SD , que ejecutan una versión del sistema operativo Linux
Dimensiones	85 x 56 x 17mm
Poder	Conector micro USB 5V, 2A
Conectores:	
Ethernet	Socket ethernet 10/100 BaseT
Salida de video	HDMI (rev 1.3 & 1.4) Compuesto RCA (PAL and NTSC)
Salida de audio	3.5mm jack, HDMI
USB	4 conectores USB 2.0
Conectores GPIO	40-pin 2.54 mm (100 mil) header de expansion: 2x20 strip Providing 27 pines GPIO así como las líneas de suministro de +3.3 V, +5 V y GND
Conector de cámara	15-pin MIPI Camera Serial Interface (CSI-2)
Conector Display	Display Serial Interface (DSI) 15 conector del cable flex de forma plana con dos pines de datos y un pin de reloj

Tarjeta de memoria Slot	SDIO
--------------------------------	------

Tabla 2.1: Características técnicas de la Raspberry pi B+.

La elección de dicha tarjeta fue la reducción del sistema, ya que se necesitaba un host que abarcara una base de datos, diferentes tipos de servidores y capaz de generar interfaces gráficas para los usuarios, la primera opción era una computadora personal, pero las ubicaciones en que se pondría el sistema eran limitadas gracias al tamaño de la misma, siendo esta una ventaja al utilizar una Raspberry, ya que su pequeño tamaño otorga la habilidad de la fácil movilidad del sistema y así ampliar los lugares en que este se colocara.

Teniendo así un procesador principal capaz de contener los diferentes tipos de servidores, bases de datos para procesar la información necesaria e interfaces amigables al usuario para otorgarle al mismo un amplio manejo de las funciones del sistema, también la tarjeta posee un tipo de conexión Ethernet el cual se utiliza para la ejecución de una interfaz WEB, reduciendo el tamaño total de todo el sistema.

2.2.2 Interface RFID.

Es un módulo transmisor/receptor, (y de escritura-lectura) que trabaja en la banda UHF. Su función principal es permitir la transferencia de información con las etiquetas asociadas a los equipos a monitorear, es decir que se encarga de generar los escaneos a una frecuencia específica, al detectar la respuesta de las etiquetas, este se lo comunica al procesador principal, dichas respuestas son (generalmente) los números EPC de las etiquetas; también es capaz de modificar dicho número en los casos que sea posible. El procesador principal tiene la capacidad de modificar las funciones de operación de la interface RFID, físicamente consta de una placa Arduino Mega 2560 y un módulo lector RFID AS3992, el arduino mega tiene la función de traducir las órdenes del procesador principal a una forma que el lector AS3992 comprenda y pueda ser ejecutada, y al contrario, también tiene la función de traducir la información que proviene del módulo AS3992 hacia la procesador central, en una forma que este comprenda y pueda procesar la información.

La figura 2.3 muestra el módulo mencionado anteriormente.

Arduino Mega 2560.

Es una placa electrónica basada en el microchip ATmega2560 cuenta con aproximadamente 54 pines de comunicación de entrada y salida de los cuales tiene la posibilidad de utilizarse como salidas PWM (Modulación por Ancho de Pulso), además de 16 entradas análogas y 4 UARTS, posee mayor memoria que el arduino uno, posee 256kB así teniendo la posibilidad de contener mayor cantidad de líneas de comando o código, poseyendo así un sistema para realizar aplicaciones más amplias y complejas, así mismo una conexión USB para poder programar el microcontrolador y darle órdenes u otorgarle las funciones necesarias al sistema a realizar.

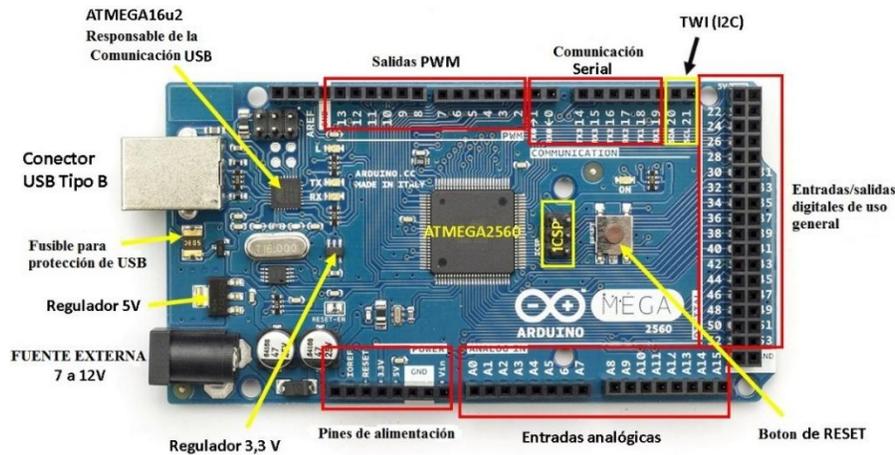


Figura 2.3: Arduino Mega 2560

En la tabla 2.2 se muestran las características técnicas del Arduino Mega 2560.

Microcontrolador	ATMEGA2560
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	(7-12V)
Voltaje de entrada (limites)	(6-20V)
Pines digitales E/S	54 (de los cuales 15 proporcionan salida PWM)
Pines de entrada analógica	16
Corriente continua para pines de E/S	40 mA
Corriente continua para pines de 3.3V	50 mA
Memoria Flash	256 KB, 8 KB utilizado por el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
Velocidad del reloj	16 MHz

Tabla 2.2: Características técnicas de Arduino Mega 2560

La tarjeta arduino se eligió como un puente de comunicación entre el procesador principal y la tarjeta RFID AS3992, el modelo Mega 2560 posee la suficiente memoria interna para contener el código de configuración para la realización de las funciones del módulo lector, transformando órdenes sencillas provenientes del procesador central, en un lenguaje que el lector RFID pueda comprender para realizar las diferentes funciones del módulo de control.

Módulo Lector AS3992 RFID UHF, Tags.

Es un módulo transmisor / receptor, (conocido como UHF RFID Reader System ‘Sistema Lector RFID UHF’, de escritura y lectura) que trabaja en el rango de frecuencia UHF. Está diseñado para la detección de etiqueta de rango medio, (midiendo aproximadamente 9 cm), soporta el uso de los protocolos EPC Gen2. La conexión a una antena externa (como por ejemplo la que se observa en la figura 2.5) se establece mediante un conector MMCX. Se utiliza alimentación auxiliar para energizar, también (dependiendo del tipo de firmware instalado) soporta dos versiones de interfaz

de comunicación una es a través de los pines UART (transmisor/receptor asíncrono universal) y la otra por puerto USB para poder comunicarse y/o indicarle las funciones a realizar.

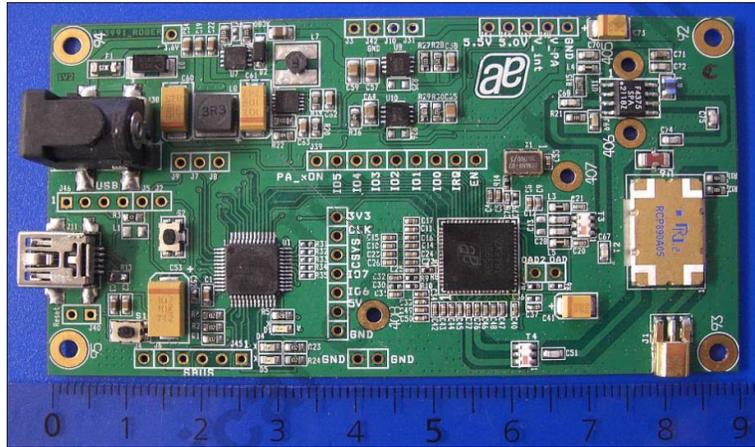


Figura 2.4: Módulo Lector AS3992 RFID UHF.



Figura 2.5: Antena de 5dBi PCB UHF RFID 902 - 928 MHz, 5 cm X 5 cm.

En caso normal, las etiquetas RFID UHF necesitan menos energía que el lector, es por eso que este tiene que tener una alta sensibilidad de recepción. En el sistema, la vía de transmisión y recepción son independientes entre sí en el lector.

Técnicamente, las numerosas aplicaciones pueden elegir diferente potencia de transmisión. Sin embargo, hay normas que se deben cumplir. Por lo general, la potencia de RF en el rango de 100mW ~ 500mW es adecuado para cada tipo de distancia; gracias a esto posee muchas ventajas, tales como: la compatibilidad con diferentes protocolos, alta velocidad de lectura de múltiples etiquetas, su sistema es amigable para el desarrollo de aplicaciones basado en el.

Este lector RFID UHF utiliza la energía de RF para leer datos de una etiqueta, como por ejemplo se observa en la figura 2.6. Se debe procurar no adherirse a una superficie metálica, ya que en dicha superficie es muy probable las ondas de radiofrecuencias sean absorbidas, por lo cual se acorta la distancia a la cual es leída, así como también tener cuidado cuando se coloque en elementos como motores, transformadores, etc., ya que son capaces de producir efectos negativos al funcionamiento del lector.



Figura 2.6: Etiqueta UHF RFID adhesiva, pasiva.

En la tabla 2.3 se muestran las características técnicas del módulo lector AS3992 UHF RFID.

Frecuencia de operación	GB (920-925 Mhz) estándar americano (902-928) estándar europeo (865-868 Mhz)
Protocolo	ISO 18000-6C (EPC GEN2), compatible con norma ISO18000-6A / B
Antena	8 dBi antena direccional o de cerámica
Potencia de RF	Max 20 dBm
Rango de lectura	1 a 6 m (para 8 dBi Antena direccional)
Tiempo de lectura	< 6 ms (multi-tag 64 identificación de números)
Modulación	ASK o PR-ASK
Antena Interfaz	MMCX
Interfaz de alta velocidad	del USB, TTL UART comunicación con PC software de aplicación
Voltaje de funcionamiento	DC 3,3 v
Consumo de energía	1.5w – 2w
Temperatura de funcionamiento	-20 ° C - + 80 ° C
Temperatura de almacenamiento	-40 ° C 125 ° C
Humedad	20 % a 95 % (sin condensación)
Dimensiones	90mmx49mmx0,51mm
Antena tamaño	8"x12"x2"

Tabla 2.3: Características técnicas del módulo RFID UHF AS3992.

La elección de la tarjeta RFID se basa en su funcionalidad, ya que sus características permiten la configuración total de sus funciones, otorgándole al sistema la libertad de escoger los procesos a realizar y modificar sus propiedades acordes a las necesidades del operador, como por ejemplo la frecuencia y/o la potencia en que se realizarán los escaneos.

2.2.3 Interface GSM:

La interfaz GSM, es un módulo capaz de operar en las cuatro bandas de comunicación celular o bandas GSM, la función de este es generar mensajes de texto de alarma hacia el teléfono celular del usuario, el mensaje contiene los número EPC de las etiquetas que no fueron detectadas por el escaneo de la interfaz RFID y que se encuentran en la base de datos del procesador principal, la banda celular en la que operará dependerá de la tarjeta SIM de telefonía móvil que le inserte el

operador, físicamente este módulo se compone de una placa arduino shield GSM SIM900 y una Arduino UNO, la función de la tarjeta arduino uno es de traducir las órdenes que provienen del procesador principal, para que el módulo shield GSM sea capaz de mandar los mensajes de texto de alarma hacia el usuario, y viceversa, también tiene la función de traducir órdenes provenientes del shield GSM SIM900, los cuales son sobre el funcionamiento del mismo.

Arduino UNO:

Es una placa electrónica basada en el micro chip ATmega328, con aproximadamente catorce pines de entrada y salida de los cuales seis tienen la posibilidad de utilizarse como salida PWM, dichas entradas tiene la capacidad de transmitir y recibir señales digitales de 0 a 5V, también posee seis entradas tipo analógicas con un cristal oscilador de 16MHz, este módulo arduino posee un adaptador USB para poder programar el microcontrolador y darle las órdenes o funciones necesarias.

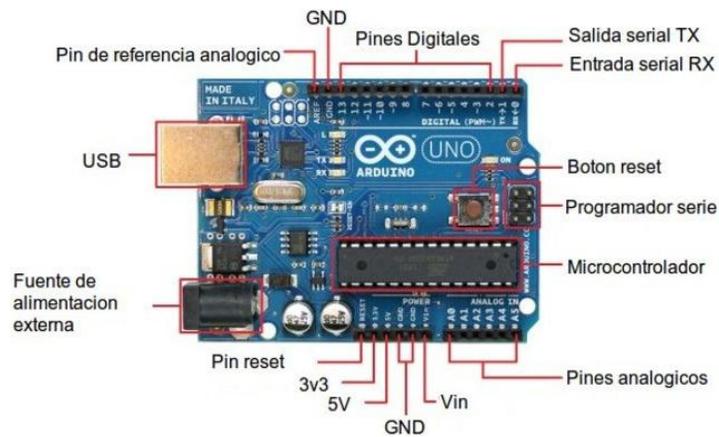


Figura 2.7: Arduino Uno

En la tabla 2.4 se muestran las características técnicas del Arduino Uno.

Microcontrolador	ATmega328
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	(7-12V)
Voltaje de entrada (limites)	(6-20V)
Pines digitales E/S	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	6
Corriente continua para pines de E/S	40 mA
Corriente continua para pines de 3.3V	50 mA
Memoria Flash	32 KB, 0.5 KB utilizado por el gestor de arranque
SRAM	2 KB
EEPROM	1 KB
Velocidad del reloj	16 MHz

Tabla 2.4: Características técnicas del Arduino Uno.

La tarjeta arduino se eligió como un puente de comunicación entre el procesador principal y la módulo GSM, el modelo UNO contiene el código de configuración para la realización de las funciones del shield GSM, traduciendo órdenes sencillas provenientes del procesador central, en un lenguaje que shield GSM pueda comprender, los cuales son las peticiones de enviar los mensajes de alerta del sistema de control.

Shield GPRS/GSM

Un Shield es una placa impresa que conecta en la parte superior de cualquier placa Arduino, para ampliar las capacidades de este; se pueden superponer varios shields. Los shields suelen ser diseños bastante simples y en general de código abierto, aunque pueden haber shields sofisticados.

El shield GPRS/GSM (se observa en la figura 2.8) es una tarjeta GPRS (Servicio general de paquetes vía radio) ultra compacta de comunicación inalámbrica compatible con módulos arduino, la tarjeta está basada en el módulo SIM900, es configurada y controlada día conexión UART usando comando AT, las cuales son instrucciones codificadas que conforman un lenguaje de programación entre el módulo GSM y el host al cual está conectado.

El shield es capaz de trabajar en las cuatro bandas de telefonía móvil (850/900/1800/1900MHz), para especificar una banda específica de funcionamiento, dependerá de la tarjeta SIM (tarjeta inteligente, desmontable utilizada en teléfonos móviles) insertada en el módulo, pueda utilizar la red de telefonía celular GSM para recibir datos desde una ubicación remota, esto lo logra por cualquiera de los 3 métodos:

- Servicio de mensajes de texto cortos
- Audio
- Servicio GPRS

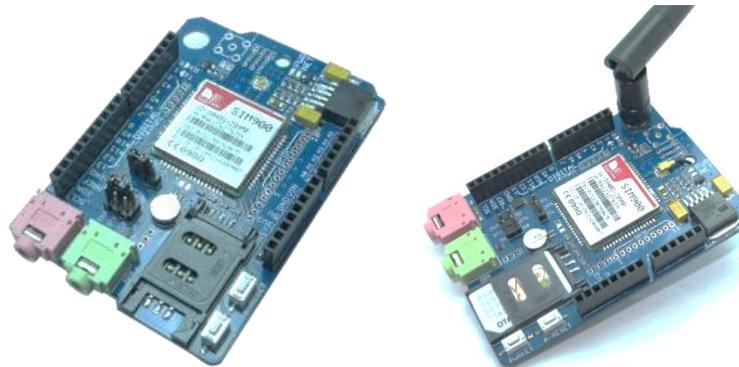


Figura 2.8: Shield GPRS/GSM

El shield GPRS es como un teléfono celular, además de las funciones de comunicación, el GPRS tiene 12 puertos GPIO 2 PWM y un ADC.

Se optó por la elección de esta tarjeta shield arduino GSM ya que trabaja en las cuatro bandas de telefonía celular, otorgándole al usuario la libertad en elegir en que banda operara el modulo, dándole así la habilidad al sistema en enviar alarmas (dichas alertas serán en forma de mensajes de texto) hacia el teléfono celular del usuario, agregando la habilidad al sistema completo de producir

un tipo de alarma remota a grupo de personas, localizados en diferentes lugares, aumentando la eficiencia del sistema de control, dándole una gran ventaja con respecto a otros sistemas similares.

2.2.4 Antenas:

Se ha analizado la conveniencia de utilizar diferentes tipos de antenas para mejorar el alcance entre los dispositivos involucrados en la comunicación inalámbrica, y para este caso en particular, los criterios de selección requeridos para operar de manera óptima con el lector de tags RFID AS3992; todos los diseños deben tener una impedancia de entrada de 50 ohm, para tener un acople perfecto con el lector de RFID; también deberán operar en la banda de UHF específicamente en la banda libre que es de 902 Mhz hasta los 928 Mhz, otra consideración a tener en cuenta es que la polarización deberá ser circular para asegurar que las antenas puedan leer las tags independientemente de la posición en que estas se encuentren ya sea vertical u horizontalmente.

2.2.4.1 Antena Parche Circular.

La antena parche circular mostrada en la figura siguiente; es una implementación de un diseño que utiliza square-ring o anillo cuadrado, con el propósito de reducir las dimensiones de la antena.

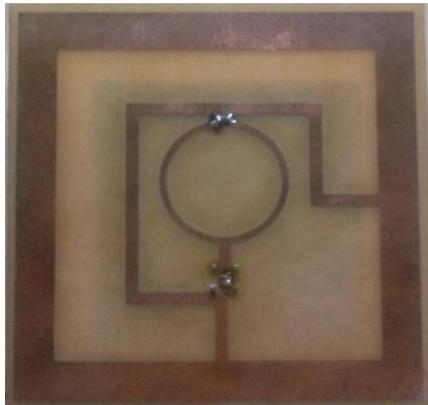


Figura 2.9: Antena parche circular implementada.

- Patrón de radiación.

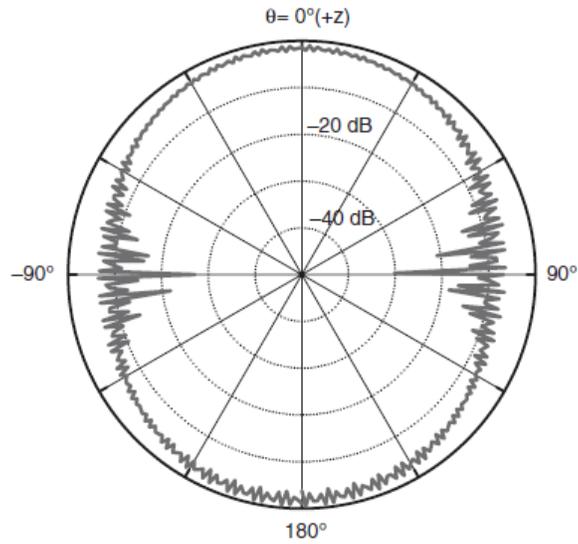


Figura 2.10: Medición de patrón de radiación circular a derechas en el plano y-z a 915 Mhz.

- Ganancia teórica.

La ganancia de la antena según se observa en la figura, tiene un valor aproximado de 2 dBi a la frecuencia de interés

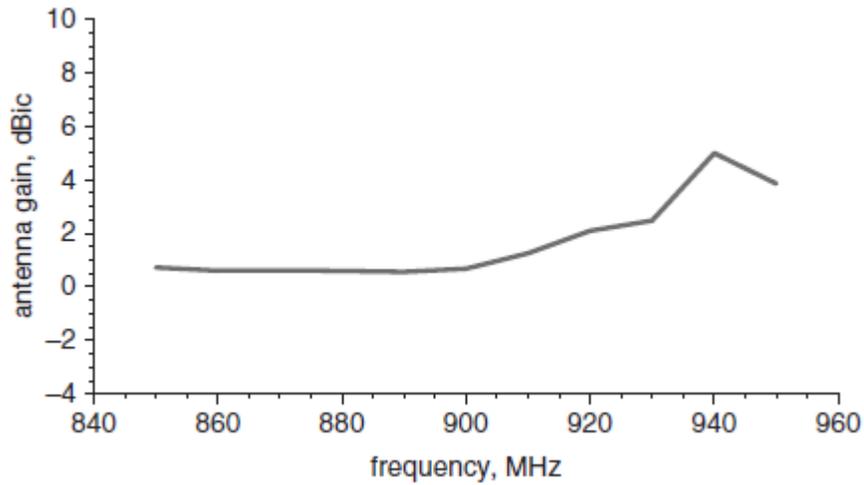


Figura 2.11: Ganancia vrs. Frecuencia.

2.2.4.2 Antena Parche Circular (Arreglo).

El diseño del parche rectangular se realizó utilizando las ecuaciones siguientes:

$$\lambda_0 = \frac{c}{f} \text{ (Longitud de onda)}$$

$$W = \frac{\lambda_0}{2} \sqrt{\frac{2}{\epsilon_r + 1}}$$

Donde ϵ_r es la permitividad del material y ϵ_{ef} es la constante efectiva del dieléctrico, tomando en cuenta que parte del campo viaja por el aire:

$$\epsilon_{ef} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left(1 + 12 \frac{h}{W}\right)^{-\frac{1}{2}}$$

Finalmente, el largo real de construcción del parche este dado por la relación:

$$\Delta L = 0.412h \frac{(\epsilon_{ef} + 0.3) \left(\frac{W}{h} + 0.264\right)}{(\epsilon_{ef} - 0.258) \left(\frac{W}{h} + 0.8\right)}$$

$$L = L_{ef} - 2\Delta L = \frac{\lambda_0}{2\sqrt{\epsilon_{ef}}} - 2\Delta L$$

El arreglo de parches rectangulares se alimentara a través de una línea microstrip. La cual se acopla directamente al parche tal y como se muestra en la figura.

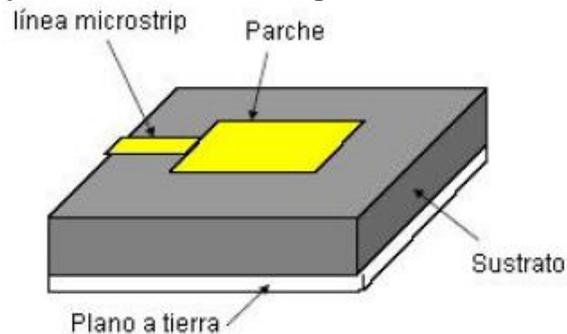


Figura 2.12: Alimentación por línea microstrip..

El cálculo de la línea microstrip se realizó con la ayuda del programa ADS con este software se obtiene las dimensiones físicas del largo y ancho de la tira:

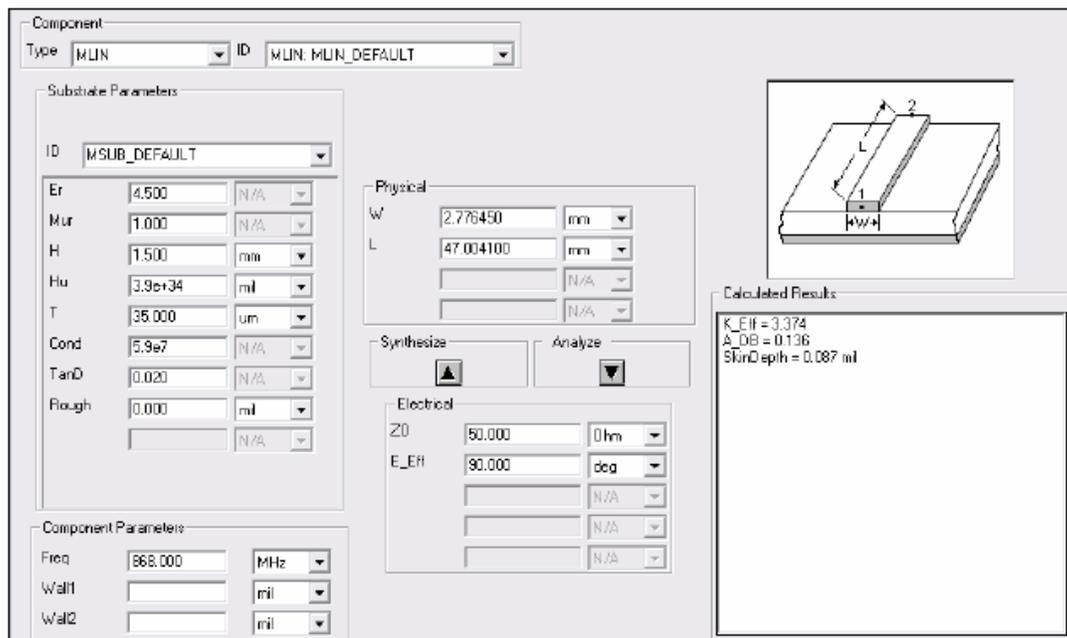


Figura 2.13: Cálculo obtenido con ADS para una línea microstrip de 50 Ω.

- Substrato

Los parches se diseñarán de modo que resuenen en la banda de frecuencias deseada, y la impedancia que presentan a esa frecuencia depende fuertemente del substrato seleccionado.

El substrato que se ha considerado para el diseño es el FR4 con cobertura de cobre en una sola cara. En la siguiente tabla podemos ver sus características:

Substrato	Metal	Constante dieléctrica	Grosor dieléctrico	Tangente de pérdidas	Grosor Metal
FR4	Cobre	4,4	0,02	1,57 mm	34 μm

Tabla 2.5: Características del substrato.

- Cálculos

Con los datos del substrato FR4 mostrados en la tabla anterior se prosigue a realizar los cálculos para una antena parche rectangular con polarización circular e impedancia de entrada de 50 Ω

$$\lambda_0 = \frac{300,000 \frac{m}{s}}{915 \text{ Mhz}} = 328 \text{ mm}$$

$$W = \frac{\lambda_0}{2} \sqrt{\frac{2}{\epsilon_r + 1}} = \frac{324 \text{ mm}}{2} \sqrt{\frac{2}{4.4 + 1}} = 98.6 \text{ mm}$$

$$\epsilon_{ef} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left(1 + 12 \frac{h}{W}\right)^{-\frac{1}{2}} = 4.04$$

$$\Delta L = 0.412h \frac{(\epsilon_{ef} + 0.3) \left(\frac{W}{h} + 0.264\right)}{(\epsilon_{ef} - 0.258) \left(\frac{W}{h} + 0.8\right)} = 0.736mm$$

$$L = \frac{\lambda_0}{2\sqrt{\epsilon_{ef}}} - 2\Delta L = 76.7mm$$

Para obtener polarización circular debemos igualar $L=W$ por lo que las dimensiones del parche serán de 7.7 cm X 7.7 cm. Para obtener una mejor ganancia se diseñara un arreglo de tres parches la separación entre parches será de $\lambda/2$ [1]. En la siguiente figura se muestra el arreglo de parche rectangular.



Figura 2.14: Arreglo de antena parche rectangular.

- Patrón de radiación.

El patrón de radiación teórico de la antena parche con polarización circular se muestra en la siguiente figura

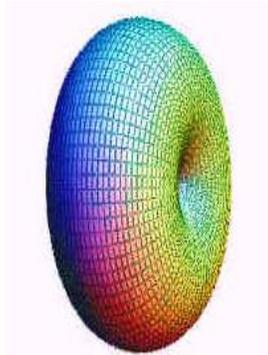


Figura 2.15: Patrón de radiación teórico de antena parche rectangular.

- Simulación

En este apartado se utilizará el software HFSS para simular la antena parche con lo cual obtendremos una aproximación de los parámetros de interés como lo son la ganancia, la impedancia de entrada, el coeficiente de reflexión etc.

La simulación de la antena parche, permite corregir los parámetros antes mencionados, variando las dimensiones del parche para así obtener un diseño que cumpla con las especificaciones técnicas requeridas.

A continuación se presenta la antena parche rectangular simulada:

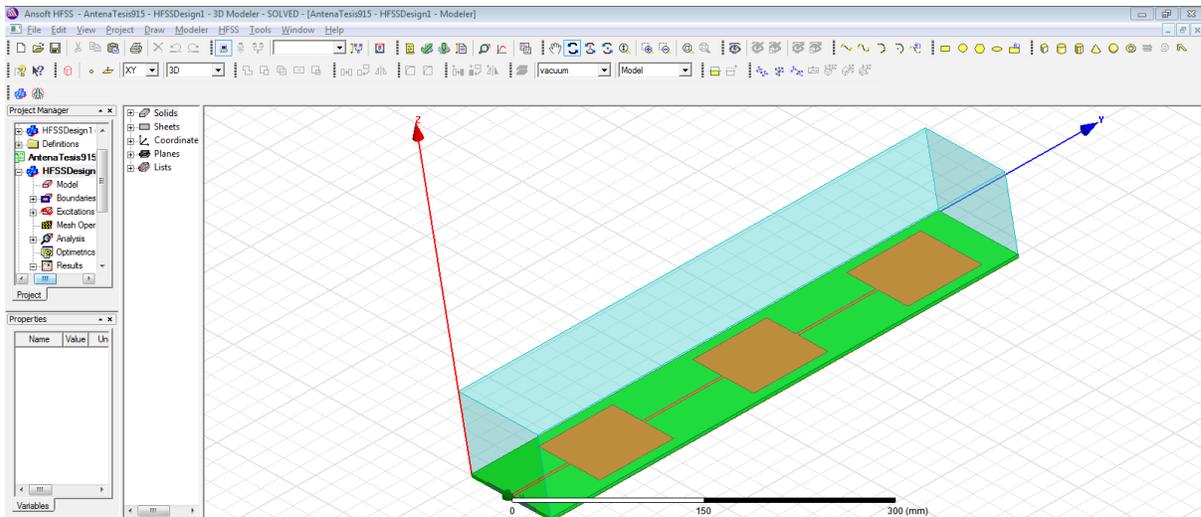


Figura 2.16: Diseño de antena parche rectangular en HFSS.

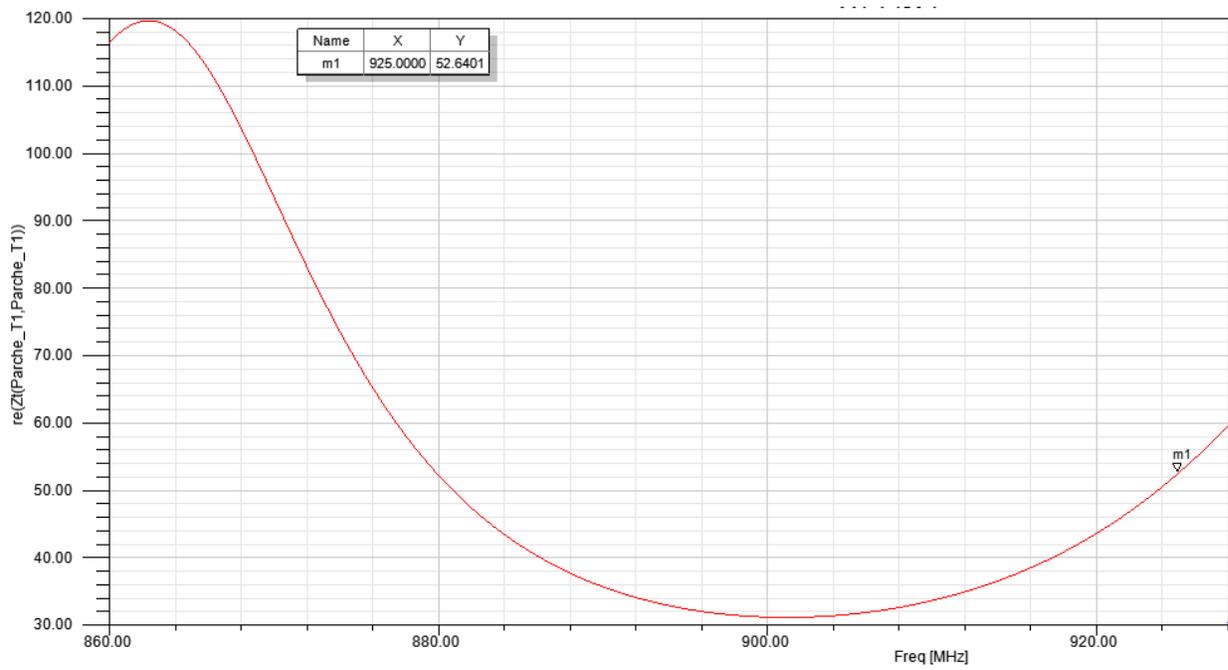


Figura 2.17: Impedancia de entrada de la antena parche a 925 Mhz.

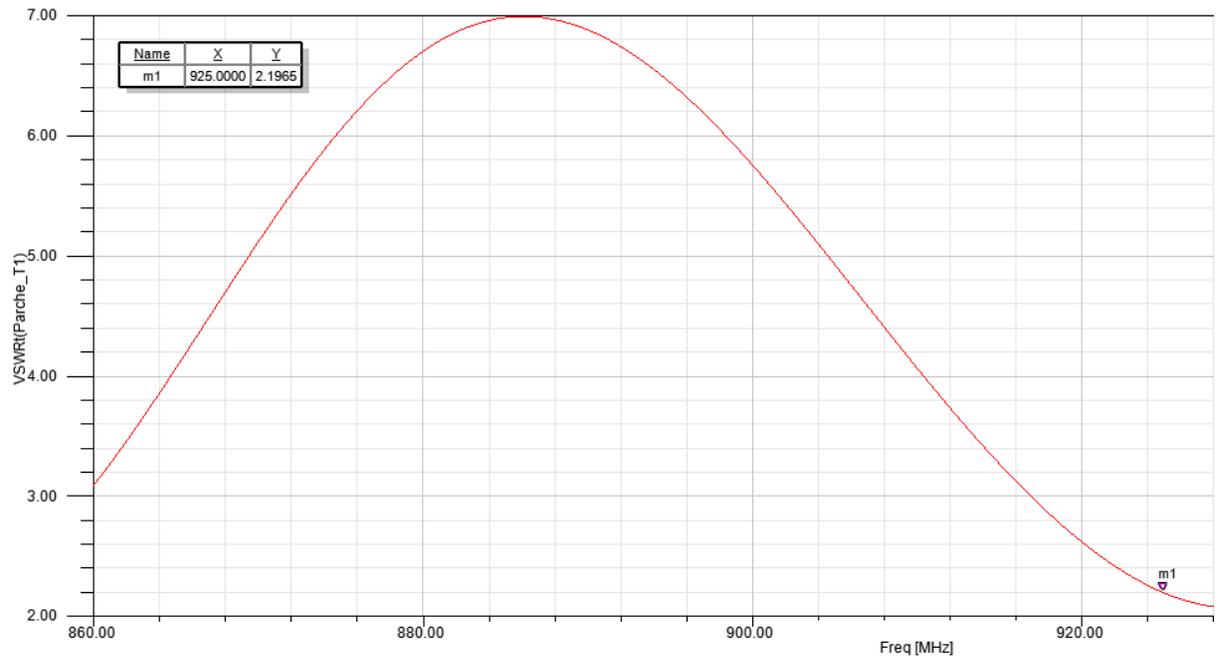


Figura 2.18: Valor del coeficiente de reflexión.

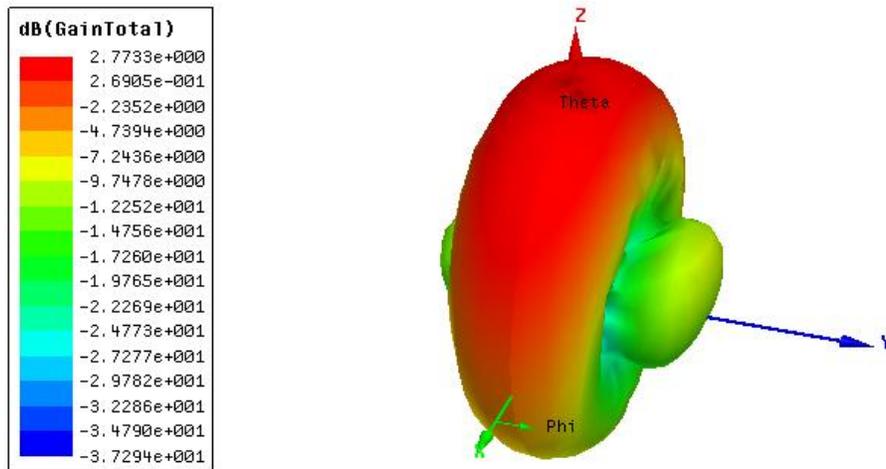


Figura 2.19: Ganancia total de la antena parche.

De acuerdo a los resultados obtenidos en la simulación, la ganancia total está por debajo de lo que se esperaba pero es lo más alto que se pudo obtener al ir variando las dimensiones físicas de la separación de los arreglos. Esta separación además de afectar el valor de la ganancia también varía los valores de impedancia, coeficiente de reflexión de manera que no podemos variar demasiado esta separación porque se afecta el valor de adaptación y de impedancia de entrada de la antena lo cual ocasionaría que no exista un buen acople. También es importante aclarar que aunque se diseñó para una frecuencia de operación de 915Mhz hubo una variación de los valores de impedancia y de la ROE lo cual se obtienen los valores adecuados de estos parámetros en la frecuencia de 925Mhz como se mostró en las figuras anteriores.

2.2.4.3 Antena helicoidal.

Las antenas helicoidales consisten básicamente de un número de espiras conductoras con un diámetro constante sobre un plano de tierra como se observa en la siguiente figura:

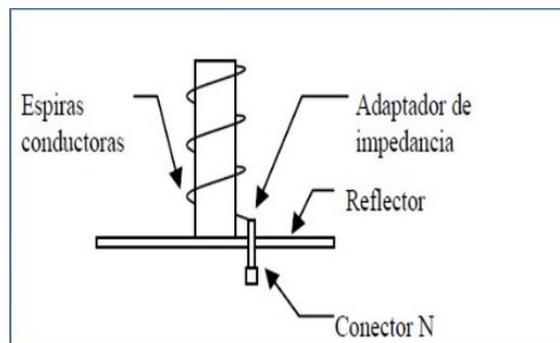


Figura 2.20: Esquema básico de antena helicoidal.

- Diseño de la antena

La geometría de una antena helicoidal está conformada por el diámetro D la separación S entre cada espira el número de vueltas N y el ángulo de separación entre espiras α . La longitud total de la antena es $L=NS$ y la circunferencia de la hélice es $C = \pi D$.

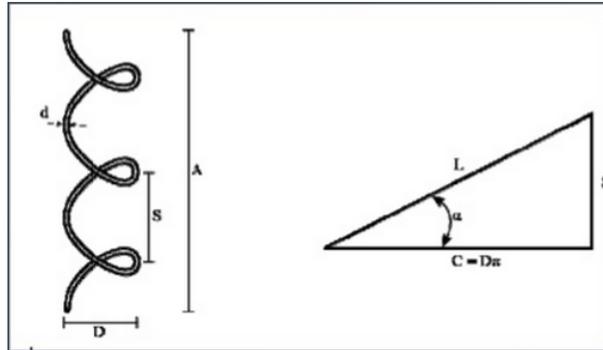


Figura 2.21: Parámetros dimensionales del helicoide.

El ángulo de separación está dado por la siguiente ecuación:

$$\alpha = \tan^{-1}\left(\frac{S}{\pi D}\right) = \tan^{-1}\left(\frac{S}{C}\right)$$

Debido a que la antena helicoidal tiene una impedancia característica que tiene aproximadamente un valor de 140Ω ; se necesita un acoplador de impedancia³⁹ entre el conector de la antena y el elemento radiante que es la helicoide; este acoplador es básicamente un triángulo rectángulo el cual tiene 17mm y 71mm en sus catetos en la siguiente imagen se muestra el adaptador



Figura 2.22: Adaptador de impedancia de antena helicoidal.

- Patrón de radiación.

La forma del patrón de radiación de la antena helicoidal es directivo tal y como puede observarse en la figura este puede modificarse modificando las características físicas de la antena como lo son el número de espiras que al ir aumentando se incrementa su ganancia y directividad.

³⁹ <http://es.slideshare.net/marcoalejandro75470/antena-helicoidal>

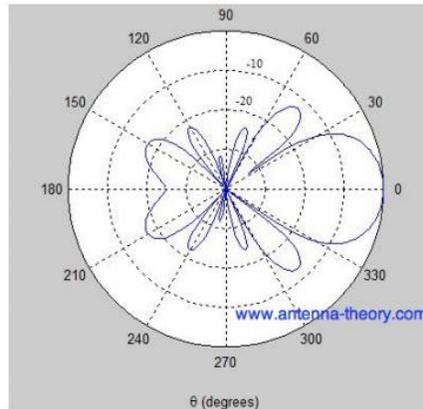


Figura 2.23: Patrón de radiación teórico de antena helicoidal.

- Cálculos

Para la realización de los cálculos se utilizó la ayuda de un sitio web⁴⁰ en el cual solamente se introduce la frecuencia de operación el número de vueltas y el diámetro del conductor para así obtener los parámetros físicos para la construcción de la antena helicoidal que se muestran en la siguiente figura:

Freq. [MHz]	915	
n	4	number of turns
λ [mm]	327.869	wavelength
Gain [dB]	11.8	approx.
l_1 [mm]	81.97	Length of matching Coax ($\lambda/4$)
d_1 [mm]	10.501	Diameter of matching Coax
d_4 [mm]	2.6	Diameter of Conductor
l_2 [mm]	327.88	Length of Helix
d_2 [mm]	104.364	Helix Diameter
d_3 [mm]	360.66	Disc Diameter

CALCULATE

Figura 2.24: Calculo de parámetros de antena helicoidal.

⁴⁰ http://www.changpuak.ch/electronics/calc_12a.php

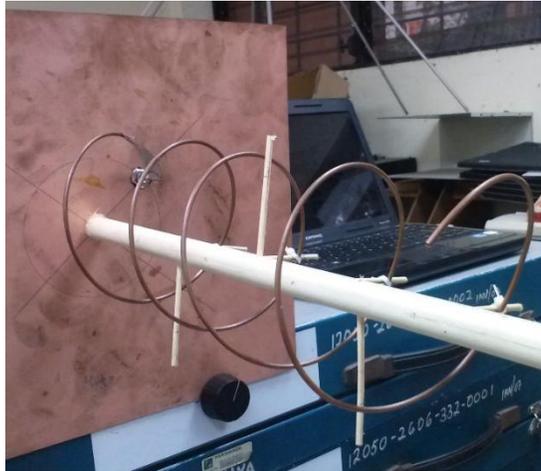


Figura 2.25: Antena helicoidal 915 Mhz implementada.

2.2.4.4 Pruebas.

Las pruebas realizadas para los diferentes tipos de antenas, tienen por objeto medir el alcance y la cobertura desde una posición lineal con la tag, en la siguiente figura se muestra un diagrama esquemático que describe la forma en que se realizaron estas mediciones:



Figura 2.26: Esquema para la realización de pruebas de los diferentes tipos de antenas.

El esquema mostrado en la figura anterior, muestra un instrumento de medición para los niveles de señal emitidos por la antena transmisora; dicho instrumento se conoce como analizador de espectro, con el podemos determinar a qué niveles de señal el lector logra establecer una comunicación con la etiqueta.

Este analizador de espectro consiste básicamente en un pequeño módulo sintonizador de RF como se puede apreciar en la siguiente figura:



Figura 2.27: Sintonizador de RF utilizado como analizador de espectro.

Este módulo por medio de un software⁴¹, puede emular un analizador de espectro virtual de muy bajo costo y de muy buenas prestaciones, en la siguiente tabla se muestran a manera de ejemplo los valores obtenidos de una prueba realizada con este analizador y la interfaz gráfica donde se muestra el valor de señal más alto perteneciente a la tabla

Distancia	Potencia lector 50	
	0,5 mt	0,6 mt
	-43 dB	-15 dB
	-30 dB	-35 dB
	-30 dB	-45 dB
	-20 dB	-32 dB

Tabla 2.6: Valores de intensidad medidos con el analizador de espectro.

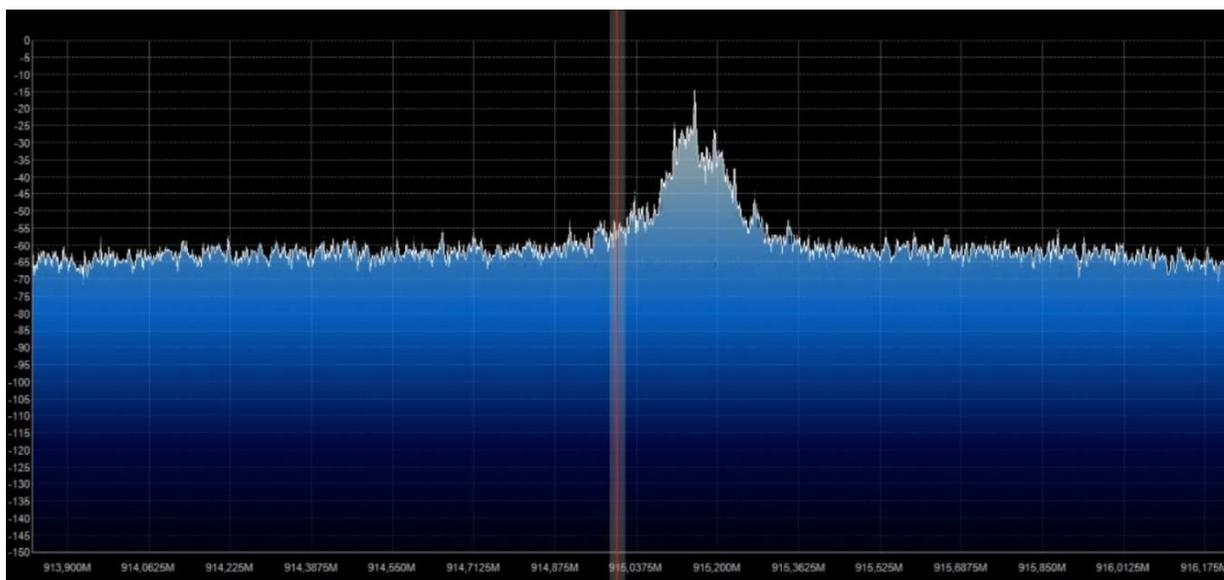


Figura 2.28: Interfaz gráfica de analizador de espectro RTL-SDR.

⁴¹ <http://www.rtl-sdr.com/big-list-rtl-sdr-supported-software/>

Las mediciones realizadas con el analizador de espectro; consisten en ir variando la potencia del módulo lector RFID, el cual se configura en un rango de 1-254, esto es un valor equivalente en binario de la potencia del módulo siendo 254 la potencia máxima del módulo la cual es 20 dBm (100mW).

Todos los datos tomados en las pruebas con los diferentes tipos de antenas descritos anteriormente, consisten en ir aumentando la distancia de la antena del analizador linealmente con la antena en prueba, a una potencia inicial fija todo este procedimiento se repite hasta llegar al valor de la potencia máxima del módulo lector, esta última lectura, para las 3 antenas construidas, es mostrada en la Figura 2.29.

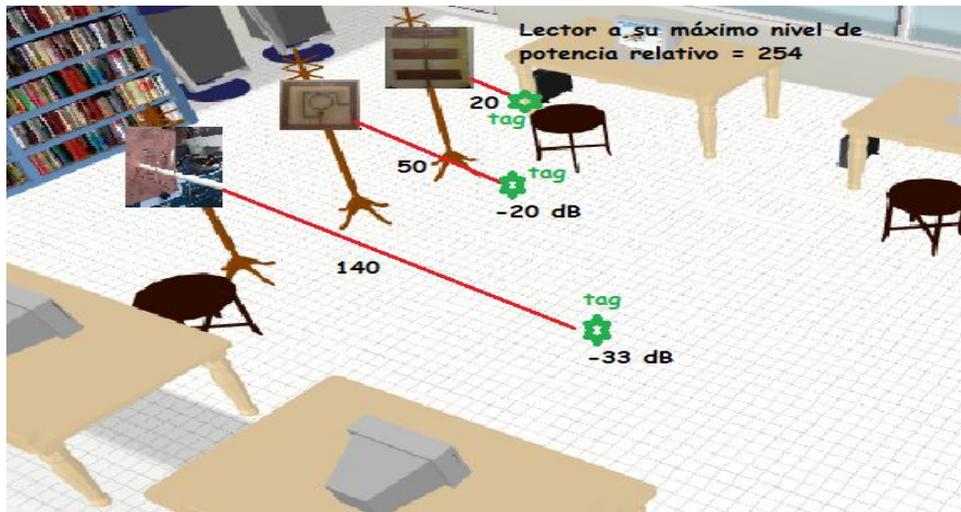


Figura 2.29: Lecturas experimentales con los 3 tipos de antenas (distancias en cms)

Todos los datos tomados para los diferentes tipos de antenas descritos anteriormente se muestran en la tabla 2.7 para una distancia inicial de 0.5 mt, hasta una distancia máxima de 1.4 mt que es la máxima distancia alcanzada; como explicación de la tabla siguiente donde no hay lectura es porque con el analizador de espectro no se alcanzaba a observar la intensidad en dB (que fue para distancias debajo de 50 cms y con la antena parche rectangular, el sistema detectaba la tag a 20 cms de distancia, por eso no se muestra la intensidad en dB; en la figura 2.29).

Distancia en Linea vista (cms)	Ant. Circular (niveles obtenidos dB)						Ant. Parche rectangular (niveles obtenidos dB)						Ant. Helicoidal (niveles obtenidos dB)					
	50	100	110	120	130	140	50	100	110	120	130	140	50	100	110	120	130	140
50	-20	---	---	---	---	---	---	---	---	---	---	---	-34	---	---	---	---	---
100	-30	---	---	---	---	---	---	---	---	---	---	---	-35	---	---	---	---	---

150	-34	---	---	---	---	---	---	---	---	---	---	---	---	-37	-18	-19	-35	---	---
200	-20	---	---	---	---	---	---	---	---	---	---	---	---	-16	-34	-18	-15	-17	-33
254	-20	---	---	---	---	---	---	---	---	---	---	---	---	-33	-15	-20	-18	-32	-33

Tabla 2.7

De acuerdo a los datos de las tablas anteriores; se puede observar que la antena helicoidal es la que tiene un mejor alcance pero no es suficiente para las necesidades operativas del sistema prototipo.

Debido al poco alcance conseguido por los diseños expuestos anteriormente; fue necesario adquirir una antena que cumpla con los valores adecuados de ganancia, polarización, impedancia necesarios para que el lector RFID pueda detectar las tags a una distancia considerable; los datos nominales de la antena se muestran en la tabla 2.8:

Electrical parameters		
Model		UHF915M-ANTPCR
Frequency Range	MHz	902-928 (Compatible 840-960MHz)
Gain	dBi	8
Horizontal Beamwidth	°	68
Vertical Beamwidth	°	68
F/B Ratio	dB	≥20
VSWR		≤1.3
Nominal Impedance	Ω	50
Polarization		Circularly polarized
Maximum Power	W	50
Connector		MMCX
Mechanical parameters		
Dimensions	mm	225×225×35
Weight	kg	1.3
Installation		Embrace rod installation (optional clip code)
Mounting Mast Diameter	mm	φ 40~50

Tabla 2.8: Parámetros técnicos de antena parche circular.

De acuerdo a la información proporcionada por el fabricante⁴²; esta antena puede alcanzar una distancia de 1 a 5 metros con el módulo lector, utilizando la frecuencia estándar de trabajo de EEUU (902.75 ~ 927.25MHz), en la siguiente imagen se muestra la antena

⁴² http://www.soliddepot.com/index.php?main_page=product_info&products_id=231



Figura 2.30: Antena parche circular 915Mhz.

Para caracterizar la antena parche circular a 915 Mhz, se procedió a obtener un mástil, donde se pudiera sostener la antena y moverse de una forma óptima dentro del área a cubrir, como se muestra en la figura 2.30. Para esta prueba no se tomaron en cuenta las alturas a la que se encuentran los artículos con etiquetas (todo fue en un plano horizontal); tanto las alturas, como ángulos de abertura horizontal; se consideraran en el capítulo 3 de este documento.

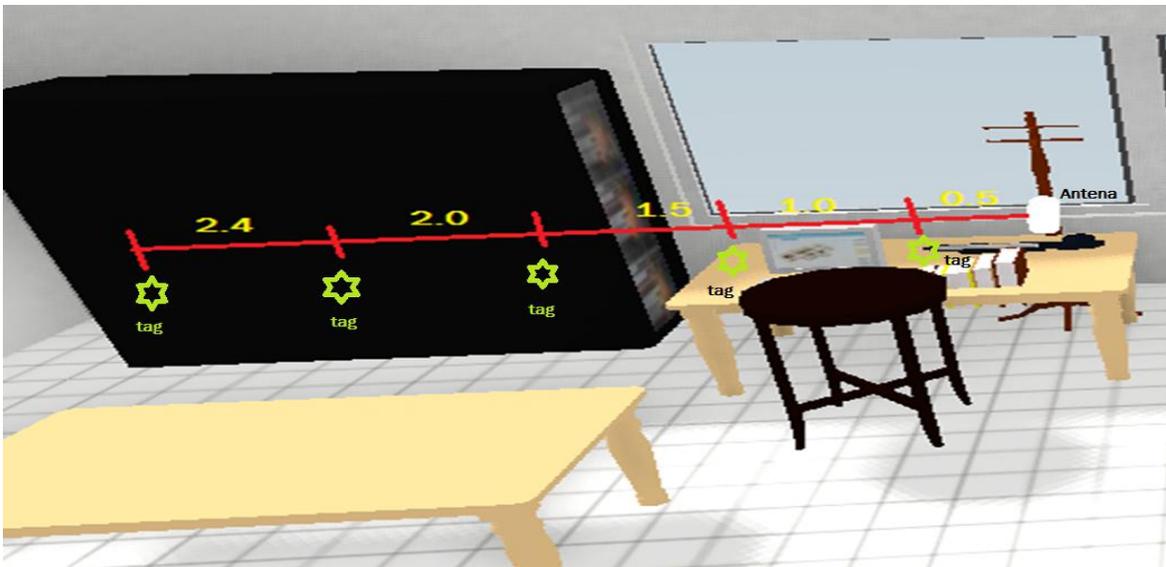


Figura 2.31

Todos los resultados obtenidos son mostrados en la tabla 2.9, estos dejan ver claramente un mejor resultado en cuanto a la distancia alcanzada; pero es importante destacar el hecho de que tan solo se pudo lograr una distancia de 2.4 metros, esto puede deberse a múltiples factores que introducen una serie de interferencias que reducen la operatividad del sistema.

Distancia en Linea vista (ems)	Posición A1 (niveles obtenidos dB)				
	50	100	150	200	240
Nivel relativo de potencia del lector					

50	-33	-33	-15	-34	-19
100	-34	-35	-30	-31	-20
150	-30	-34	-24	-32	-34
200	-13	-31	-34	-34	-35
254	-13	-32	-34	-34	-35

Tabla 2.9: Datos obtenidos con antena parche circular 915Mhz.

Con la caracterización de la antena parche circular, podemos afirmar que con valores de intensidad de potencia recibida por parte de la tag, por debajo de -40 dB la antena no alcanza a detectarla⁴³. También se puede apreciar de la tabla 2.9 que al máximo nivel relativo de potencia del lector (254) es cuando los niveles obtenidos por el analizador de espectro en dB, son más confiables; en la figura 2.32 se observa esta medición; por lo que, para todas las lecturas efectuadas en el capítulo 3, se llevaron a cabo con este nivel relativo de potencia.

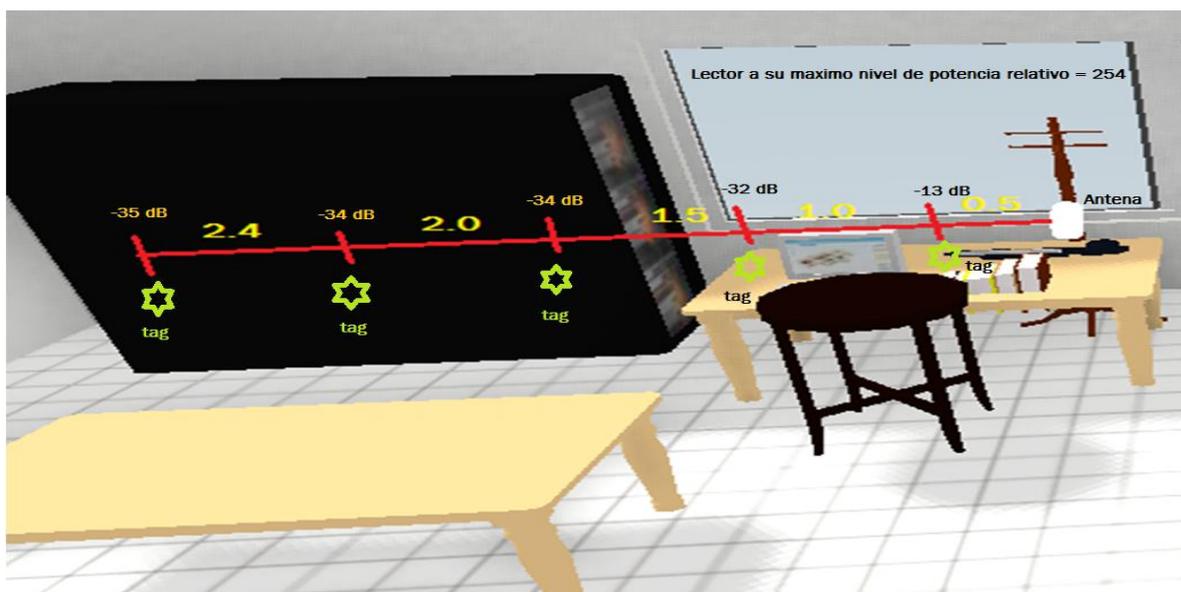


Figura 2.32

⁴³ Ver figura 2.28 para escala en dB

2.2.4.5 Elección de etiqueta para un mejor rendimiento.

Con el sistema se realizaron varias pruebas sobre el funcionamiento del mismo y de la distancia de lectura pero también se realizó pruebas con distintas clases de etiqueta RFID, mostrando así que esta parte que constituye un sistema RFID afecta la distancia o alcance de lectura o escaneo del lector

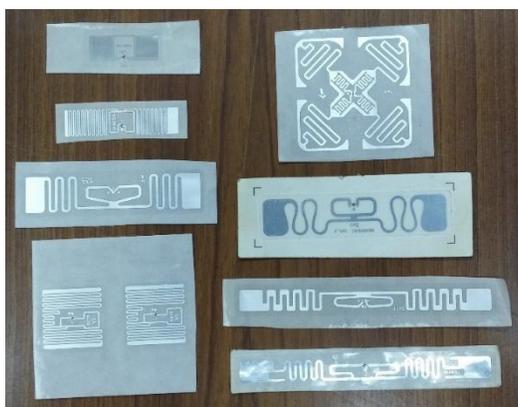
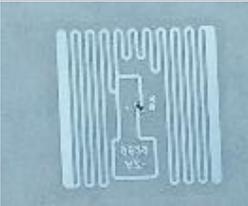
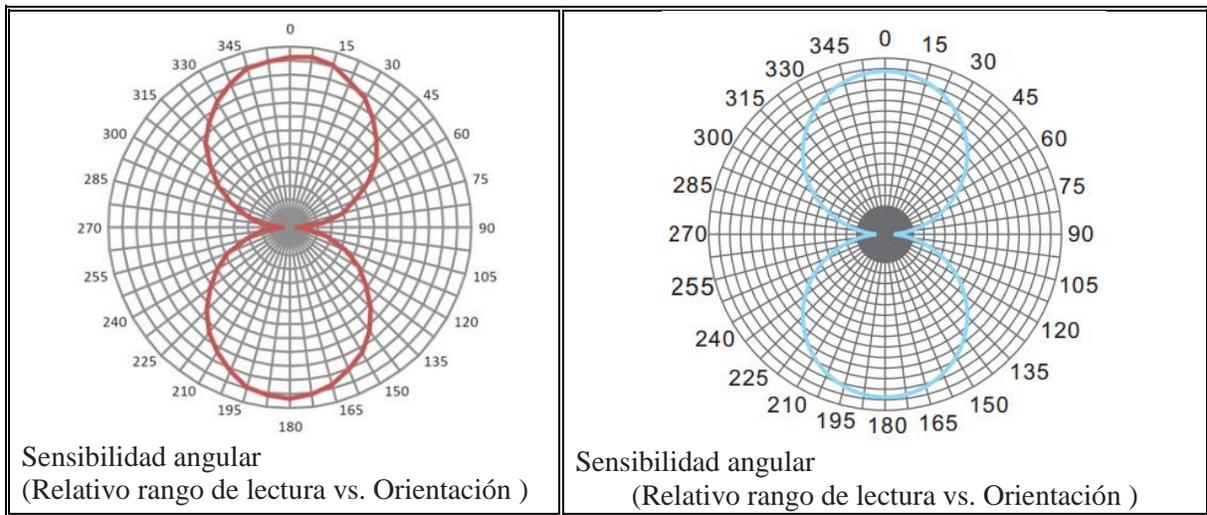
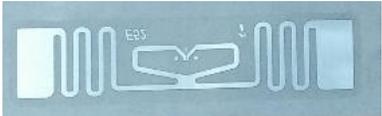
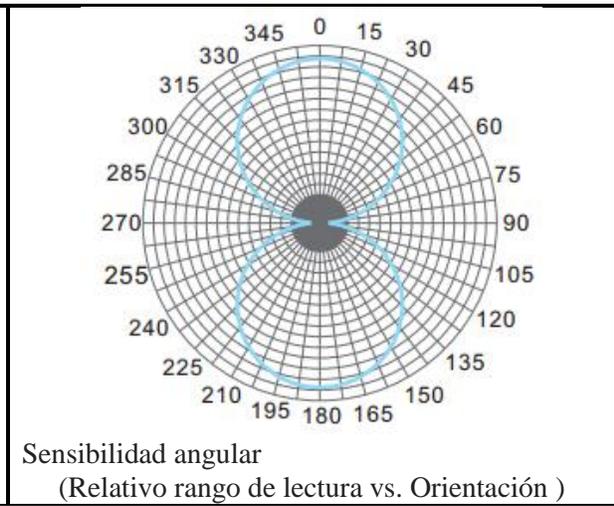
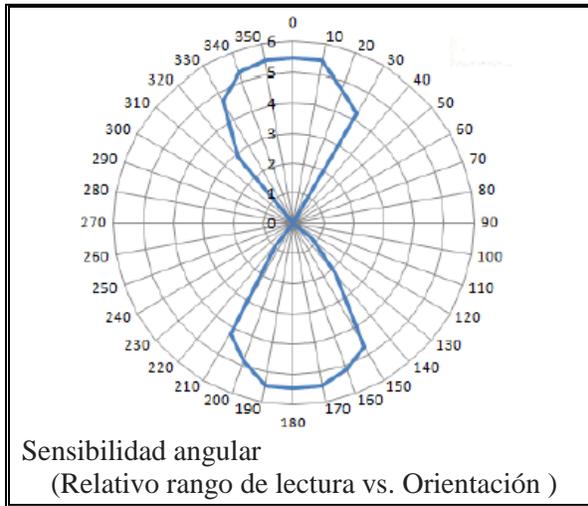


Figura 2.33: Distintas clases de etiqueta RFID, con las cuales se realizaron las pruebas.

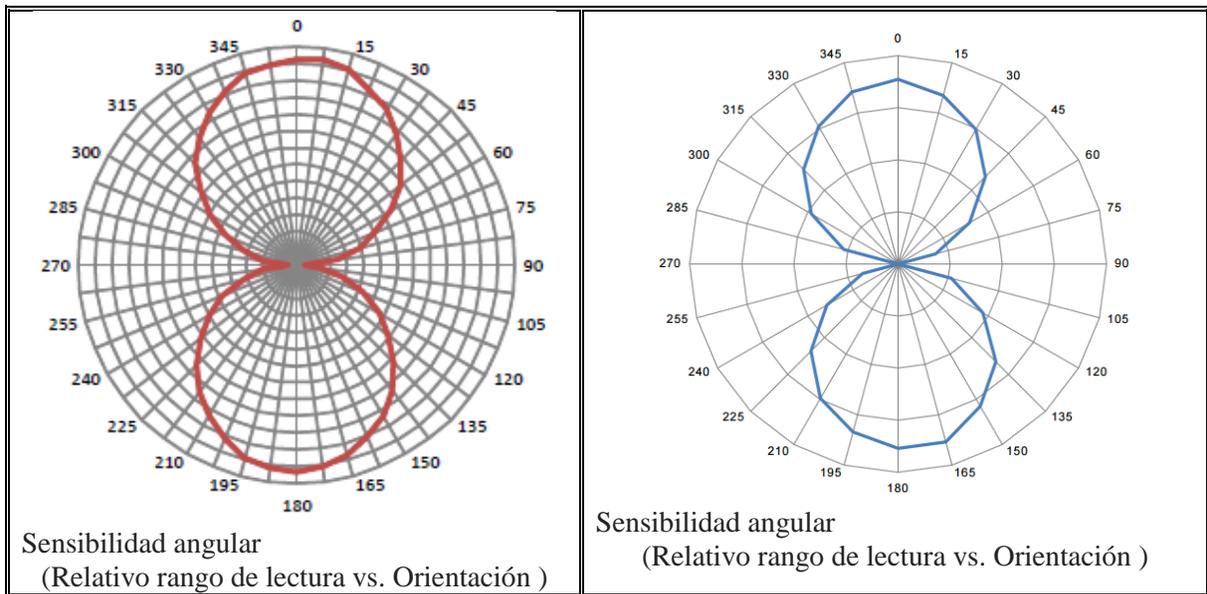
			
Nombre	ALN 9610	Nombre	ALN 9629
Protocolos soportados	ISO / IEC 18000-6C Clase 1 EPCglobal Gen 2	Protocolos soportados	EPCglobal Class 1 Gen 2 ISO/IEC 18000-6C
Circuito integrado	Alien Higgs-3	Circuito integrado	Alien Higgs-3
Frecuencia de operación	840–960 MHz	Frecuencia de operación	840–960 MHz
EPC Tamaño	96 - 480 Bits	EPC Tamaño	96 - 480 Bits
Memoria de usuario	512 Bits	Memoria de usuario	512 Bits
TID	32 Bits	TID	32 Bits
TID único	64 Bits	TID único	64 Bits
Contraseña de acceso	32 Bits	Contraseña de acceso	32 Bits
Sensibilidad angular		Sensibilidad angular	



			
Nombre	Laxcen-E52	Nombre	Laxcen-H47
Sustrato	PET	Sustrato	PET
dimensiones de la etiqueta	68 x 14 mm / 2.68"x0.55"	dimensiones de la etiqueta	44 x 44 mm / 1.73"x1.73"
Tipo de dispositivo	Class 1 Generation 2 passive UHF RFID transponder	Tipo de dispositivo	Class 1 Generation 2 passive UHF RFID transponder
Protocolo de interfaz aérea	EPC Global Class1 Gen2 ISO 18000-6C	Protocolo de interfaz aérea	EPC Global Class1 Gen2 ISO 18000-6C
frecuencia operativa	Global 860-960MHz	frecuencia operativa	Global 860-960MHz
tipo de IC	Monza 5 (En adición a los requisitos especiales)	tipo de IC	Monza 4QT (Además de los requisitos especiales)
configuración de la memoria	128bits EPC ; 48bits TID únicos ; 0bits Memoria de usuario	configuración de la memoria	128bits EPC ; 48bits TID únicos ; 512bits Memoria de usuario
Sensibilidad angular		Sensibilidad angular	



			
Nombre	ALN 9740	Nombre	SMARTRAC Belt
Protocolos soportados	ISO / IEC 18000-6C Clase 1 EPCglobal Gen 2	Protocolo de interfaz de aire:	EPCglobal UHF Clase 1 Gen2 (ISO 18000-6C)
Circuito integrado	Alien Higgs-4	Frecuencia de operación:	Global 860-960 MHz
Frecuencia de operación	840–960 MHz	Memoria EPC contenido:	No garantizado única
EPC Tamaño	128 Bits	Aplicables Materiales superficiales:	Plástico, cartón, vidrio
Memoria de usuario	128 Bits	Dimensiones:	2.87 x 0.67 in (73 x 17 mm)
TID	32 Bits	Temperatura de funcionamiento:	-40 ° a + 85 ° C
TID único	64 Bits	Circuito Integrado (IC)	Impinj Monza 5
Contraseña de acceso	32 Bits	Tamaño antena	69,8 x 14 mm (2.75 x 0.55 in)
Sensibilidad angular		Sensibilidad angular	



			
Nombre	E41-C	Nombre	ALIEN 9620
Sustrato	Pet	Protocolo estándar	ISO / IEC 18000-6C
Tag Dimensiones	100x15mm/3.94 "x 0.59"	Frecuencia de operación	860-960MHz
Tipo de dispositivo	Clase 1 generación 2 pasiva Transpondedor RFID UHF	Único TID	64 bits
Protocolo de interfaz aérea	EPC Class1 Gen2 Global ISO 18000-6C	EPC	96 bits
Frecuencia de funcionamiento	Global 860-960 MHz	Memoria de usuario	512 bits
IC Tipo	Monza 4QT (Además de la Requisitos especiales)	Tamaño de la antena	27x9.7 mm
Configuración de la memoria	EPC 128 bits; TID único 48 bits; Memoria de usuario 512 bits	Substrate Material	PET
		Lectura y escritura:	100000 veces (> 10 años)
		Temperatura de funcionamiento:	-13 ° F a 122 ° F [-25 ° C a + 50 ° C]

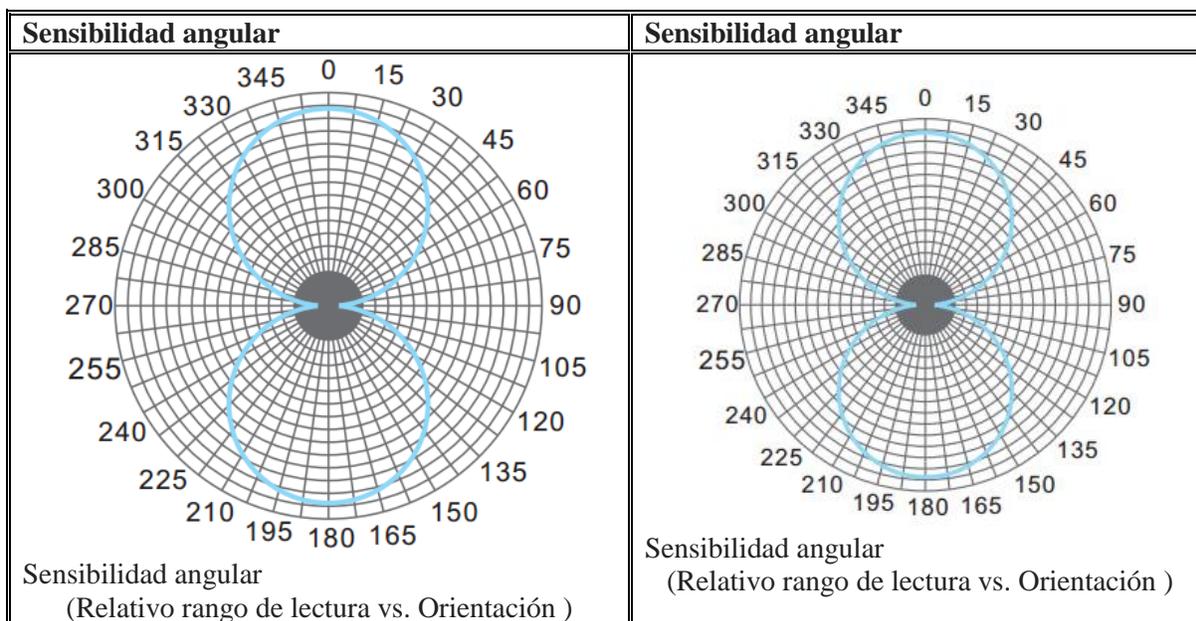


Tabla 2.10: Características técnicas de las etiquetas.

La prueba consistió en utilizar cada tipo de etiqueta para verificar a que distancia el sistema la logra detectar, el funcionamiento de la lectura de las etiquetas pasivas consiste en que el modulo lector RFID genera un pulso de radio frecuencia el cual es detectado por una etiqueta, la misma absorbe la energía de dicho pulso, utilizando parte de ella para que su chip realice sus operaciones y la demás energía es utilizada para que la etiqueta genere un pulso de respuesta con el código EPC de la misma, por consiguiente si la etiqueta posee mayor área de absorción o una antena más grande tendrán una mayor cantidad de energía para generar el pulso de respuesta, por consiguiente más distancia de lectura de la misma.

Por esa razón y los resultados de las pruebas con cada etiqueta, así mismo de las especificaciones técnicas de cada una, verificaron dicha teoría, eligiendo así la etiqueta mostrada en la siguiente figura, siendo esa la que posee la antena de mayor tamaño, y por consiguiente se logró la mayor distancia de lectura de la misma.



Figura 2.34: Etiqueta RFID.

Especificaciones técnicas de la etiqueta elegida:

- **Nombre:** Impinj H47 (True 3D)
- **Chip:** Impinj Monza 4, 4E Monza, Monza 4QT opcional
- **Dimensiones de la etiqueta:** 44 mm * 44 mm/ 1.73" * 1.73"
- **Sustrato:** PET
- **Protocolos soportados:** EPC clase 1 Gen 2, ISO18000 - 6C
- **Tipo de dispositivo:** Clase 1 Generación 2 pasiva Transpondedor RFID UHF
- **Configuración de la memoria:** 128bits EPC; 48bits TID únicos; Dimensiones de la etiqueta 44 x 44 mm / 1,73 " x1.73 " 512bits Memoria de usuario
- **Rango de lectura (2W ERP):** Hasta 4 m / 13,12 ft
- **Temperatura de funcionamiento:** -20°C ~ 80°C / -4°F ~ 176°F
- **Frecuencia de funcionamiento:** 860 ~ 960MHz RFID UHF
- **Tamaño de la antena:** 44 x 44 mm
- **Aplicaciones:** ropa, venta por menor, gestión de activos, logística
- Adhesivo como etiqueta engomada
- **Peso:** 3g/pcs

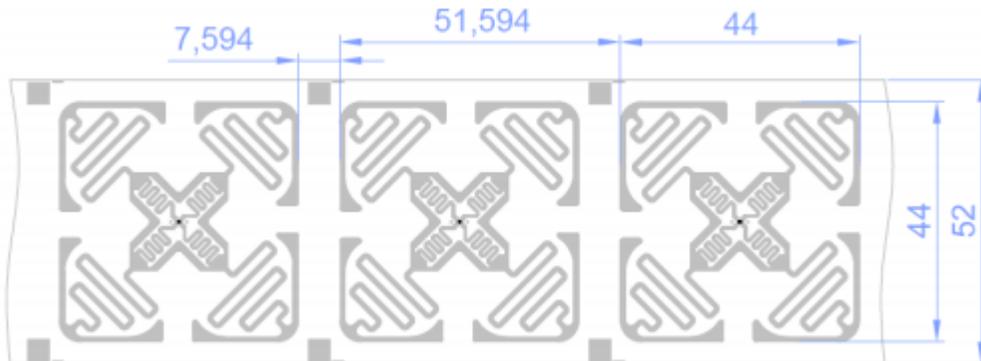


Figura 2.35: Dimensiones física de la etiqueta Impinj H47.

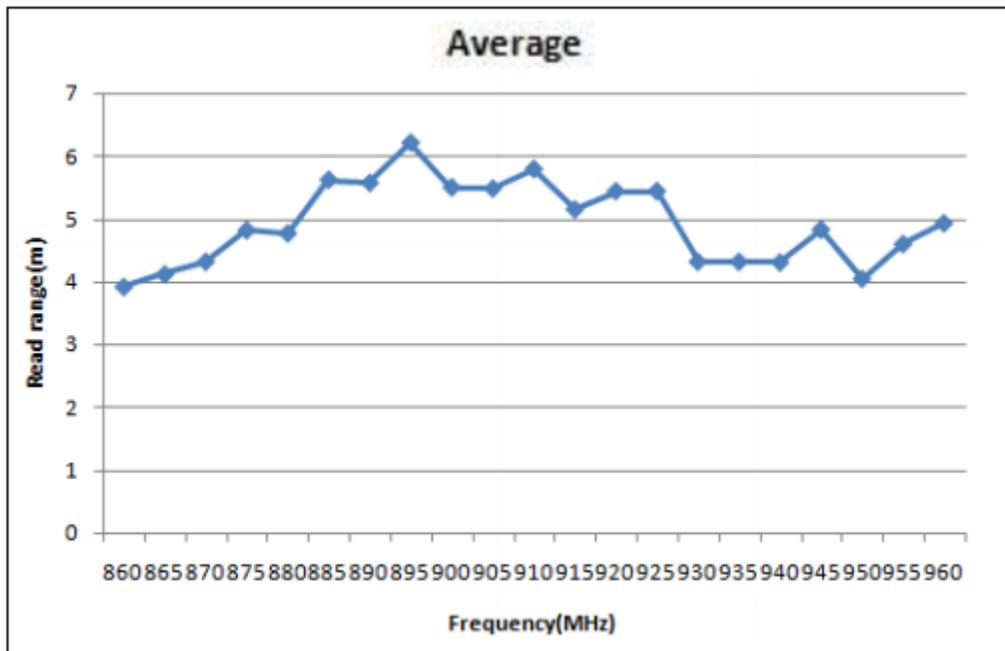


Figura 2.36: Grafica de comparación entre distancia de lectura de la etiqueta Impinj H47 y la frecuencia a la cual se lee.

2.3 SOFTWARE

Se agrupa aquí la programación implementada para el control y funcionamiento de los diferentes componentes del sistema, lo que incluye la interfaz de usuario respectiva.

El modulo central posee dos tipos de interfaces gráficas, las cuales son las que interactúan con el usuario, estas muestran de forma amigable al operador las diferentes funciones que posee el sistema de seguridad RFID, y la forma en que se ejecuta cada una, detalles sobre dicha programación se brindan a continuación.

La interfaz de usuario, la cual se encuentra físicamente en el procesador principal o Raspberry pi B+, se ejecuta con la presencia del usuario cerca (físicamente) del módulo de control; la segunda es una interface WEB, la cual tiene la capacidad de ejecutarse lejos del sistema principal, dando la libertad de ejecutar las funciones donde se crea conveniente. Las dos interfaces gráficas le permiten total libertad con la ejecución de cada función del sistema de control.

2.3.1 Descripción del Software del Sistema.

El sistema prototipo de seguridad RFID y todas sus funciones, se ejecutan por medio de un código de configuración creado en el lenguaje de programación PYTHON, el cual proporciona comunicación y control con cada parte física o proceso lógico del sistema, también existen sub códigos que se ejecutan en relación con el principal, estos están creados en el lenguaje de programación ARDUINO; el primero se encuentra en el ARDUINO MEGA 2560 el cual posee la función de traducir todas las órdenes provenientes del sistema principal hacia el módulo RFID

AS3992, en un lenguaje que el módulo lector comprenda, así como también la respuesta que genera dicho módulo hacia la Raspberry pi B+; el segundo se encuentra en el ARDUINO UNO el cual convierte las órdenes provenientes del procesador principal hacia el ARDUINO GSM SIM900 en una forma que el entienda, también realiza el mismo procedimiento en forma contraria con las respuestas generadas por el módulo GSM.

NOTA: Los códigos cargados en el ARDUINO MEGA 2560 y ARDUINO UNO se encuentran en el ANEXO B.

Para mayor comprensión a continuación se describirá el funcionamiento de cada función, así como también se desglosaran para un mejor entendimiento del código principal, creado en el lenguaje de programación PYTHON:

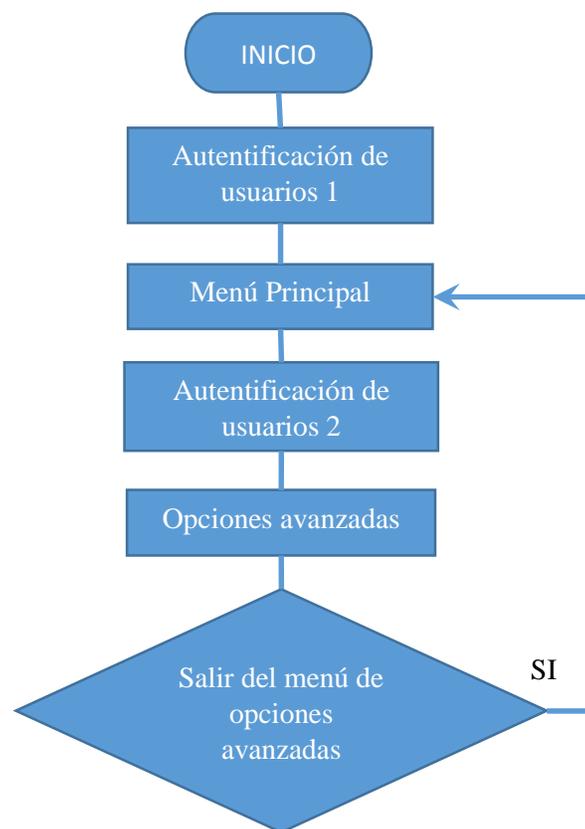


Figura 2.37: Diagrama de flujo del funcionamiento básico del código del sistema.

El sistema contiene dos menús primordiales, el principal que contiene las funciones básicas, y el de opciones avanzadas el cual contiene procesos capaces de modificar físicamente todo el sistema así como eliminar y modificar la información de los elementos en la base de datos, luego se tiene dos autenticaciones de usuarios, en la primera se necesita un permiso básico para poder ingresar y poder operar las funciones del menú principal; la segunda el usuario necesita los permisos de un súper usuario para poder acceder al menú de opciones avanzadas, y por último se tiene la opción de poder regresar al menú principal.

Para mayor comprensión del sistema, se describirá detalladamente el funcionamiento de cada parte que lo compone, así como también la forma que este se comunica con cada uno de sus componentes.

- **Autenticación de usuarios.**

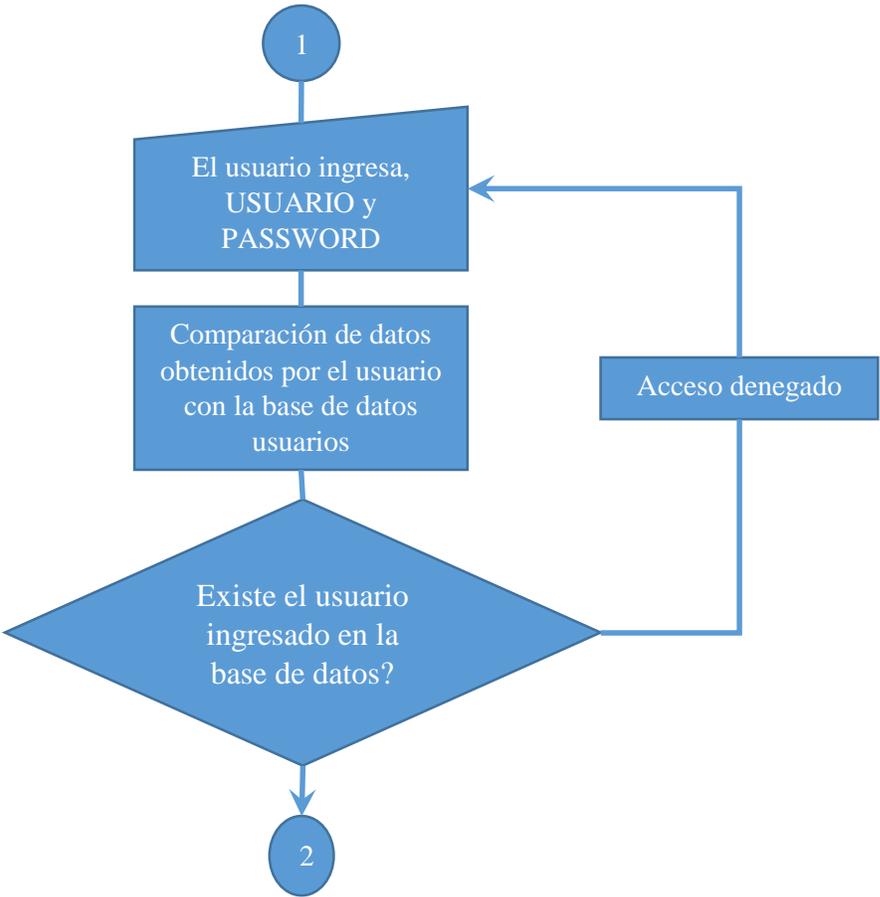


Figura 2.38: Diagrama de flujo del funcionamiento de la función de autenticación de usuarios.

El operador ingresa los datos de USUARIO y PASSWORD al sistema, luego el sistema compara estos datos con los de la base de datos de usuarios, esta posee la información pertinente de los usuarios con permisos para el ingreso al sistema de seguridad RFID, como se mencionó antes, existen dos tipos de permisos, el permiso de usuarios normales, estos permisos solo les permite interactuar con las funciones que se encuentran en el menú principal, pero no se le permite ingresar al menú de opciones avanzadas, el segundo tipo de permisos llamados de súper usuarios, tienen la libertad de interactuar con las funciones ya sea con las del menú principal así como también del menú de opciones avanzadas; cuando el sistema encuentra la información de usuario y password se le otorgará el acceso a este y a las funciones del mismo, si no se le negará el acceso y le pedirá de nuevo que se ingrese los datos de usuario y password.

Las dos autenticaciones de usuarios realizan las mismas funciones, con la diferencia de que el primero le permite el acceso a los usuarios con permisos (cualquiera de los dos mencionados

anteriormente) y el segundo solo el ingreso al menú de opciones avanzadas a los que poseen el permiso de súper usuario.

🚦 MENU PRINCIPAL.

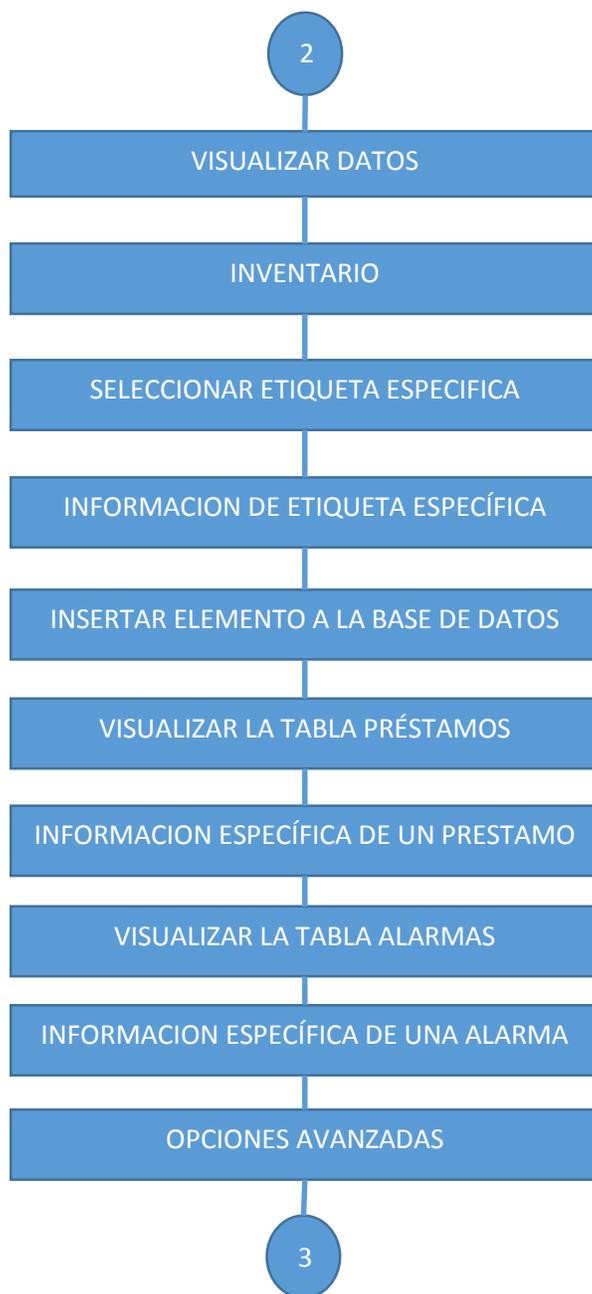


Figura 2.39: Diagrama de flujo del menú principal del sistema.

El menú principal contiene las funciones básicas del sistema, como se muestra en la figura 2.39, dichas funciones le otorgan el control, de la cantidad de elementos, la existencia de los mismos (gracias a la función inventario que realiza un escaneo para saber la existencia física de los objetos),

e información específica, préstamo o alarma que el usuario necesite conocer; a continuación se describirá más detalladamente las funciones del menú principal.

- **VISUALIZAR DATOS.**

Le permite al usuario conocer cada elemento que se ha registrado en el sistema, extrae toda la información de los elementos de la base de datos, luego los divide y clasifica, en datos específicos, es presentada en forma ordenada y fácil de interpretar, la función también le otorga la opción de visualizar de nuevo la base de datos principal, o de salir totalmente de ella; la base de datos principal es donde el sistema guarda toda la información pertinente a los elementos a controlar, esta información se divide en:

- ◇ **ID:** Es el número de identificación EPC o de inventario que se le ha otorgado al elemento, este número proviene de la etiqueta RFID adherida al elemento, este número debe ser de 24 caracteres hexadecimales máximo.
- ◇ **Nombre:** El nombre que el usuario le otorga al elemento para diferenciarlo.
- ◇ **Descripción:** Datos extras sobre el elemento que el usuario piense que son importantes agregarlos, para tener una mejor identificación del elemento.
- ◇ **Estado:** Es una etiqueta textual que se le otorga al elemento para conocer la situación del mismo, esta etiqueta se divide en tres tipos:
 - **PRESENTE:** Le informa al usuario que el elemento se encuentra físicamente en el rango de escaneo del sistema.
 - **PRESTADO:** Le informa al usuario que el elemento ha salido del rango de escaneo del sistema con permiso.
 - **AUSENTE:** Le informa al usuario que el elemento no se detecta en el rango de escaneo del sistema, y que no tiene permisos de salir del mismo.
- ◇ **Fecha y Hora:** Le informa la hora y la fecha en que el elemento fue ingresado a la base de datos principal del sistema.

- **INVENTARIO.**

En la presente función posee las siguientes subfunciones, las cuales se detallaran:

- [A] REALIZAR INVENTARIO Y AL NO DETECTAR UN ELEMENTO REGISTRADO EL SISTEMA GENERA UNA ALARMA.
- [B] REALIZA ESCANEOS Y AL DETECTAR ELEMENTOS REGISTRADOS EL SISTEMA GENERA ALARMAS.

El sistema realiza varios tipos de escaneos gracias a su módulo RFID AS3992, este es capaz de efectuar escaneos de radiofrecuencia para determinar la existencia de los elementos registrados en la base de datos, a continuación se describirán dichas funciones:

- ⇒ [A] REALIZAR INVENTARIO Y AL NO DETECTAR UN ELEMENTO REGISTRADO EL SISTEMA GENERA UNA ALARMA

La presente subfunción se divide en:

- ≈ INVENTARIO SENCILLO
- ≈ INVENTARIO CON RSSI

≈ INVENTARIO CONTINUO

El sistema es capaz de realizar tres tipos de inventarios, sencillo, con RSSI y continuo; un inventario es un registro documental de los bienes pertenecientes a una persona o grupo de personas, realizado con orden y precisión, el sistema de control realiza este tipo de inventario recolectando los número de identificación EPC o ID de cada etiqueta por medio de escaneos de radio frecuencia, agilizando así el tiempo y la eficiencia de la realización de un inventario, logrando un control de elementos más eficaz y preciso.

» INVENTARIO SENCILLO.

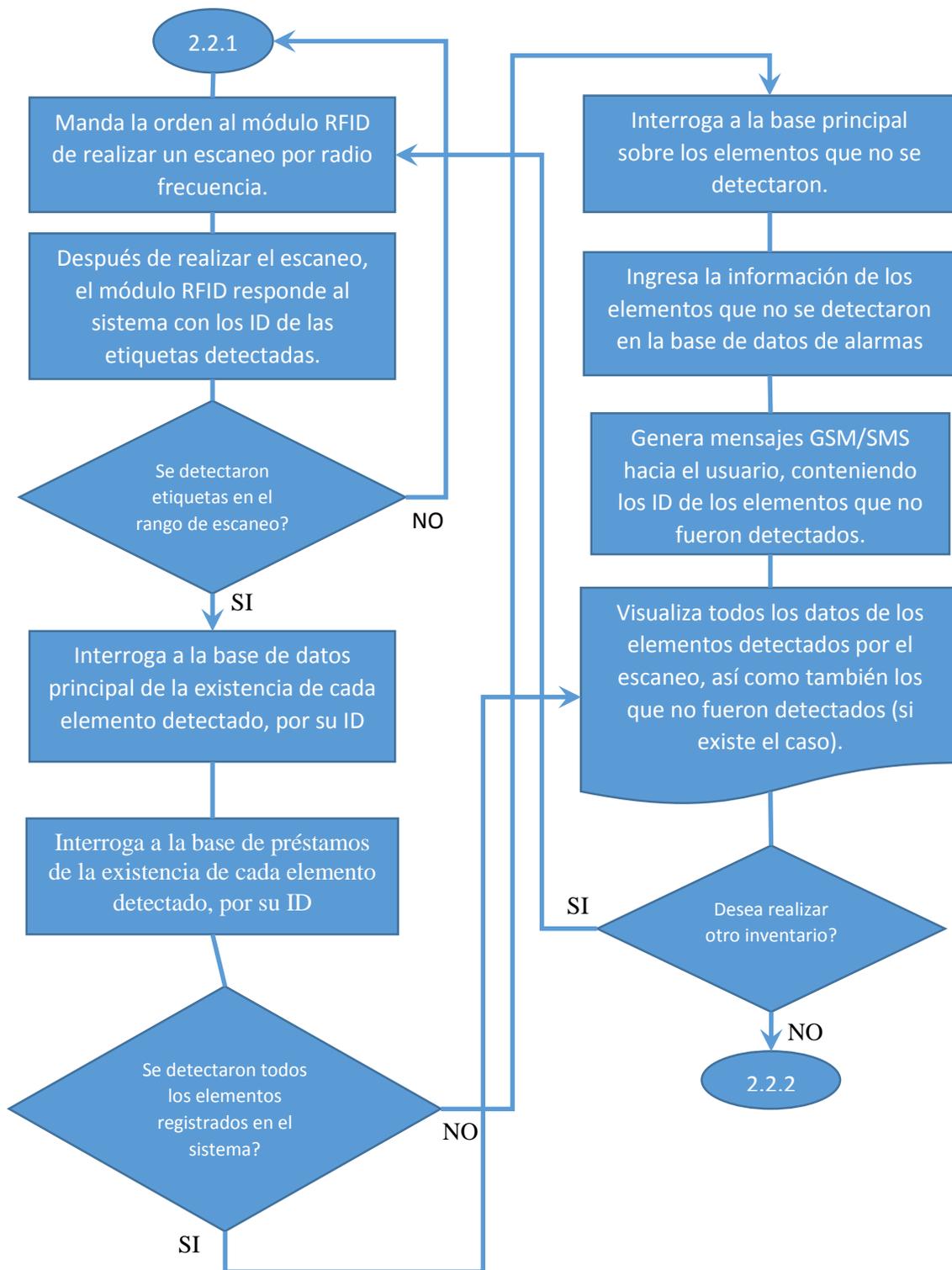


Figura 2.40: Diagrama de flujo del funcionamiento del proceso de inventario sencillo.

La función genera una orden al módulo RFID de realizar un escaneo a una frecuencia fija, después el módulo espera a la respuesta de las etiquetas (estas respuestas contiene el ID), el módulo lector manda estas respuestas, hacia el procesador principal, la función interroga a la base de datos principal, ID por ID (obtenidos del escaneo realizado) sobre la existencia de los elementos detectados, luego de terminar, se realiza el mismo interrogatorio a la base de datos de préstamos, ya que existe la posibilidad que si un elemento no fue detectado por el escaneo, fue prestado a un usuario, si se detecta que existen elementos registrados en el sistema pero que no se detectaron en la realización del escaneo, la función interroga de nuevo a la base de datos principal sobre dichos elementos, teniendo esa información se ingresa en la base de datos de alarmas, se obtiene de la base de datos de usuarios sms el número telefónico de los usuarios a los cuales se enviará el mensaje de alerta a sus teléfonos móviles, después se mandaran conteniendo el ID de los elementos no detectados, y el sistema genera la visualización de tanto de la información de los elementos detectados como de los que no fueron detectados, y por último pregunta al usuario si requiere realizar un nuevo inventario, si no lo requiere se retorna al submenú de los tipos de inventarios.

Si el sistema no detectó ninguna etiqueta, se le informará al usuario y se le preguntará si requiere realizar un nuevo inventario, pero no continuara con los demás procesos de la función, ya que el sistema no tendrá la información de ID para interrogar a las bases de datos.

» INVENTARIO CON RSSI.

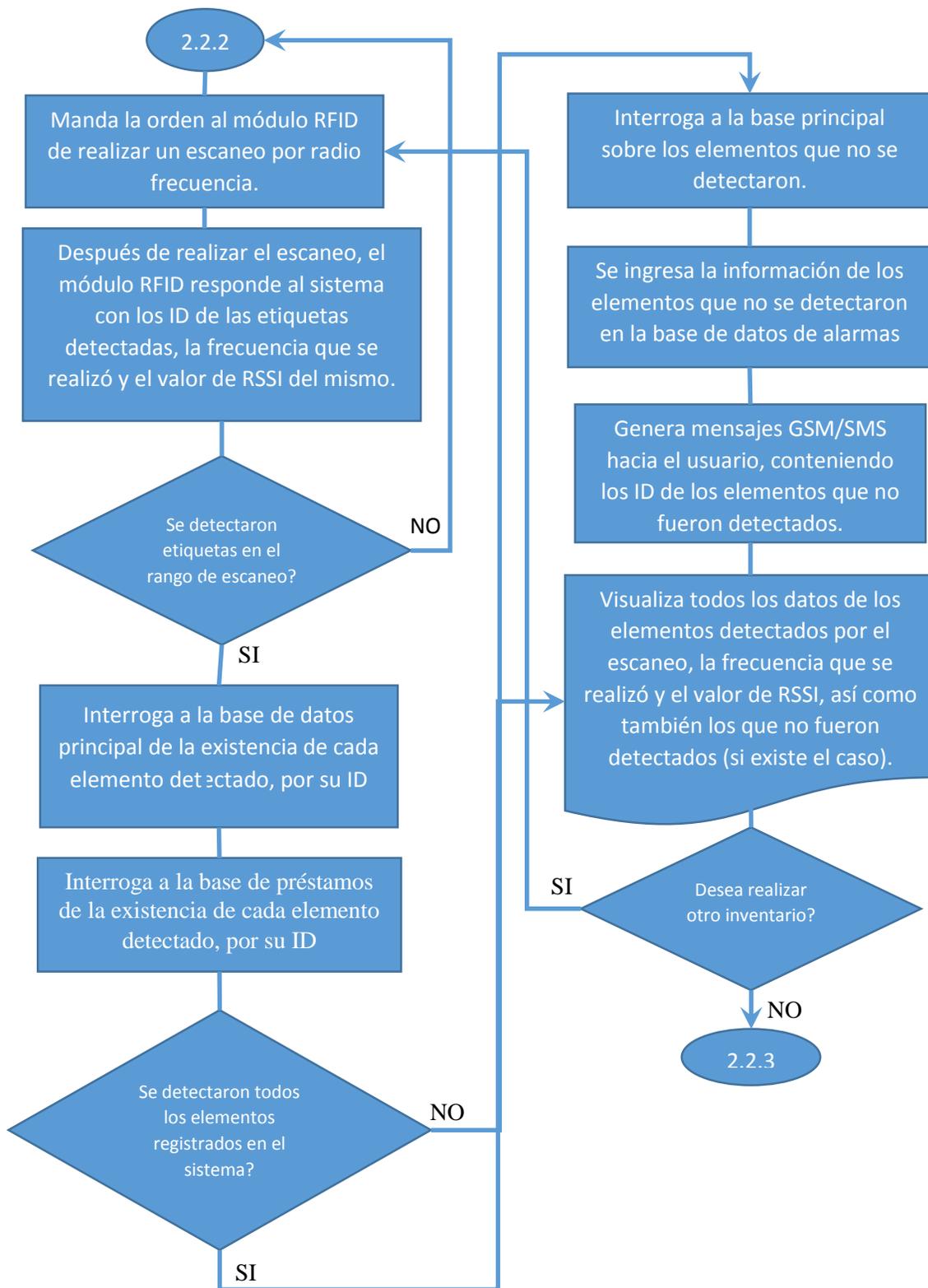


Figura 2.41: Diagrama de flujo del funcionamiento del proceso de inventario con RSSI.

La realización del inventario con RSSI es idéntica a la que ejecuta el inventario sencillo, la diferencia es que se le agrega la información de la frecuencia a la cual el sistema realiza el escaneo, esto quiere decir que el sistema tiene la capacidad de realizar escaneos a diferentes frecuencias, pero la información que otorga el inventario con RSSI es exactamente la frecuencia del escaneo que se realizó en el momento, también muestra el valor de RSSI que se obtuvo en el momento de realizarlo.

El valor de RSSI es el indicador de fuerza de la señal recibida (RSSI por sus siglas en inglés Received Signal Strength Indicator), es una escala de referencia (en relación a 1mW) para medir el nivel de potencia de las señales recibidas por una etiqueta RFID (en este caso); la escala tiene el valor de 0 como centro, representa 0 RSSI o 0 dBm, generalmente los valores de RSSI se expresan en valores negativos, cuando dicho valor es más negativo, mayor representará la pérdida de la señal (El valor RSSI indica la intensidad de la señal recibida, NO de la calidad de dicha señal).

En esta escala, un nivel de 0 dBm corresponde a 1mW, en una escala de 0 a -80 RSSI:

- 0:** Señal ideal, muy difícil de lograr en la realidad.
- 40 a -60:** Señal idónea con tasa de transferencias estables.
- 60:** Representa un enlace bueno
- 70:** Enlace normal-bajo, es una señal medianamente buena, aunque se pueden presentar problemas por efecto de fenómenos externos o climáticos
- 80:** Representa una señal mínima aceptable para establecer la conexión, pueden ocurrir caídas que representan en cortes en la comunicación entre el sistema y las etiquetas.

» **INVENTARIO CONTINUO.**

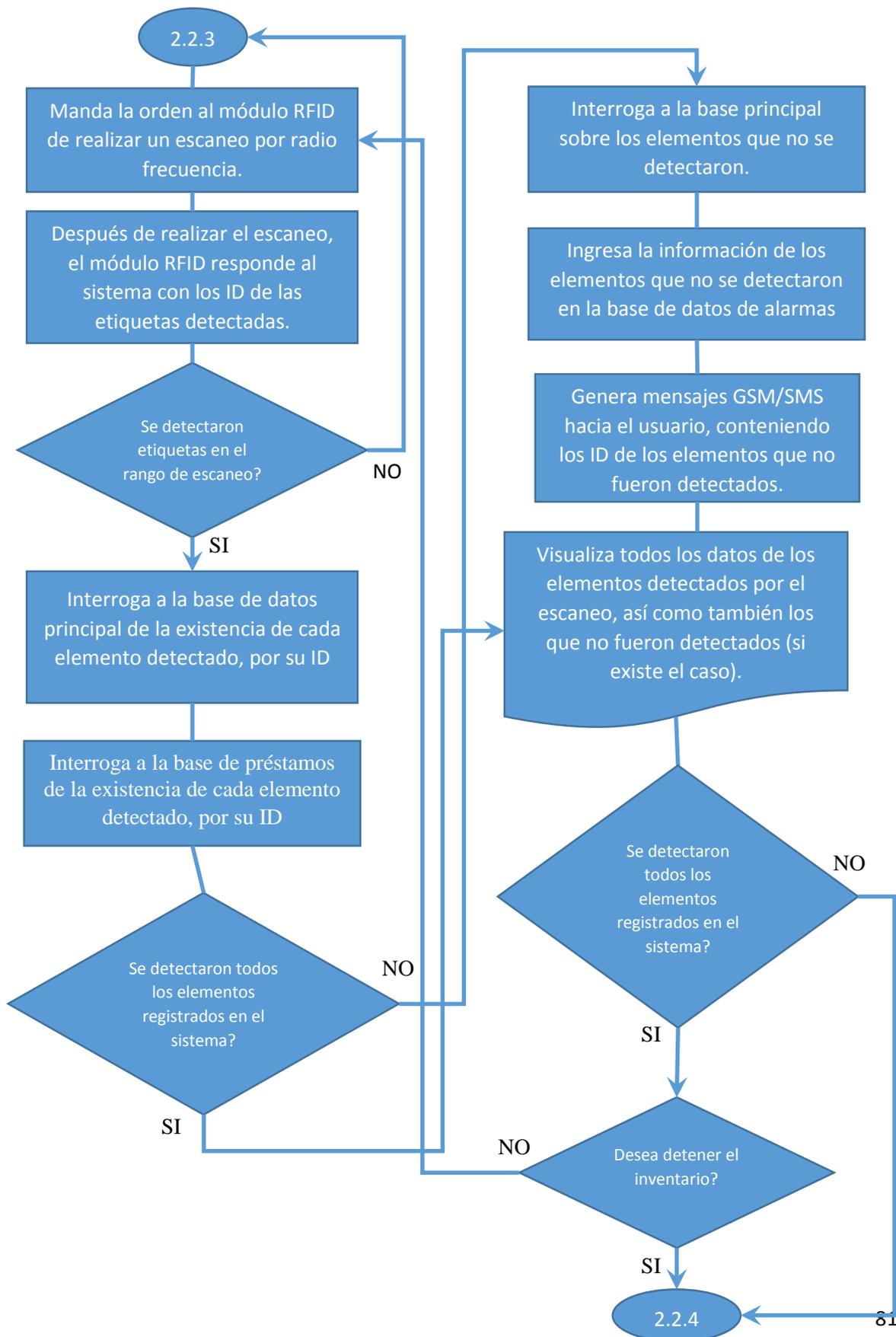


Figura 2.42: Diagrama de flujo del funcionamiento del proceso de inventario continuo.

La función del inventario continuo realiza el mismo procedimiento que la función de inventario sencillo, con la diferencia de que se realizarán varios escaneos, con intervalos de tiempo cada uno, y se detendrá hasta que el usuario lo requiera o hasta que el sistema no detecte algunas o una etiqueta, en estos casos el sistema detendrá la función y mandara el mensaje de texto de alarma a los teléfonos celulares de usuarios específicos.

⇒ [B] REALIZA ESCANEOS Y AL DETECTAR ELEMENTOS REGISTRADOS
EL SISTEMA GENERA ALARMAS

La presente subfunción se divide en:

- ≈ ESCANEEO SENCILLO
- ≈ ESCANEEO CON RSSI
- ≈ ESCANEEO CONTINUO

El sistema realiza este tipo de escaneo cuando el usuario requiere el control de los elementos en la zona que se quiere controlar, cuando se activa esta función el sistema se coloca en lugares de acceso a dicha zona como por ejemplo puertas, luego de colocar el modulo, el sistema realiza escaneos a una frecuencia específica, al contrario de la funciones de inventario descritas anteriormente, no se espera que detecte algún elemento ya que lo que se quiere es un control de salida del sitio, y si detecta un elemento sin permisos de préstamos el sistema ejecutara una alarma, avisando al usuario de la salida no permitida de dicho elemento.

» **ESCANEEO SENCILLO.**

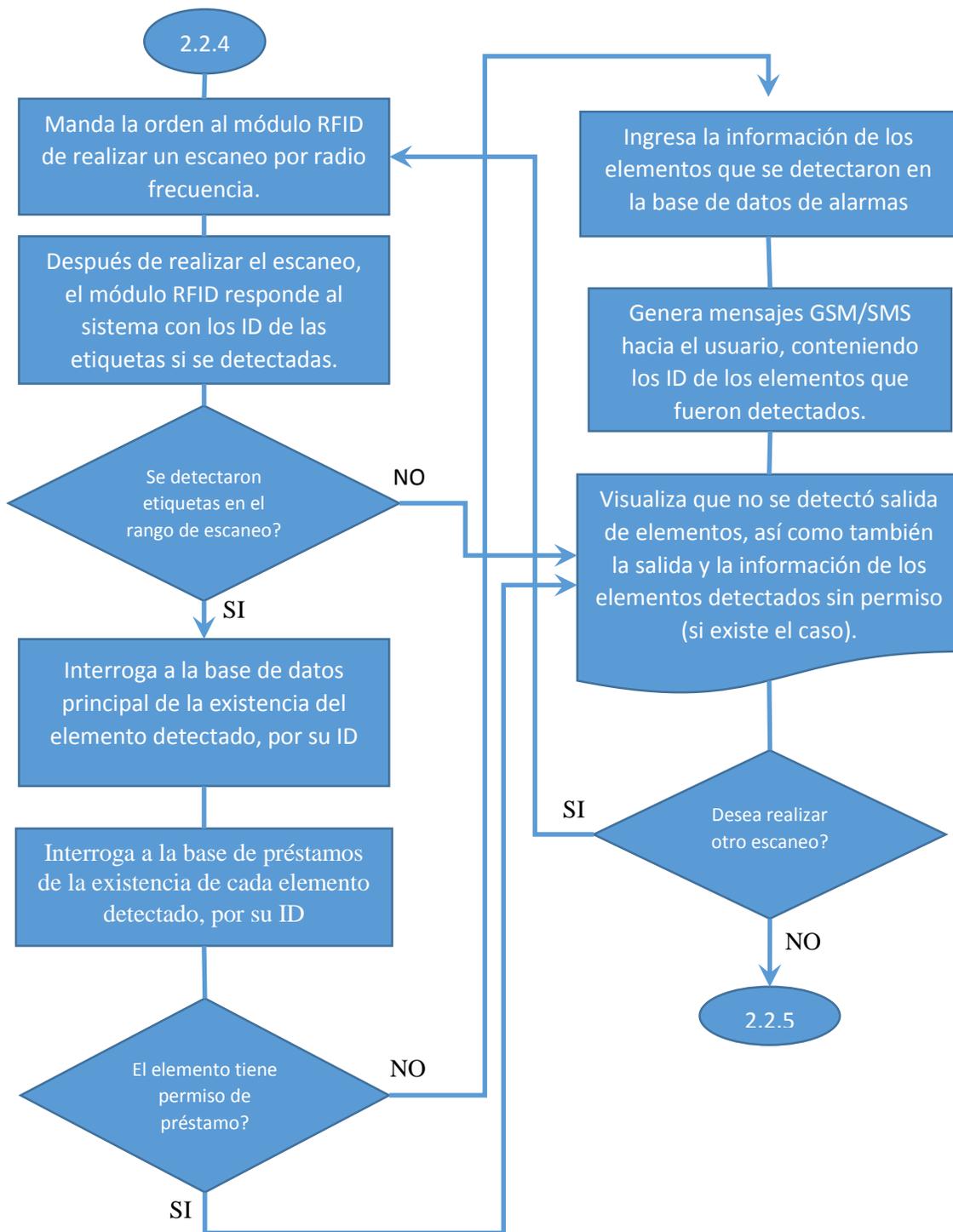


Figura 2.43: Diagrama de flujo del funcionamiento del proceso de escaneo sencillo.

La función genera una orden al módulo RFID de realizar un escaneo a una frecuencia fija, después el módulo espera a la respuesta de las etiquetas si detecta alguna (estas respuestas contiene el ID), el módulo lector manda estas respuestas, hacia el procesador principal, si no se detectó ninguna

etiqueta el sistema informa al usuario de que no se detectó ninguna salida de elementos, pero si se detectó alguna etiqueta la función interroga a la base de datos principal con el ID del elemento, sobre la existencia de los elementos, luego de terminar, se realiza el mismo interrogatorio a la base de datos de préstamos, ya que existe la posibilidad que si un elemento está saliendo de la zona a controlar, fue prestado a un usuario, si dicho elemento no se encontró en la base de datos préstamos, teniendo esa información se ingresa en la base de datos de alarmas, se obtiene de la base de datos de usuarios sms el número telefónico de los usuarios a los cuales se enviará el mensaje de alerta a sus teléfonos móviles, después se mandaran conteniendo el ID de los elementos detectados, y el sistema genera la visualización de la información de los elementos, y por último pregunta al usuario si requiere realizar un nuevo escaneo, si no lo requiere se retorna al submenú de los tipos de escaneos.

» **ESCANEAO CON RSSI.**

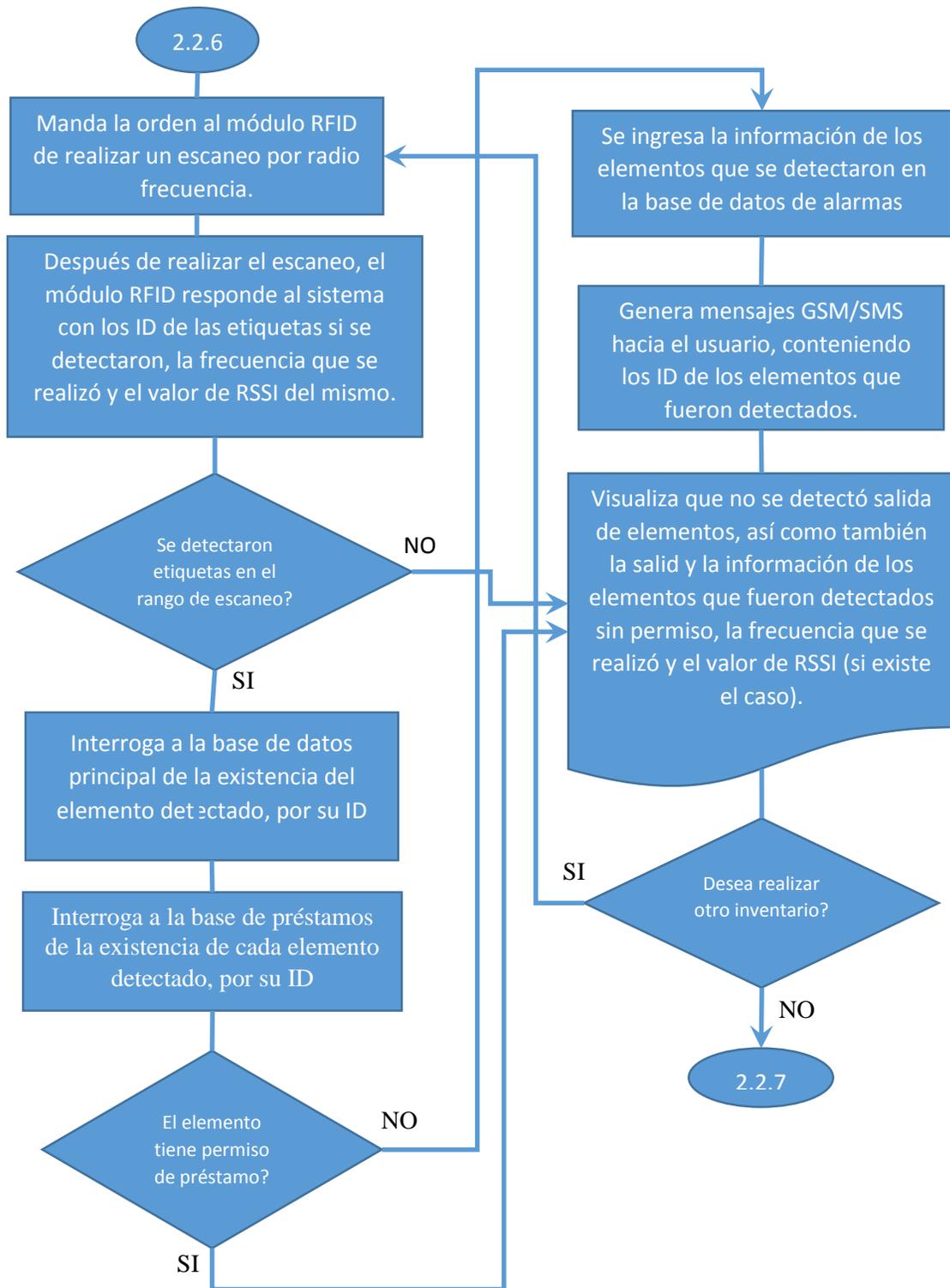


Figura 2.44: Diagrama de flujo del funcionamiento del proceso de escaneo con RSSI.

La realización del escaneo con RSSI es idéntica a la que ejecuta el escaneo sencillo, la diferencia es que se le agrega la información de la frecuencia a la cual el sistema realiza el escaneo, esto quiere decir que el sistema tiene la capacidad de realizar escaneos a diferentes frecuencias, pero la

información que otorga el inventario con RSSI es exactamente la frecuencia del escaneo que se realizó en el momento, también muestra el valor de RSSI que se obtuvo en el momento de realizarlo.

» **ESCANEAO CONTINUO.**

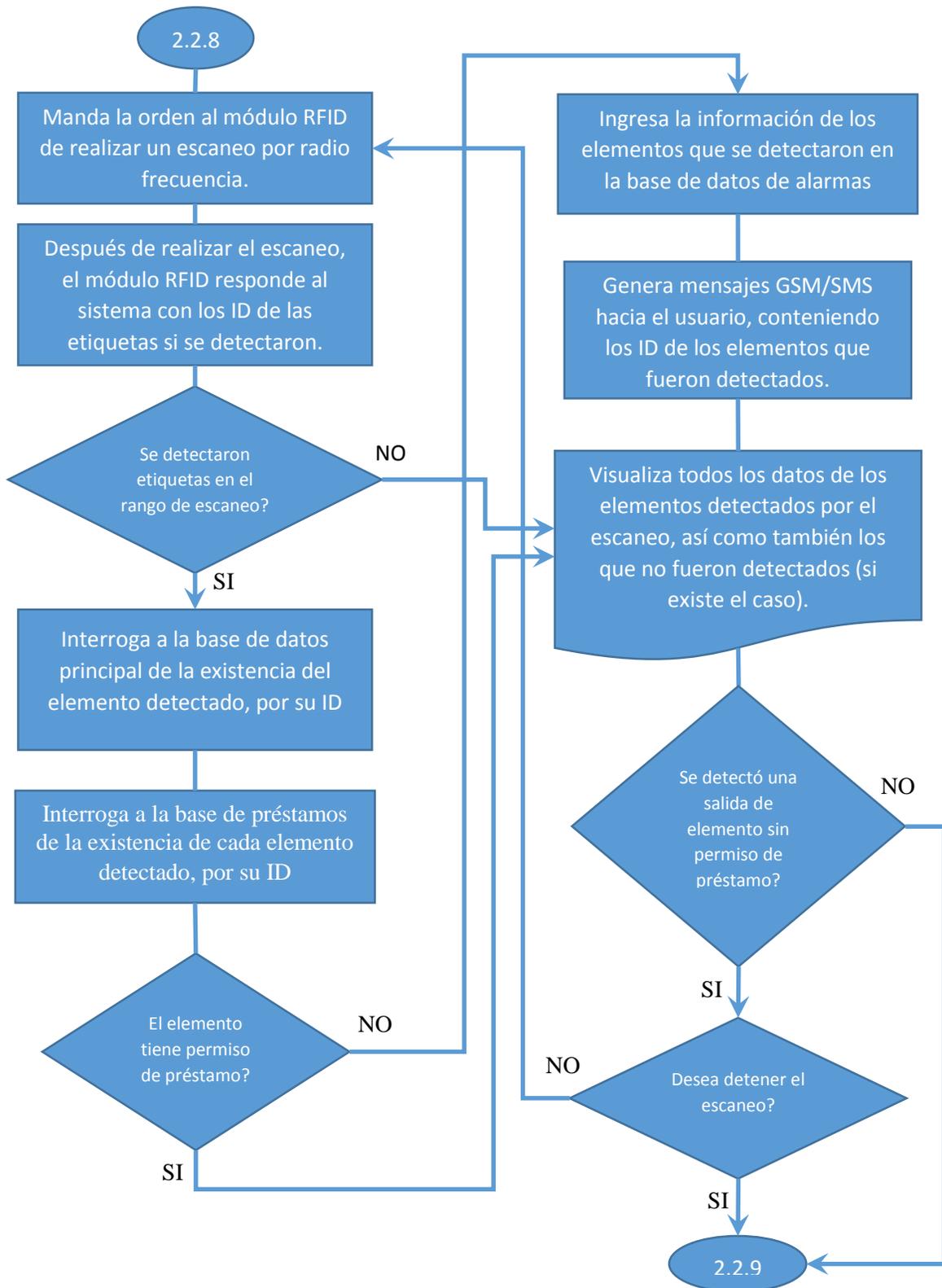


Figura 2.45: Diagrama de flujo del funcionamiento del proceso de escaneo continuo.

La función del escaneo continuo realiza el mismo procedimiento que la función de escaneo sencillo, con la diferencia de que se realizarán varios escaneos, con intervalos de tiempo cada uno, y se detendrá hasta que el usuario lo requiera o hasta que el sistema detecte algunas o una etiqueta, en estos casos el sistema detendrá la función y mandará el mensaje de texto de alarma a los teléfonos celulares de usuarios específicos.

- **SELECCIONAR ETIQUETA ESPECÍFICA.**

La función le pide al usuario ingresar un ID de una etiqueta específica, le advierte si lo ha ingresado correctamente, este debe de componerse de 24 números en formato hexadecimal, luego manda el orden al módulo RFID de realizar un escaneo para buscar la etiqueta determinada, en este proceso el módulo emite un pulso de radiofrecuencia el cual interroga a cada etiqueta que se encuentre en el rango de cobertura, pero con cada respuesta se compara el ID escaneado con el ID ingresado por el usuario al sistema, el lector realiza este proceso hasta que el ID de una respuesta sea igual que se ingresó, o también se detiene el proceso cuando al terminar ninguna etiqueta contenga el ID que se ingresó al sistema; al terminar de realizar el proceso, se manda una respuesta al procesador principal advirtiéndole si la etiqueta fue o no encontrada, y el sistema muestra el resultado imprimiendo dicho mensaje.

Luego la función le preguntará al usuario si requiere seleccionar una nueva etiqueta específica, si no es el caso realizará la función terminará.

NOTA: Al realizar el proceso de seleccionar una etiqueta específica el módulo lector, si la etiqueta es detectada en el escaneo, guarda su ID en su memoria, pero esta no es muy amplia y solo es capaz de guardar un número ID, si el módulo realiza cualquier otro proceso después del de seleccionar una etiqueta específica, se borra el ID de la etiqueta de la memoria y/o si ejecuta otro proceso de seleccionar etiqueta específica después de realizar el mismo proceso anteriormente, el módulo borra el ID guardado y lo reemplaza por el ID de la nueva etiqueta seleccionada, el usuario debe tener muy en cuenta esta situación ya que el ID guardado en la memoria del módulo RFID es utilizado en otras funciones del sistema.

- **INFORMACIÓN DE ETIQUETA ESPECÍFICA.**

La función pide al usuario el ingreso del ID o el NOMBRE de un elemento específico, advierte si ha cometido algún error a la hora de ingresar los datos, por ejemplo el ID debe de ser de 24 números en formato hexadecimal, después que se ingrese los datos requeridos del elemento a buscar, se procede a interrogar a la base de datos principal ya sea con el ID o el NOMBRE del elemento, luego responde con la información correspondiente a dicho elemento (si este ha sido registrado en el sistema), se divide para que se pueda mostrar al usuario en forma ordenada y fácil de interpretar.

Si al interrogar a la base de datos sobre la información de un elemento, y este no se encuentra allí el sistema advierte al usuario que dicho elemento no se encuentra registrado.

- **INSERTAR ELEMENTO A LA BASE DE DATOS.**

Se pide al usuario que ingrese los datos correspondientes al nuevo elemento, los cuales son:

- ◇ **ID:** Número de identificación EPC que se le otorgará al elemento, es el mismo número de identificación que posee la etiqueta RFID adherida al elemento.
- ◇ **NOMBRE:** Es el nombre que el usuario le asigna al elemento para poder diferenciarlo de los demás.
- ◇ **Descripción:** Son datos extras que el usuario decida que son pertinentes para especificar el elemento

Así también la función advierte si ha cometido algún error al ingresar los datos, y le pide que lo haga de forma correcta, el ID y el NOMBRE del elemento, estos datos son obligatorios para el proceso, si no los ingresa, comunicara que se ha cometido un error y le pedirá que ingrese dicha información, pero el dato de DESCRIPCIÓN no es obligatorio, por ende el usuario tiene la libertad de ingresarlo o no, y la función le permitirá proseguir sin ningún error.

Luego que la función obtenga los datos pertinentes del elemento, provenientes del usuario, interroga a la base de datos principal, con el ID del elemento, si este ya ha sido registrado anteriormente, se detiene el proceso y notifica sobre la existencia del mismo.

Si la base de datos principal le responde que no existe o no ha sido registrado anteriormente, la función escribe o registra la información del elemento, luego de que este proceso haya sido completado correctamente el sistema muestra toda la información que ha sido registrada; por último se pregunta si requiere registrar un nuevo elemento, si se acepta se reinicia la función de lo contrario la función termina.

- **VISUALIZAR LA TABLA DE PRÉSTAMOS.**

Realiza el mismo proceso que la función VISUALIZAR DATOS, con la diferencia que se extrae la información de la base de datos de préstamos; la base de datos de préstamos es donde el sistema guarda toda la información pertinente a los elementos que no se encuentran pero que poseen permiso para no estar, esta información se divide en:

- ◇ **Estado:** Es una etiqueta textual que se le otorga al elemento para conocer la situación del mismo, en este caso la etiqueta es PRESTADO, informado que el elemento ha sido prestado.
- ◇ **ID:** Es el número de identificación EPC o de inventario que se le ha otorgado al elemento, este número proviene de la etiqueta RFID adherida al mismo, debe de contener 24 caracteres hexadecimales máximo.
- ◇ **Nombre:** El nombre que el usuario le otorga al elemento para diferenciarlo.
- ◇ **Fecha y Hora:** Le informa la hora y la fecha en que el elemento fue ingresado a la base de datos de préstamos del sistema, por ende el tiempo en que fue prestado el elemento.
- ◇ **Nombre de quien realizó el préstamo:** Es el nombre de la persona que realiza el préstamo del elemento.
- ◇ **Nombre del documento de identidad de quien realiza el préstamo:** Es el tipo de documento que demuestre la identidad de la persona que realiza el préstamo.
- ◇ **Número del documento de identidad de quien realiza el préstamo:** Es un número que lo relacione con el documento de identidad que proporcionó la persona que realiza el préstamo.
- ◇ **Tiempo de préstamo:** Es el tiempo que el usuario decida pertinente que dure el préstamo del elemento.

- ◇ **Descripción:** Datos extras sobre el elemento que el usuario crea que son importantes agregarlos, para tener una mejor identificación del elemento prestado.

- **INFORMACIÓN DE PRÉSTAMO ESPECÍFICO.**

Realiza el mismo proceso que la función INFORMACIÓN DE ETIQUETA ESPECÍFICA, con la diferencia que se interroga a la base de datos de préstamos, y le pide al usuario el ingreso del ID, el nombre del elemento, el nombre o el número del documento de identidad de quien realizó el préstamo de un elemento préstamo específico, para realizar la búsqueda de dicha información en la base de datos.

- **VISUALIZAR LA TABLA ALARMAS.**

Realiza el mismo proceso que la función VISUALIZAR DATOS, con la diferencia que se extrae la información de la base de datos de alarmas; la base de datos de alarmas es donde el sistema guarda toda la información pertinente a los elementos que no se encuentran en la zona a controlar y que no tienen permiso de préstamo, esta información se divide en:

- ◇ **Estado:** Es una etiqueta textual que se le otorga al elemento para conocer la situación del mismo, en este caso la etiqueta es AUSENTE, informando que el elemento no se encuentra físicamente.
- ◇ **ID:** Es el número de identificación EPC o de inventario que se le ha otorgado al elemento, este número proviene de la etiqueta RFID adherida al elemento, este número debe ser de 24 caracteres hexadecimales máximo.
- ◇ **Nombre:** El nombre que el usuario le otorga al elemento para diferenciarlo.
- ◇ **Descripción:** Datos extras sobre el elemento que el usuario piense que son importantes agregarlos, para tener una mejor identificación del elemento ausente.
- ◇ **Fecha y Hora:** Le informa la hora y la fecha en que el elemento fue ingresado a la base de datos de alarmas del sistema.

- **INFORMACIÓN DE UNA ALARMA ESPECÍFICA.**

Realiza el mismo proceso que la función INFORMACIÓN DE ETIQUETA ESPECÍFICA, con la diferencia que se interroga a la base de datos de alarma, y le pide al usuario que ingrese el ID o el NOMBRE de un elemento específico que produjo una alarma, para realizar la búsqueda de dicha información en la base de datos.

- **OPCIONES AVANZADAS.**

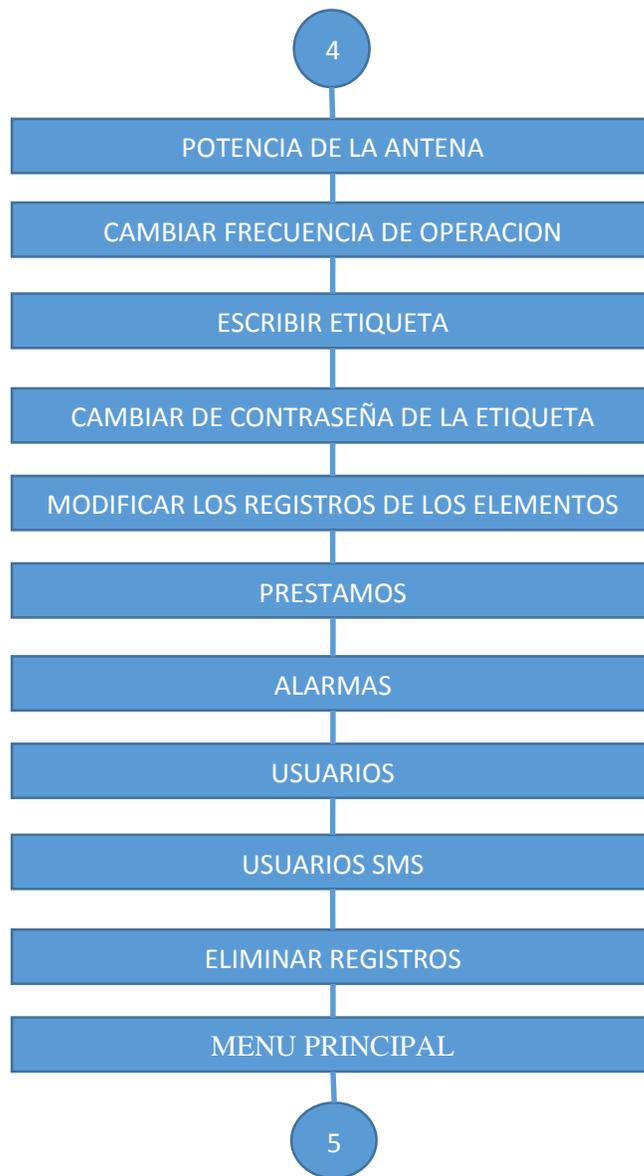


Figura 2.46: Diagrama de flujo del funcionamiento del proceso de las opciones avanzadas.

El menú de opciones avanzadas son un grupo de funciones que son capaces de cambiar las características físicas de funcionamiento del sistema y modificar los registros de todas las bases de datos, otorgando un gran control al usuario del sistema, teniendo muy en cuenta que los tipos de usuarios que tiene permitido ingresar a este menú son con permisos de súper usuarios, lo cual le permite total interactividad con dichas funciones, a continuación se describirá más detalladamente el funcionamiento de dichos procesos:

- **POTENCIA DE LA ANTENA.**

La presente función contiene las sub funciones, las cuales se explicaran más ampliamente:

- ACTIVAR ANTENA
- CAMBIAR POTENCIA DE LA ANTENA
- DESACTIVAR ANTENA

El sistema tiene la capacidad de modificar la potencia de la antena externa del módulo RFID, así como activarla y desactivarla, dándole al usuario un gran control en la forma de que el sistema realiza escaneos.

⇒ **ACTIVAR ANTENA.**

El proceso de activar la antena externa le permite al sistema poder realizar escaneos a una distancia más grande, ya que si no se activa, la distancia no sobrepasa algunos centímetros, el sistema le manda la orden al módulo RFID y este responde con la afirmación que la antena ha sido activada.

⇒ **MODIFICAR LA POTENCIA DE LA ANTENA.**

El sistema es capaz de modificar la potencia de la antena, el usuario debe de ingresar el valor de la potencia que se requiera, debe de estar en el rango de 1 a 254, el sistema advierte cuando éste ha cometido un error al ingresar dicho valor, ya que debe de estar en formato decimal, y no sobrepasar su rango de valores, y no deja avanzar con el proceso de la función si se detecta un error al ingresar el dato, al ingresar correctamente la potencia de la antena el sistema manda la orden al módulo RFID de modificar la potencia con que radia la antena externa, y dicho módulo responderá advirtiéndole que se realizó correctamente la modificación.

⇒ **ANTENA DESACTIVADA.**

El sistema le manda la orden al módulo RFID y este responde con la afirmación que la antena ha sido desactivada, el usuario debe de tomar en cuenta que cuando el módulo lector o el sistema se energice, la antena estará desactivada por defecto.

- **CAMBIAR FRECUENCIA DE OPERACIÓN.**

La presente función contiene las sub funciones, las cuales se explicaran más ampliamente:

- ACTIVAR MODO SALTO (Agregar la frecuencia a la lista de frecuencias)
- DESACTIVAR MODO SALTO (Limpiar lista de frecuencias)

El sistema es capaz de modificar la frecuencia en que el módulo RFID realiza los escaneos, esto le otorga más eficiencia y control, ya que se puede realizar inventarios a una frecuencia determinada y que no sea interferido por algún elemento externo que radie en la misma banda, el módulo lector radia en la banda de UHF, para ser más preciso en el rango de 800 a 990MHz, existen dos procesos para realizar dichos escaneos: Activar modo salto y Desactivar modo salto.

⇒ **ACTIVAR MODO SALTO (Agregar la frecuencia a la lista de frecuencias).**

La función le pide al usuario que ingrese el valor de la frecuencia, el cual debe de ser en kHz, por ejemplo para ingresar la frecuencia 850MHz el usuario debe ingresar 850000 kHz, así también se advertirá si ha cometido un error al ingresar el valor, como por ejemplo se ha sobrepasado el valor del rango establecido el cual es 800-990MHz, luego la función mandará la orden al módulo RFID

de agregar la frecuencia a la lista para así activar el modo salto, si el sistema tuvo un problema al agregar la frecuencia se advertirá del error.

El modo salto es un proceso que posee el módulo RFID de realizar los escaneos, esto quiere decir que cada vez que se ejecuta una lectura lo hace a frecuencias diferentes, pero el usuario debe tener presente que el módulo no hace un barrido de frecuencias, sino que al realizar un escaneo lo hace a una determinada frecuencia y al realizar el siguiente lo ejecuta a una diferente frecuencia y así sucesivamente, pero al llegar al final de la lista de frecuencias, el módulo reinicia la lista y el siguiente lo realiza con la primera frecuencia; se debe tomar en cuenta que al energizar el módulo y el sistema; el modo salto estará activado por defecto ya que la lista de frecuencias será: 866300 kHz y 866900 kHz, si ingresa una frecuencia con la opción del modo salto activado esta frecuencia se agregara a la lista anterior.

⇒ **DESACTIVAR MODO SALTO (LIMPIAR LISTAS DE FRECUENCIAS).**

Al desactivar el modo salto, el módulo RFID borra la lista de frecuencias que poseía y coloca únicamente una sola, la cual es la frecuencia que ingresó el usuario, debe de tener en mente que si después de ejecutar el proceso de desactivar el modo salto y posteriormente se realiza el proceso de activarlo nuevamente, lo que hará el módulo RFID es inicializar la lista de frecuencias colocando como primera en la lista, la frecuencia que se introdujo cuando se ejecutó el proceso de desactivar el modo salto y en segundo lugar en la lista se colocará la que se ingresó cuando se ejecutó la función de activar el modo salto, y así se inicializará la lista de frecuencias.

• **ESCRIBIR ETIQUETA.**

El sistema es capaz de modificar número EPC que posee la etiqueta RFID, el usuario debe de tener claro que la etiqueta debe de ser reescribible; se requiere que se ingrese la contraseña de la etiqueta a escribir, esta se compone de 8 caracteres en formato hexadecimal, se advertirá al usuario si ha cometido un error al ingresar dicha información.

Debe de tenerse en cuenta que las etiquetas RFID tienen una contraseña por defecto, la cual es 00 00 00; la contraseña para escribir la etiqueta RFID es de vital importancia ya que esta es el permiso que la etiqueta le pide al usuario para poder modificar su ID, luego la función pedirá que ingrese el nuevo ID de la etiqueta, este parámetro debe de componerse de 24 dígitos máximo en formato hexadecimal, y de igual manera se advertirá si este se ha ingresado de forma errónea, luego se mandará al módulo RFID la orden de escribir dicha etiqueta.

NOTA: Para que el módulo RFID pueda realizar correctamente el proceso de escribir o modificar el ID de una etiqueta, anteriormente se debe de haber realizado el proceso de SELECCIONAR ETIQUETA ESPECÍFICA ya que como se mencionó anteriormente al realizar esta función el módulo guarda en su memoria el ID de la etiqueta seleccionada, inmediatamente el usuario debe de realizar el proceso de ESCRIBIR ETIQUETA ya que la etiqueta que será escrita será la que contendrá el ID que el módulo ha guardado en su memoria, si no se ejecutan en ese orden el módulo lector no encontrará la etiqueta a modificar, advertirá dicho error, ya que en su memoria no habrá ningún ID.

Luego de que el módulo reciba la orden, realizara un escaneo para buscar la etiqueta a escribir, al encontrarla se modificara el ID, pero el usuario debe tener en cuenta que la contraseña que ingresó puede que no sea la correcta para dejar que el lector modifique su información, luego se mostrará el

mensaje al sistema si se escribió correctamente la etiqueta o se produjo un error y cual es este; es muy importante que la etiqueta debe de estar cercana del lector, para que en dicho proceso no se creen errores.

- **CAMBIAR CONTRASEÑA DE LA ETIQUETA.**

El sistema es capaz de modificar la contraseña que posee la etiqueta RFID, el usuario debe de tener en mente que se necesita que se pueda escribir ya que existen etiquetas que no se pueden modificar su contenido o datos.

El usuario debe de contar que las etiquetas RFID tiene una contraseña por defecto, la cual es 00 00 00 00; la contraseña es de vital importancia para poder modificar el ID de una etiqueta así como también modificar la misma, luego pedirá que ingrese la nueva contraseña, este parámetro debe de componerse de 8 dígitos máximo en formato hexadecimal, y de igual manera el sistema le advertirá si este se ha ingresado de forma errónea, luego la función mandara al módulo RFID la orden de modificar la contraseña.

NOTA: Para que el módulo RFID pueda realizar correctamente el proceso de escribir o modificar la contraseña de una etiqueta, anteriormente debe de haber realizado el proceso de SELECCIONAR ETIQUETA ESPECÍFICA ya que como se mencionó anteriormente al realizar esta función el módulo guarda en su memoria el ID de la etiqueta seleccionada, inmediatamente el usuario debe de realizar el proceso de CAMBIAR LA CONTRASEÑA DE LA ETIQUETA ya que la etiqueta que será modificada será del ID que contendrá la memoria del lector, si el usuario no ejecuta esas dos funciones en ese orden el lector no encontrará la etiqueta a escribir, advertirá del error, ya que en su memoria no habrá ningún ID.

Luego de que el lector reciba la orden, realizara un escaneo para buscar la etiqueta, al encontrarla el módulo modificara la contraseña de la etiqueta, luego mandará al sistema la información si se modificó correctamente o se produjo un error y cual es.

- **MODIFICAR LOS REGISTROS DE LOS ELEMENTOS.**

El sistema es capaz de modificar los registros de los elementos en la base de datos principal, estos son el ID, el NOMBRE y la DESCRIPCIÓN de los mismos, se le advierte al usuario si desea modificar el ID de un elemento, tener en mente que este dato es el que está vinculado con la etiqueta RFID adherida al elemento, y si se modifica, no corresponderá con el registrado en la base de datos principal, en la información de dicho elemento, y ocasionara falsas alarmas a la hora de tratar de encontrar la etiqueta físicamente.

La función pide que ingrese ya sea el NOMBRE o el ID del elemento a modificar para que el sistema pueda individualizarlo, para solo modificar ese. El proceso el cual sigue la modificación de dichos registros; se divide en tres subfunciones cada una tiene la capacidad de poder modificar una de las partes de los registros, estas son:

- **Modifica el NOMBRE del elemento**
- **Modificar el ID del elemento**
- **Modificar la DESCRIPCIÓN del elemento**

La primera subfunción modifica el nombre del elemento específico; la segunda realiza la modificación del ID y la tercera modifica la descripción del mismo, en las sub fusiones de cambiar el ID y el nombre del elemento el ingreso de los datos es obligatorio ya que si el usuario no ingresa estos datos se advertirá del error y no lo dejará avanzar, pero la sub función de modificar la descripción no es obligatorio ingresar los nuevos parámetros, pero cuando no se ingresa se elimina la descripción que estaba anteriormente y la sustituye una celda en blanco; las tres subfunciones siguen el mismo proceso, luego de que se modifiquen los registros se interroga a la base de datos por la existencia del mismo, si no se encuentra dicha información se advertirá del error de lo contrario los datos son presentados con las modificaciones correspondientes, luego dicha información es presentada al usuario para verificar la alteración de datos, y por último se le pregunta si se requiere que se modifique otro elemento, si no se necesita la función termina.

- **PRESTAMOS.**

La presente función contiene las subfunciones, las cuales se explicaran más ampliamente:

- Insertar elemento en la tabla
- Modificar registros de elementos de la tabla
- Eliminar elemento específico

El sistema contiene una base de datos préstamos, la cual contiene registrados todos los elementos que correspondientes a este estado, el cual lo convierte en un sistema más eficiente, esta base de datos describe los artículos y las personas que realizaron el préstamo, dando así un registro más específico, a continuación se describirán las funciones de préstamos:

⇒ **INSERTAR ELEMENTO EN LA TABLA.**

La función le pide al usuario que ingrese los datos correspondientes al nuevo préstamo de un elemento, los cuales son:

- ◇ **ID:** Número de identificación EPC que se le otorgará al elemento, es el mismo número de identificación que posee la etiqueta RFID adherida al elemento.
- ◇ **Nombre de la persona que realiza el préstamo:** Registro del nombre completo de acuerdo a un documento de identificación que presenta la persona que realiza el préstamo..
- ◇ **Nombre del documento de identidad de la persona quien realiza el préstamo:** Es el tipo de documento presentado por la persona que realiza el préstamo.
- ◇ **Número del documento de identidad de la persona quien realiza el préstamo:** Es el número de algún documento de identidad de la persona que realiza el préstamo.
- ◇ **Duración del préstamo:** Es el tiempo que el usuario establezca que sea conveniente que dure el préstamo.
- ◇ **Descripción:** Son datos extras que el usuario decida que son pertinentes para especificar el elemento.

Así también la función advierte si ha cometido algún error al ingresar los datos, y pide que se ingrese de forma correcta, el ID (el cual será el mismo de la etiqueta), nombre de la persona, nombre y/o el número del documento de identidad de la persona que realizará el préstamo del elemento son datos obligatorios, si no los ingresa, se informará que cometió un error y pedirá que ingrese dichos datos, pero la descripción y duración del préstamo no es obligatorio, por ende se tiene la libertad de ingresarlo o no, y se permitirá proseguir sin ningún error.

Luego que la función obtenga los datos pertinentes del préstamo, interroga a la base de datos principal, con el ID de dicho elemento, si está registrado, si no es así se advertirá y no se registrará la información del préstamo, ya que no se podría prestar un elemento que no exista el registro en la base de datos principal; luego de determinar que se encuentra registrado el sistema interrogará a la de préstamos, con el ID de dicho elemento, si ya ha sido registrado anteriormente, se detiene el proceso y se notifica mostrando la información pertinente.

Y si la base de datos de préstamos le notifica que no existe o no ha sido registrado el elemento anteriormente, la función escribirá la información del elemento, luego de que este proceso haya sido completado correctamente se muestra toda la información de dicho elemento, así como también se modifica el registro ESTADO del elemento con el nuevo estado PRESTADO; por último se preguntará si requiere ingresar un nuevo préstamos a la base de datos, si se acepta se reinicia la función de lo contrario la función termina.

⇒ **MODIFICAR LOS REGISTROS DE LOS ELEMENTOS DE LA TABLA.**

El sistema es capaz de modificar los registros de los elementos en la base de datos de préstamos, estos son la descripción del elemento, la información de la persona quien realizó el préstamo y el tiempo que durará.

La función le pide al usuario que ingrese ya sea el nombre o el ID del elemento a modificar para que el sistema pueda individualizarlo, y solo se modifique ese campo. El proceso el cual sigue la modificación de dichos registros de los elementos, se divide en tres subfunciones, cada una tiene la capacidad de poder cambiar una de las partes de la información de los elementos, estas son:

- **Modifica la DESCRIPCIÓN del elemento**
- **Modificar la información de la persona que realizó el préstamo**
- **Modificar el tiempo que durará el préstamo**

La primera subfunción modifica los datos extras que el usuario crea que son convenientes, para tener una mejor descripción del elemento, la segunda modifica el nombre de la persona, el tipo y número del documento de identidad del mismo que realizó el préstamo, esto le da al usuario la libertad de modificar los datos de la persona responsable del elemento en préstamo, y por último la tercera sub función modifica el tiempo en que el operador ha otorgado a la persona el permiso para el préstamo del elemento, en la sub fusiones de cambiar la información de la persona que realiza el préstamo es obligatorio ya que si no se ingresa estos datos la función le advertirá del error y no lo dejará avanzar, pero las subfunciones de modificar la descripción y modificar el tiempo que durará el préstamo no es obligatorio al ingresar la nueva información, cuando el usuario no la ingresa se elimina la descripción y/o el tiempo que se había designado y la sustituye una celda en blanco, las tres subfunciones siguen el mismo proceso que la función MODIFICAR LOS REGISTROS DE LOS ELEMENTOS, con la diferencia que se interroga y modificara los datos de la base de datos de préstamos.

⇒ **ELIMINAR ELEMENTO ESPECÍFICO.**

La función tiene la capacidad de eliminar los registros de un elemento que se encuentra en la base de datos de préstamos, se ingresa ya sea el ID o el NOMBRE del elemento, luego se interroga a la base de datos sobre la existencia del mismo, si responde que no se encuentra registrado, la función

advierte al usuario y se reinicia, pero si responde con la existencia del elemento en la base de datos, prosigue con eliminarlo, si existe un problema al ejecutar el proceso, el sistema le informará del error, pero si el sistema logra eliminar la información, prosigue con la modificación del registro del mismo elemento en la base de datos principal, solo se modifica el registro ESTADO de dicho elemento se cambia de PRESTADO a PRESENTE, y por último el sistema informa al usuario sobre el éxito de su proceso de eliminación de la información en la base de datos de préstamos.

⇒ **ELIMINAR TABLA.**

La función elimina en su totalidad la información de los elementos que se encuentran en la base de datos de préstamos, antes de realizar este proceso realiza la pregunta “Realmente quiere eliminar tabla de préstamos?” esto se realiza para que el usuario se encuentre advertido que si acepta la interrogante se eliminará toda la información, y si la rechaza el sistema saldrá de la función y evitará que el usuario elimine toda la información por accidente, luego de realizar el proceso de eliminar todos los registros se modificará todos los elementos en la base de datos principal en el registro ESTADO cambiando el estado PRESTADO por PRESENTE.

- **ALARMAS.**

La presente función contiene las sub funciones, las cuales se explicaran más ampliamente:

- Eliminar elemento de la tabla
- Eliminar tabla

El sistema contiene una base de datos de alarmas, esta contiene la información de los elementos que no están presentes, y que se extrajeron sin permiso del usuario, en esta la información se ingresa cuando se realiza un escaneo de inventario, el sistema al no detectar un elemento guarda la información en la base de datos de alarmas, teniendo así una mejor documentación sobre la existencia de los elementos a controlar, dándole así al usuario un mejor y eficiente registro de los elementos.

⇒ **ELIMINAR ELEMENTO ESPECÍFICO.**

Realiza el mismo proceso que la función ELIMINAR ELEMENTO ESPECÍFICO de la opción de PRESTAMOS, con la diferencia que dicho proceso lo realiza en la base de datos de alarmas y luego de eliminar la información del elemento, prosigue con la modificación de los datos del mismo elemento en la base de datos principal, solo se modifica el registro ESTADO se cambia de AUSENTE por PRESENTE.

⇒ **ELIMINAR TABLA.**

Realiza el mismo proceso que la función ELIMINAR TABLA de la opción de PRÉSTAMOS, con la diferencia que se eliminara toda la información de la base de datos de alarmas, luego el sistema modificará todos los elementos en la base de datos principal en el registro ESTADO cambiando el estado AUSENTE por PRESENTE.

- **USUARIOS.**

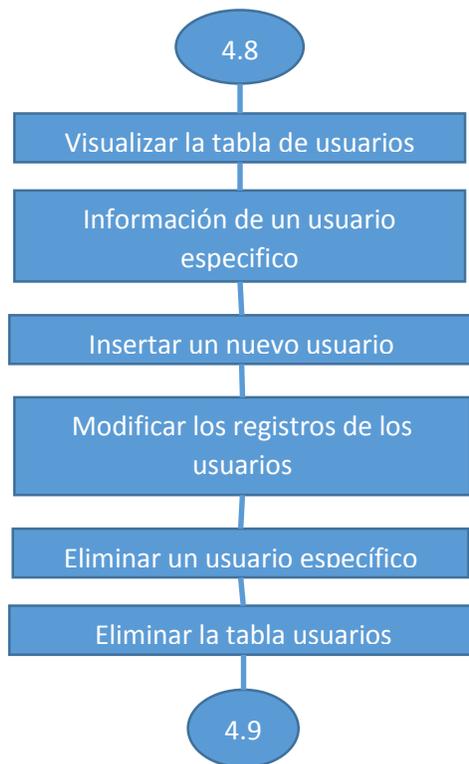


Figura 2.47: Diagrama de flujo del funcionamiento del proceso de usuarios.

El sistema contiene una base de datos de usuarios, esta contiene toda la información pertinente a los usuarios que tienen permitido interactuar con las funciones del sistema, también abarca el tipo de permisos que se le da a cada uno, se debe tener muy en cuenta que existe un tercer tipo de usuario (además de los usuarios y los súper usuarios), el cual tiene los permisos de super usuario por defecto y que no se puede eliminar, este usuario es SYSRFID y su password es SYSRFID este es utilizado cuando la base de datos de usuario este sin ningún registro, así los usuarios tendrán un punto de acceso por defecto, es un tipo de “puerta trasera”, por eso se les advierte que se utilicen moderadamente. A continuación se describirán las sub funciones que posee la función de usuarios:

⇒ **VISUALIZAR LA TABLA DE USUARIOS.**

Realiza el mismo proceso que la función VISUALIZAR DATOS, con la diferencia que se extrae la información de la base de datos de usuarios; esta base de datos es donde el sistema guarda toda la información pertinente a los usuarios registrados, esta información se divide en:

- ◇ **Nombres:** Son los nombres del usuario.
- ◇ **Apellidos:** Son los apellidos del usuario.
- ◇ **Nombre del documento de identidad:** Es el tipo de documento que corrobore la identidad del usuario.
- ◇ **Número del documento de identidad:** El número de un documento que corrobore la identidad del usuario.
- ◇ **USUARIO:** Es un nombre que se le otorga al usuario para que el sistema lo identifique al ingresar al sistema.

- ◇ **PASSWORD:** Es una palabra clave que el usuario otorgara para identificarlo con el sistema al querer ingresar a los menús.
- ◇ **Permisos:** Son los permisos que el sistema le otorga al usuario para el acceso a diferentes partes del mismo, los permisos son:
 - **USUARIO:** El permiso le otorga al usuario acceso al sistema, pero le permite la interacción solamente con las funciones que abarca el menú principal.
 - **SUPERUSER:** El permiso le otorga al usuario acceso al sistema, le permite el acceso a todo el sistema, le permite la iteración sobre las funciones del menú principal y al de opciones avanzadas.
- ◇ **Fecha y Hora:** Le informa la hora y la fecha en que el usuario fue ingresado a la base de datos de usuarios del sistema.

⇒ **INFORMACIÓN DE UN USUARIO ESPECÍFICO.**

Realiza el mismo proceso que la función INFORMACIÓN DE ETIQUETA ESPECÍFICA, con la diferencia que se interroga a la base de datos de usuarios, y la función pide el ingreso ya sea del nombre, apellido, el número del documento de identidad o el nombre de USUARIO, del usuario específico, advierte si ha cometido algún error a la hora de ingresar la información, después que se ingrese los datos pertinentes, se procede a interrogar a la base de datos con dicha información (una en específico). Si al interrogar a la base de datos sobre la información de un usuario específico, y este no se encuentra registrado, advierte que dicha información no existe.

⇒ **INSERTAR UN NUEVO USUARIO.**

La función pide que se ingrese los datos correspondientes al nuevo usuario, los cuales son:

- ◇ **Nombres:** Son los nombres del nuevo usuario.
- ◇ **Apellidos:** Son los apellidos del nuevo usuario.
- ◇ **Nombre del documento de identidad:** Es el tipo de documento que corrobore la identidad del usuario.
- ◇ **Número del documento de identidad:** El número de un documento que corrobore la identidad del usuario.
- ◇ **USUARIO:** Es un nombre que se le otorga al usuario para que el sistema lo identifique al ingresar al sistema.
- ◇ **PASSWORD:** Es una palabra clave que el usuario otorgara para identificarlo con el sistema al querer ingresar a los menús.
- ◇ **Permisos:** Son los permisos que el sistema le otorga al usuario para el acceso a diferentes partes del mismo, los permisos son:
 - **USUARIO:** El permiso le otorga al usuario acceso al sistema, pero le permite la interacción solamente con las funciones que abarca el menú principal.
 - **SUPERUSER:** El permiso le otorga al usuario acceso al sistema, le permite el acceso a todo el sistema, le permite la iteración sobre las funciones del menú principal y al de opciones avanzadas.

Así también la función advierte si ha cometido algún error al ingresar los datos, y le pide que se ingrese de forma correcta, todos los datos son obligatorios, si no se digitan, la función le comunicara que cometió un error y le pedirá que los ingrese.

Luego que la función obtenga los datos pertinentes del nuevo usuario, interroga a la base de datos de usuarios, con el nombre de USUARIO, si este ya ha sido registrado anteriormente, si es así se detiene el proceso y notifica mostrándole la información pertinente sobre la existencia de dicho usuario, luego realiza el mismo proceso que la función INSERTAR ELEMENTO A LA BASE DE DATOS con la diferencia que se realiza en la base de datos de usuarios.

⇒ **MODIFICAR LOS REGISTROS DE LOS USUARIOS.**

El sistema es capaz de modificar los registros de los usuarios en la base de datos, estos son el nombre de USUARIO, PASSWORD o los datos personales del usuario.

La función pide que ingrese el nombre de USUARIO a modificar para que el sistema pueda individualizarlo, para que solo modifique ese. El proceso el cual sigue la modificación de dichos registros de los usuarios, se divide en tres subfunciones cada una tiene la capacidad de poder modificar una de las partes de los registros, estas son:

- **Modifica los datos personales de un usuario**
- **Modificar el nombre de USUARIO**
- **Modificar el PASSWORD del usuario**

La primera sub función modifica los nombres, apellidos del usuario, el tipo y número del documento de identidad del mismo; la segunda modifica el nombre con que el sistema identifica al usuario al ingresar al mismo; y la tercera modifica la contraseña que utiliza el usuario para el ingreso del sistema así como también de los diferentes menús; en la sub funciones de cambiar la información personal del usuario, el ingreso de la nueva información de la persona (los nombres, apellidos del usuario, el nombre y el número del documento de identidad), es obligatorio ya que si no ingresa estos datos se advertirá del error y no lo dejará avanzar, así como también es obligatorio la información del nuevo nombre de USUARIO o PASSWORD, la información de cada una es por individual; las tres subfunciones siguen el mismo proceso que la función MODIFICAR LOS REGISTROS DE LOS ELEMENTOS, con la diferencia que se interroga y modificara los datos de la base de datos de usuarios.

⇒ **ELIMINAR REGISTROS DE USUARIO ESPECÍFICO.**

Realiza el mismo proceso que la función ELIMINAR ELEMENTO ESPECÍFICO de la opción de PRESTAMOS, con la diferencia que dicho proceso lo realiza en la base de datos de usuarios, y se ingresa el nombre de USUARIO para individualizar la información del usuario a eliminar.

⇒ **ELIMINAR TODOS LOS USUARIOS.**

Realiza el mismo proceso que la función ELIMINAR TABLA de la opción de PRÉSTAMOS, con la diferencia que se eliminara toda la información de la base de datos de usuarios.

- **USUARIOS SMS.**



Figura 2.48: Diagrama de flujo del funcionamiento del proceso de usuarios SMS.

El sistema contiene una base de datos de usuarios SMS, esta base de datos contiene toda la información pertinente a los usuarios a los cuales se les mandará un mensaje de texto GSM/SMS de alarma, cada vez que el sistema realiza un inventario y existen elementos registrados en la base de datos principal que no fueron detectados, también cuando se realiza un escaneo y se detecta una salida no autorizada de elementos en la zona a controlar, este proceso es controlado por la interfaz GSM del sistema, a continuación se describirán las subfunciones que posee la función de usuarios SMS:

⇒ **VISUALIZAR LA TABLA DE USUARIOS SMS.**

Realiza el mismo proceso que la función VISUALIZAR DATOS, con la diferencia que se extrae la información de la base de datos de usuarios SMS; la base de datos es donde el sistema guarda toda la información pertinente a los usuarios a los cuales se les enviará los mensajes de texto de alerta, esta información se divide en:

- ◇ **Nombres:** Son los nombres del usuario.
- ◇ **Apellidos:** Son los apellidos del usuario.
- ◇ **Nombre del documento de identidad:** El tipo de documento que corrobore la identidad del usuario.
- ◇ **Número del documento de identidad:** El número de un documento que corrobore la identidad del usuario.
- ◇ **Número de teléfono:** Es el número de teléfono al cual el sistema GSM le enviará los mensajes de texto.

- ◇ **Comentario:** Es información extra que el usuario del sistema haya puesto por su relevancia para la identificación del usuario SMS.
- ◇ **Fecha y Hora:** Le informa la hora y la fecha en que el usuario fue ingresado a la base de datos de usuarios SMS del sistema.

⇒ **INFORMACIÓN DE UN USUARIO SMS ESPECÍFICO.**

Realiza el mismo proceso que la función INFORMACIÓN DE ETIQUETA ESPECÍFICA, con la diferencia que se interroga a la base de datos de usuarios SMS y pide el ingreso ya sea del nombre, el apellido del usuario SMS a buscar, el número del documento de identidad o el número de teléfono, del mismo.

⇒ **INSERTAR UN NUEVO USUARIO SMS.**

La función pide que ingrese los datos correspondientes al nuevo usuario SMS, los cuales son:

- ◇ **Nombres:** Son los nombres del nuevo usuario SMS.
- ◇ **Apellidos:** Son los apellidos del nuevo usuario SMS.
- ◇ **Nombre del documento de identidad:** El tipo de documento que corrobore la identidad del usuario.
- ◇ **Número del documento de identidad:** El número de un documento que corrobore la identidad del usuario.
- ◇ **Número de teléfono:** Es el número de teléfono al cual el sistema GSM le enviará los mensajes de texto.
- ◇ **Comentario:** Es información extra que el usuario del sistema haya puesto por su relevancia para la identificación del usuario SMS.

Así también advierte si ha cometido algún error al ingresar los datos, y le pide que se ingrese de forma correcta, todos los datos son obligatorios para la función, si el usuario no los ingresa, le comunicara que cometió un error y le pedirá que ingrese dichos datos.

Luego que se obtenga los datos pertinentes del nuevo usuario, interroga a la base de datos de usuarios SMS, con el número de teléfono, si ya ha sido registrado anteriormente, si es así se detiene el proceso y notifica mostrando la información pertinente sobre la existencia de dicho usuario SMS.

Y si responde a la función que no existe o no ha sido registrado anteriormente, la función escribe o registra la información del usuario SMS en la base de datos, luego de que este proceso haya sido completado correctamente el sistema muestra toda la información que ha sido registrada; por último la función pregunta si requiere ingresar otro nuevo usuario SMS, si se acepta el sistema reinicia la función y si no la función termina.

⇒ **MODIFICAR LOS REGISTROS DE LOS USUARIOS SMS.**

El sistema es capaz de modificar los registros de los usuarios SMS en la base de datos, estos registros son el número de teléfono, comentario o los datos personales del mismo. La función pide que ingrese el número de teléfono del usuario SMS a modificar para que el sistema pueda individualizar la información, para que solo modifique ese. El proceso se divide en tres subfunciones cada una tiene la capacidad de poder modificar una de las partes de los registros de los usuarios SMS, estas son:

- **Modifica los datos personales de un usuario SMS**
- **Modificar el NUMERO DE TELEFONO**
- **Modificar comentario**

La primera sub función modifica los nombres, apellidos del usuario, el nombre y el número del documento de identidad; la segunda modifica el número telefónico con que el sistema mandará el mensaje de texto de alarma sobre las etiquetas no detectadas durante un inventario y salidas no autorizadas de elementos durante un escaneo; y por último la tercera sub función modifica la información extra que se ingresa para identificar específicamente al usuarios SMS; en la sub función de cambiar la información personal del usuario SMS, la nueva información de la persona (los nombres, apellidos del usuario, el nombre y el número del documento de identidad), y sobre el nuevo número telefónico, es obligatorio ya que si no ingresa estos datos la función le advertirá del error y no lo dejará avanzar, y la información del nuevo comentario no es obligatorio, si no se ingresa la información el sistema borra la celda donde se encuentra dicha información y la deja en blanco; luego realiza el mismo proceso que la función INSERTAR ELEMENTO A LA BASE DE DATOS con la diferencia que se realiza en la base de datos de usuarios SMS.

⇒ **ELIMINAR REGISTROS DE USUARIO SMS ESPECÍFICO.**

Realiza el mismo proceso que la función ELIMINAR ELEMENTO ESPECÍFICO de la opción de PRESTAMOS, con la diferencia que dicho proceso lo realiza en la base de datos de usuarios SMS, y se ingresa el nombre de USUARIO para individualizar la información del usuario SMS a eliminar.

⇒ **ELIMINAR TODOS LOS USUARIOS SMS.**

Realiza el mismo proceso que la función ELIMINAR TABLA de la opción de PRÉSTAMOS, con la diferencia que se eliminara toda la información de la base de datos de usuarios SMS.

- **ELIMINAR REGISTROS.**

La presente función contiene las sub funciones, las cuales se explicaran más ampliamente:

- **ELIMINAR ELEMENTO**
- **ELIMINAR TODO**

El sistema tiene la capacidad de eliminar los registros de la base de datos principal, así se le otorga al usuario la libertad de eliminar los registros de los elementos que ya no se tomarán en cuenta en la documentación al realizar el inventario, y así no causará problemas a la hora de identificar los elementos que se encuentran y los que no, a continuación se describirá la sub funciones que realizan dicho proceso:

⇒ **ELIMINAR ELEMENTO.**

Realiza el mismo proceso que la función ELIMINAR ELEMENTO ESPECÍFICO de la opción de PRESTAMOS, con la diferencia que dicho proceso lo realiza en la base de datos principal, y se ingresa el NOMBRE o ID para individualizar la información del elemento a eliminar, pero si se

logra eliminar la información del elemento, prosigue con la modificación del registro del mismo elemento en la base de datos de préstamo y de alarmas (si dicha información se encuentra registrada en esas bases de datos), se elimina también la información del elemento en las bases de datos de préstamos y alarmas, y por último se informa sobre el éxito de su proceso de eliminación.

⇒ **ELIMINAR TODO.**

Realiza el mismo proceso que la función ELIMINAR TABLA de la opción de PRÉSTAMOS, con la diferencia que se eliminara toda la información de la base de datos de usuarios SMS, luego de realizar el proceso de eliminar de todos los registros de los elementos en la base de datos principal, se borra toda la información de la bases de datos de préstamos y de alarmas, ya que estas bases de datos están relacionadas con la principal ya que los elementos que se prestan y/o se genera las alarmas son los elementos registrados en la base de datos principal, por ende el usuario debe de tener mucho cuidado al realizar la presente función ya que eliminaría todos los datos registrados en el sistema relacionados con los elementos a controlar.

CAPÍTULO 3: Análisis y propuestas de diseño.

Introducción

El presente capítulo describe el desempeño del prototipo propuesto, para lo cual se realizaron diferentes pruebas en diferentes espacios al interior de la EIE. El objetivo de dichas pruebas, fue caracterizar el funcionamiento del módulo, así mismo su viabilidad como dispositivo de control de ubicación de equipos.

Las pruebas consistieron en la verificación de cobertura, cotejo de información asociada a cada tag, detección de movimientos de equipos y validación de interfaz de usuario. En función de lo anterior, se determinó ocupar la información recopilada en el capítulo 2, en cuanto a la caracterización de las diferentes antenas y utilizar para estas pruebas, la antena de mejor desempeño, siendo esta la antena tipo parche circular.

3.1 Descripción de las Pruebas

Para el análisis de la viabilidad de la solución propuesta, así como verificación de su desempeño, se siguieron los siguientes pasos:

- Pruebas de detección de equipos con Tag asignada, en diferentes ubicaciones
- Pruebas generales de la solución propuesta

3.1.1 Pruebas de detección de Tag

En primera instancia se deben especificar las zonas al interior de la EIE objeto de estas pruebas, lo cual es mostrado en las figuras 3.1 y 3.2.

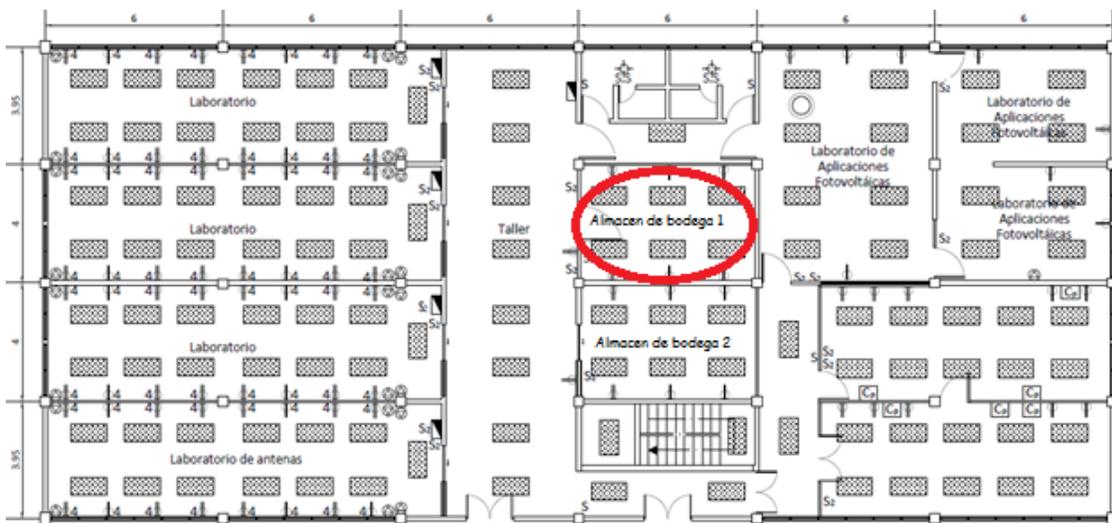


Figura 3.1: Primera planta del edificio de EIE.

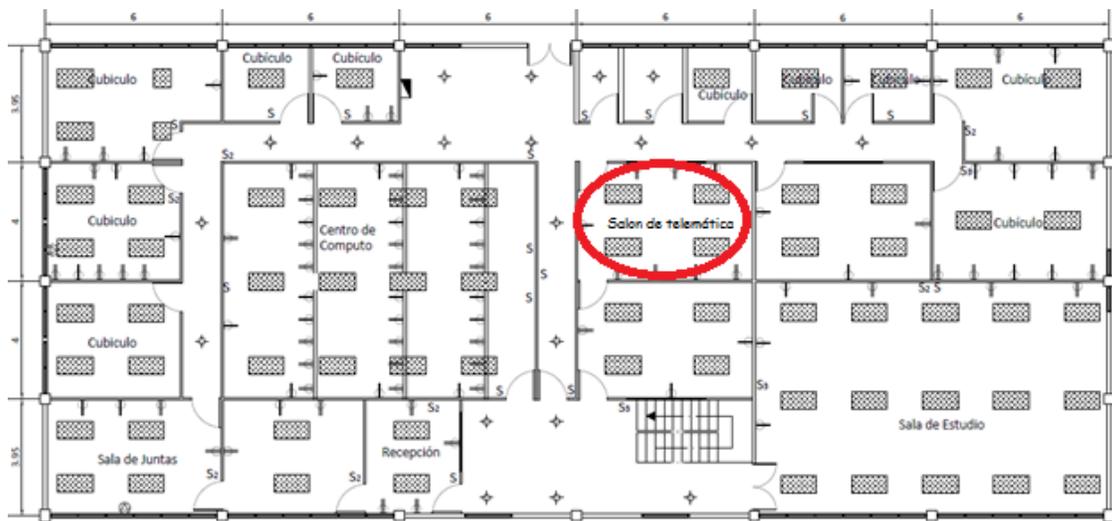


Figura 3.2: Segunda planta del edificio de EIE.

En cuanto al salón o laboratorio de telemática, la figura 3.3 muestra una imagen del mismo; mientras que la figura 3.4 describe las ubicaciones utilizadas para colocar la antena asociada al lector, desde la cual se tomaron las lecturas de cobertura.



Figura 3.3

3.1.2 Propuesta de solución

3.1.2.1 Control de seguridad de equipos de laboratorio

El sistema de control de seguridad permitirá monitorear equipo importante, como primer lugar en el salón de telemática del edificio de la EIE y en segundo lugar el salón de bodega 1 de la EIE. En la figura 3.4 observamos las diferentes posiciones, del salón de telemática, donde se decidió colocar el módulo lector, para cubrir todos los puntos donde están los equipos con tag's asignadas.

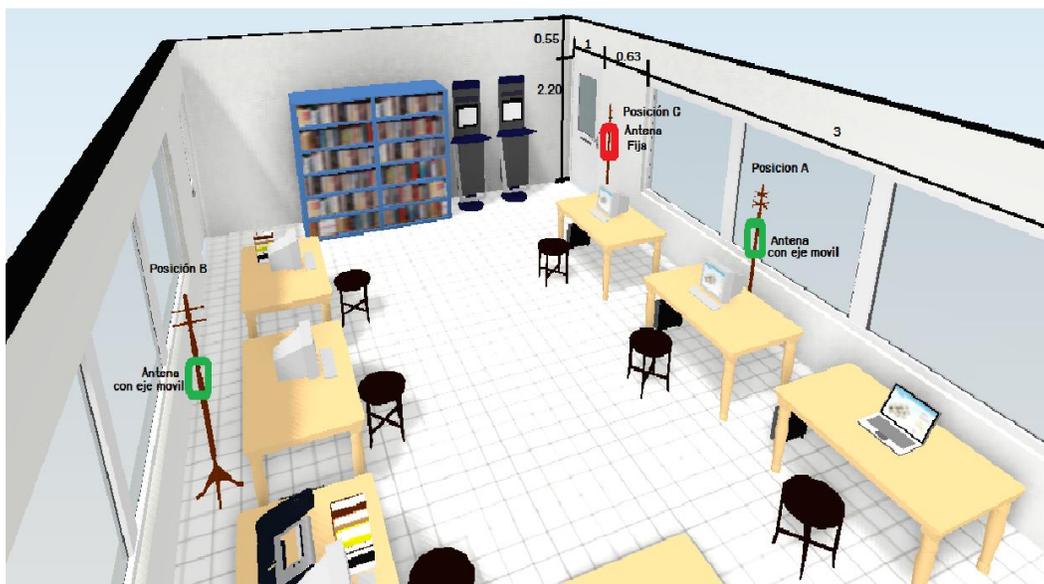


Figura 3.4

La antena se colocara sobre un mástil, que la pueda sostener. Según la figura anterior, se propone colocar dos unidades móviles y una fija, de módulos lectores; lo que se detalla a continuación:

3.1.2.1.1 Dos unidades móviles de módulos lectores:

Justificación: Se decidieron colocar en las posiciones A y B mostradas en la Figura 3.4, debido a que fueron los puntos que permitían cubrir las zonas donde los objetos están más expuestos al libre tránsito de los usuarios; y ya que se realizaron diferentes mediciones en el salón de telemática, se pudo corroborar en cuánto a economía del proyecto⁴⁴ y a su vez zona de cobertura⁴⁵; la propuesta siguiente: Lo que se plantea es un sistema móvil de escaneo, de la siguiente forma: en un motor paso a paso (ó similar, pudiese ser un mástil móvil) a través de un acople, colocar la antena; y este subsistema (motor y antena) se pondría en el mástil. Este tipo de motores lo que realiza básicamente es girar un determinado ángulo, con un tiempo de retardo específico y de esta forma lograr realizar un escaneo para la aplicación requerida. Con un motor paso a paso que pueda girar, en 3 posiciones específicas, se podrá lograr una cobertura más amplia en cuanto a la capacidad de escaneo de la antena, todo esto sincronizado con la parte de control del sistema RFID en general.

Limitantes: Entre esta se tienen que no se construyó de forma automática el sistema móvil propuesto (es decir no se adaptaron al sistema, los motores paso a paso); sino que se fue realizando manualmente el movimiento de la antena con el mástil, dentro del área a cubrir.

- Descripción de las pruebas, para antenas con eje móvil.

✓ Posición A

La figura 3.5 nos muestra en detalle, las 3 posiciones A1, A2 y A3 en las que la antena radiara y podrá detectar las tags específicas.

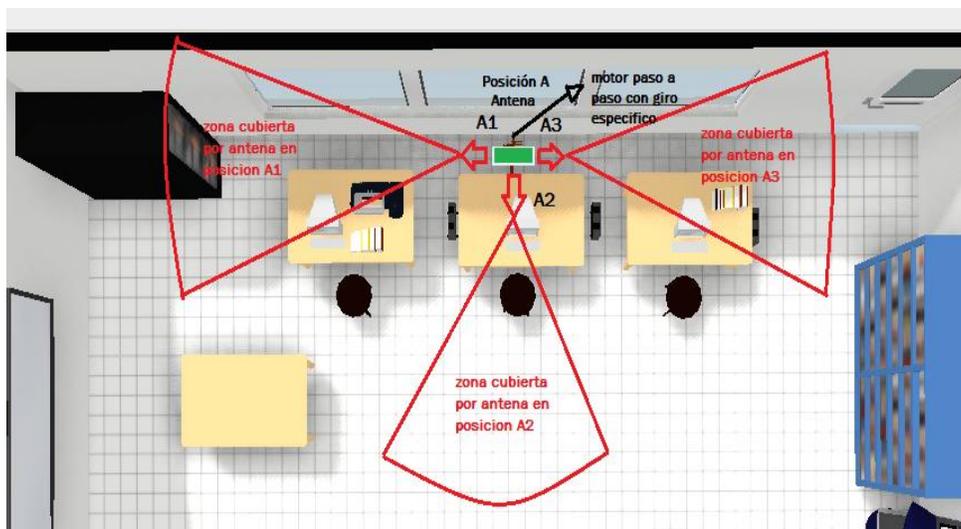


Figura 3.5

➤ Plano Horizontal

La figura 3.6, nos muestra la prueba, para las lecturas tomadas en el plano horizontal.

¹²Ver apartado 3.1.2.1.3 para opciones descartadas, como posibles soluciones al control de seguridad con pruebas realizadas

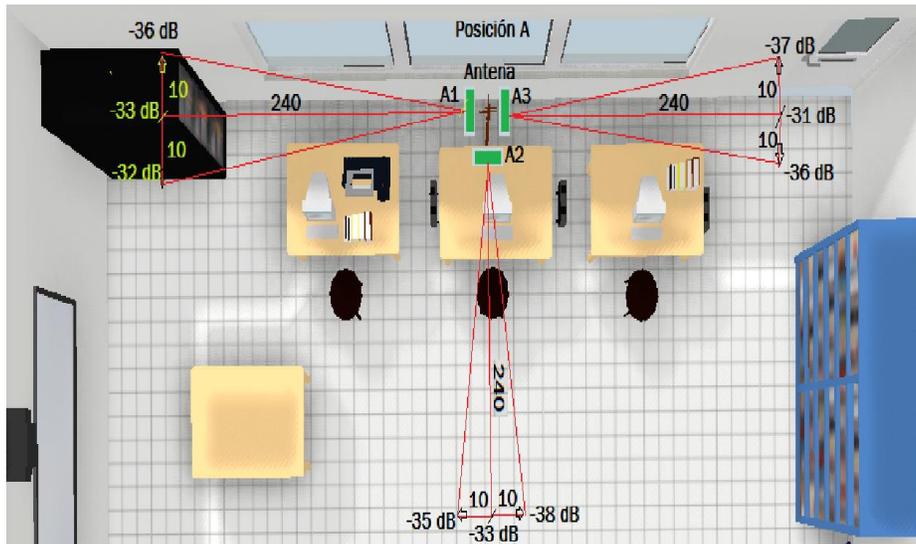


Figura 3.6

La prueba se efectuó de la siguiente forma: ya conocida la distancia máxima a la que la antena lee la tag, que es de 240 cms, obtenida con la caracterización de la antena parche circular, en el capítulo 2 de este documento; se movía esta tanto a la derecha como a la izquierda del eje en que era detectada, permitiendo conocer las distancias máximas de apertura, que es igual a 10 cms; con su correspondiente lectura en dB. Superior a 10 cms (-38 dB) tanto a la derecha, como a la izquierda, la antena ya no detecta la tag. Se pudo comprobar que la apertura horizontal, con respecto al eje de la antena, es prácticamente simétrico, ya que era poca la variación de las distancias.

Luego se procedió a escoger otras 2 distancias como puntos característicos, el primer punto se muestra en la Figura 3.7 (con la antena en la posición A2 proporcionada por el motor paso a paso) y fue 115 cms, obteniendo la distancia máxima de apertura de 60 cms.

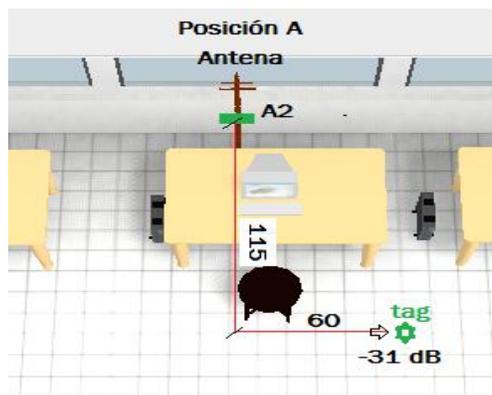


Figura 3.7

Siempre en la posición A2 de la antena. El segundo punto escogido fue 200 cms, obteniendo la distancia máxima de apertura de 50 cms, tal como se muestra en la Figura 3.8

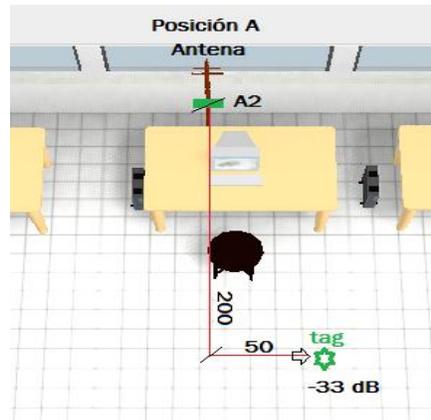


Figura 3.8

La tabla 3.1 resume las lecturas de aberturas horizontales obtenidas con el lector en la posición A.

Distancia en Línea vista (cms) Abertura Horizontal (cms)	Posición A1 (niveles obtenidos dB)						Posición A2 (niveles obtenidos dB)						Posición A3 (niveles obtenidos dB)					
	115		200		240		115		200		240		115		200		240	
	Dcha	Izq.	Dcha.	Izq.	Dcha	Izq.	Dcha	Izq.	Dcha	Izq.	Dcha	Izq.	Dcha	Izq.	Dcha	Izq.	Dcha.	Izq.
60	-35	-33	-35	-34	-37	-35	-32	-33	-31	-35	-35	-35	-34	-36	-32	-31	-34	-32
50	-30	-34	-35	-35	-33	-36	-31	-36	-36	-37	-33	-34	-36	-37	-33	-34	-35	-31
10	-36	-33	-31	-32	-36	-32	-33	-38	-35	-39	-35	-38	-38	-31	-34	-33	-37	-36

Tabla 3.1

➤ Plano Vertical

Para estas pruebas se tomaron los mismos tres puntos característicos considerados en el plano horizontal. La figura 3.9 muestra una representación de la prueba a 200 cms, para las medidas de alturas.

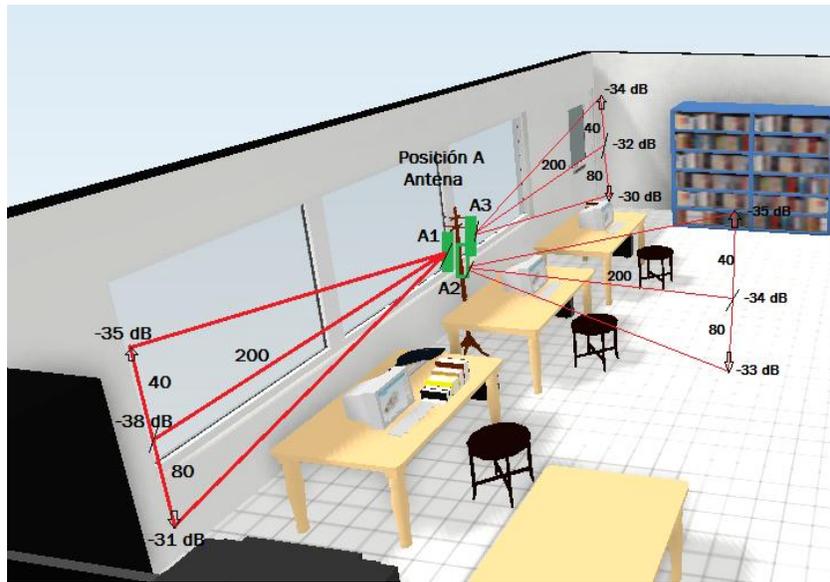


Figura 3.9

La prueba se realizó de la siguiente manera: la antena se colocó sobre un mástil a 0.90 metros con respecto al piso del salón y teniendo en cuenta que la altura de la base de los muebles es de aproximadamente 0.7 metros; se prosiguió midiendo el alcance del módulo y las etiquetas colocadas en objetos; pero ahora en el plano vertical (pudiéndose conocer ángulos de altura), es decir, conociendo la lectura a 200 cms, se procedió a mover el artículo con su respectiva etiqueta, pudiéndose conocer la abertura máxima vertical, que fue 40 cms hacia arriba del eje de referencia de la antena y 80 cms hacia abajo. Superior a 40 cms (-35 dB) hacia arriba y 80 cms (-33 dB) hacia abajo, la antena ya no detecta la tag.

Luego se procedió el mismo procedimiento descrito en el párrafo anterior a 115 cms, la Figura 3.10 nos muestra las máximas aberturas verticales alcanzadas, con la antena en la posición A2, que fueron de 70 cms hacia arriba y 100 cms hacia abajo.

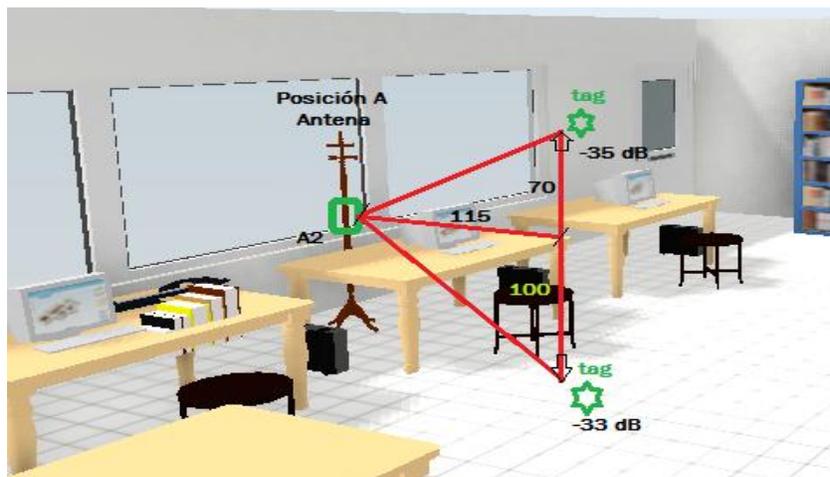


Figura 3.10

A la distancia máxima alcanzada 240 cms, la Figura 3.11 nos muestra la lectura de la máxima altura alcanzada, que fue de 55 cms hacia abajo del eje de la antena.

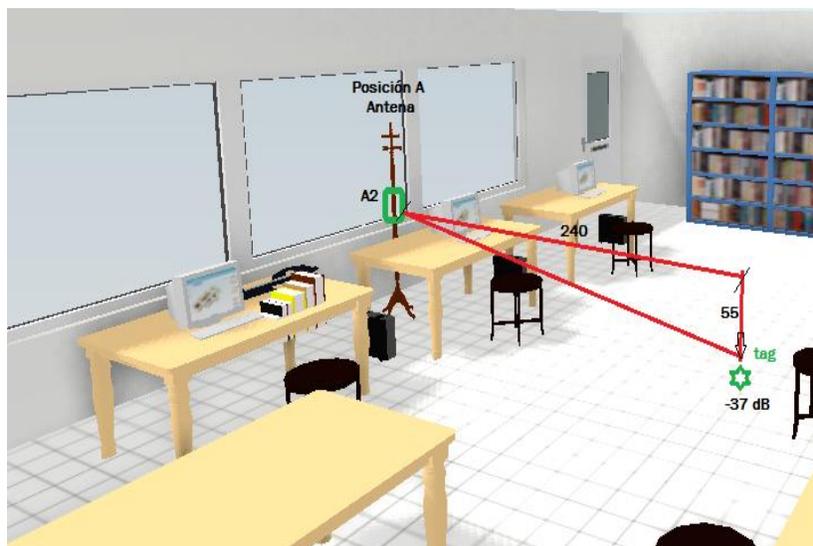


Figura 3.11

Cabe mencionar que a esta distancia máxima, la antena solo lee la tag hacia debajo de su eje horizontal y ya no hacia arriba. Hacia abajo de 55 cms (-37 dB), la antena ya no lee la tag.

La tabla 3.2 resume las lecturas de aberturas verticales, obtenidas con el lector, en la posición A.

Distancia línea vista (cms) \ Abertura vertical (cms)	Posición A1 (niveles obtenidos dB)						Posición A2 (niveles obtenidos dB)						Posición A3 (niveles obtenidos dB)					
	115		200		240		115		200		240		115		200		240	
	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo
70	-35	-33	-35	-34	-37	-35	-32	-33	-31	-35	-35	-35	-34	-36	-32	-31	-34	-32
100	-30	-34	-35	-35	-33	-36	-31	-36	-36	-37	-33	-34	-36	-37	-33	-34	-35	-31
40	-36	-33	-35	-31	-34	-37	-33	-38	-35	-33	-32	-33	-38	-31	-34	-30	-37	-36
80	-34	-32	-39	-34	-33	-37	-39	-37	-38	-32	-35	-36	-37	-38	-39	-34	-33	-32
55	-35	-37	-38	-31	-32	-31	-36	-34	-39	-31	-35	-36	-35	-39	-37	-32	-33	-31

Tabla 3.2

✓ Posición B

Los pasos expuestos para la posición A, se repitieron para la posición B.

➤ Plano Horizontal

La Tabla 3.3 muestra las lecturas de abertura horizontal.

Distancia en Línea vista (cms)	Posición B1 (niveles obtenidos dB)						Posición B2 (niveles obtenidos dB)						Posición B3 (niveles obtenidos dB)					
	115		200		240		115		200		240		115		200		240	
Abertura Horizontal (cms)	Dcha	Izq.	Dcha.	Izq.	Dcha	Izq.	Dcha	Izq.	Dcha	Izq.	Dcha	Izq.	Dcha	Izq.	Dcha	Izq.	Dcha.	Izq.
60	-35	-33	-35	-34	-37	-35	-32	-33	-31	-35	-35	-35	-34	-36	-32	-31	-34	-32
50	-32	-34	-35	-35	-33	-36	-37	-36	-36	-37	-33	-39	-35	-38	-33	-34	-35	-31
10	-36	-33	-31	-32	-36	-32	-33	-38	-35	-39	-35	-39	-34	-37	-34	-33	-37	-36

Tabla 3.3

➤ Plano vertical

La Tabla 3.4 muestra las lecturas de abertura vertical.

Distancia línea vista (cms)	Posición B1 (niveles obtenidos dB)						Posición B2 (niveles obtenidos dB)						Posición B3 (niveles obtenidos dB)					
	115		200		240		115		200		240		115		200		240	
Abertura vertical (cms)	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo
70	-31	-34	-36	-38	-39	-33	-31	-36	-36	-38	-39	-32	-34	-34	-30	-30	-31	-30
100	-32	-33	-35	-38	-33	-39	-31	-36	-36	-37	-33	-34	-36	-35	-31	-33	-39	-36
40	-36	-33	-35	-32	-34	-36	-33	-38	-35	-33	-32	-33	-38	-36	-32	-32	-38	-37
80	-34	-32	-39	-31	-33	-34	-39	-37	-38	-32	-35	-36	-37	-37	-33	-31	-36	-33
55	-35	-37	-38	-30	-32	-32	-36	-34	-39	-31	-35	-36	-35	-38	-34	-30	-34	-33

Tabla 3.4

3.1.2.1.2 Una unidad fija de módulo lector:

Justificación: Se decidió colocar un lector fijo en la posición C mostrada en la Figura 3.12, la antena radiaría en forma tal, que se cubra la zona de la puerta de acceso al salón.

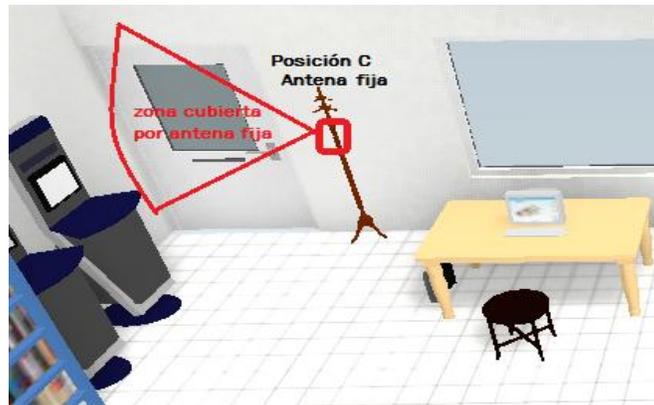


Figura 3.12

Se propone colocar aquí, debido a que con esta posición, se lograría controlar la entrada y salida de los elementos con tag's asignadas al salón y de esta forma resguardar, todos los artículos, no cubiertos por las antenas con eje móvil, por ejemplo los objetos, que se encuentran en el archivero y el estante, mencionar que según la Figura 3.4, la puerta de acceso mide 2.20 metros de alto por 1 metro de ancho, esto sirvió de partida junto con las distancias alcanzadas en el apartado 3.1.2.1.1. A continuación se describen las pruebas.

- **Descripción de las pruebas, para antena con eje fijo.**

✓ Posición C

Los pasos expuestos para una de las posiciones de la antena A (por ejemplo la A3) se repitieron para la posición C.

➤ Plano horizontal

Las lecturas de abertura horizontal de la posición C, se resumen en la Tabla 3.5.

Distancia línea vista (cms)	Posición C (niveles obtenidos dB)					
	115		200		240	
Abertura Horizontal (cms)	Dcha.	Izq.	Dcha.	Izq.	Dcha.	Izq.
60	-36	-32	-31	-33	-32	-34
50	-32	-30	-38	-34	-31	-33
10	-32	-31	-37	-35	-32	-31

Tabla 3.5

➤ Plano Vertical

La Tabla 3.6 muestra las lecturas de abertura vertical.

Distancia línea vista (cms) Abertura vertical (cms)	Posición C (niveles obtenidos dB)					
	115		200		240	
	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo
70	-32	-39	-37	-31	-34	-32
100	-33	-37	-38	-32	-37	-35
40	-32	-35	-34	-38	-39	-38
80	-33	-36	-33	-33	-36	-39
55	-36	-38	-32	-32	-33	-35

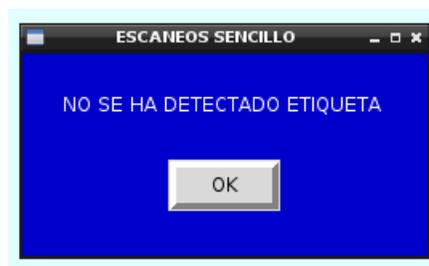
Tabla 3.6

- **Resultado de las pruebas, para antena con eje fijo y con eje móvil**

El sistema prototipo posee dos tipos de interfaces para poder interactuar: una local y otra WEB.

El sistema también tiene la funcionalidad de seguridad, este se basa en que este se coloca en lugares de acceso que se quieran controlar, por ejemplo puertas, luego se realizan una serie de escaneos, y si se detecta un elemento, se interroga a la base de datos de préstamos si se tiene permiso de extraerlo de la zona a controlar el sistema no realiza ninguna acción y si no, se genera una alarma y se manda un mensaje al teléfono celular a un usuario específico, informando de una salida no autorizada; la función de seguridad genera una alarma si detecta una etiqueta ya que esto significa que se desea extraer un elemento del lugar (si esta no tiene permiso de ser sacada). La función principal de esta es ofrecerle al usuario un control más preciso del flujo de entrada y salida de elementos de la zona a controlar, este proceso se puede realizar de dos formas, a petición de usuario, esto quiere decir que él decide cuándo y cuantos escaneos se realizaran y de forma automática el cual el sistema genera múltiples escaneos y se detiene cuando se detecta un elemento, el cual no tiene permiso.

A continuación se muestra un ejemplo del funcionamiento del sistema, con el proceso de seguridad, mostrando primero un escaneo donde no se detecta etiqueta, y luego uno donde se detecta una etiqueta que no posee permiso para salir, dicho escaneo se realizó con las dos interfaces del sistema,





b)

Figura 3.13: Se realizan un escaneo y no se detectó ninguna etiqueta, a) fue realizado con la interface principal, b) fue realizado con la interface WEB.

Las imágenes anteriores muestran los resultados de la realización de escaneo de seguridad, y el resultado del mismo en la dos interfaces del sistema, mostrando que el acceso al lugar nadie se sacara algún elemento sin permiso o que el acceso esta libre, pero en la siguiente figura se muestra la reacción y la ejecución de las alarmas cuando el sistema detecta un elemento sin permiso de salir,



a)



b)

Figura 3.14: Se realizó un escaneo y se detectó una etiqueta, la cual no tenía autorización para salir del lugar a controlar, a) fue realizado con la interface principal, b) fue realizado con la interface WEB.

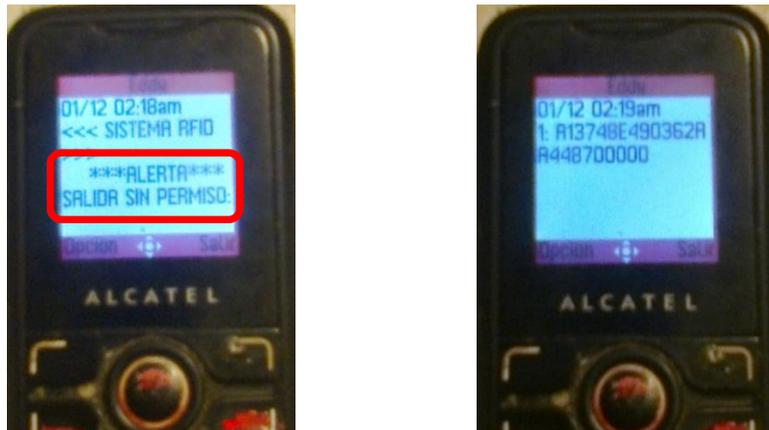


Figura 3.15: Mensaje de texto de alarmas del sistema prototipo RFID dirigidos a un usuario específico, al realizar un escaneo de seguridad.

3.1.2.1.3 Cobertura alcanzada con los 3 módulos lectores propuestos:

Antes de proponer ocupar 1 lector fijo y 2 móviles; se realizaron diferentes pruebas con las que se determinó zona de cobertura, por ejemplo, la figura 3.16 nos muestra una primera prueba, donde se detalla las zonas de cobertura por cada módulo lector fijo.

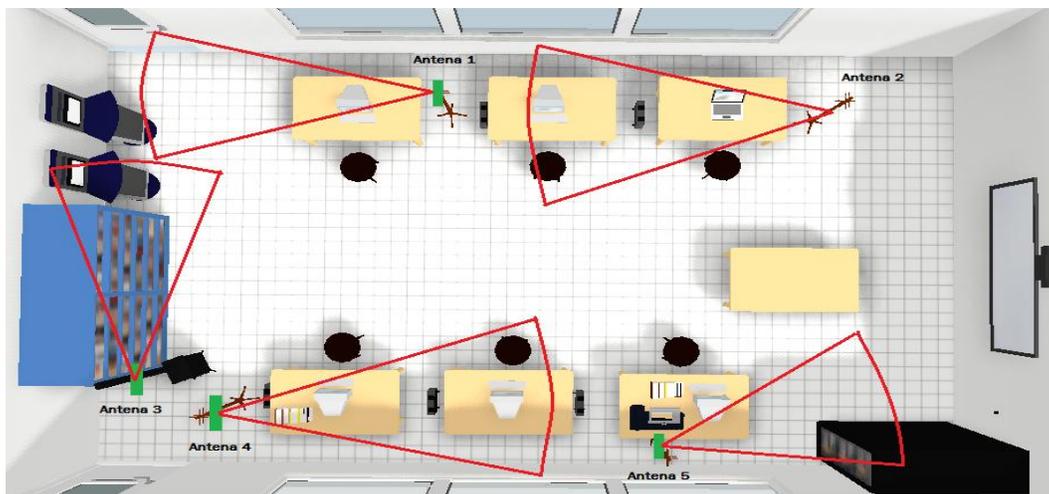


Figura 3.16

Como se puede apreciar en la imagen anterior, se necesitarían 5 módulos lectores para cubrir toda el área del salón de telemática; el inconveniente se da al querer controlar los artículos dentro de los estantes.

La figura 3.15 muestra otra forma en la que fueron colocados, los lectores.

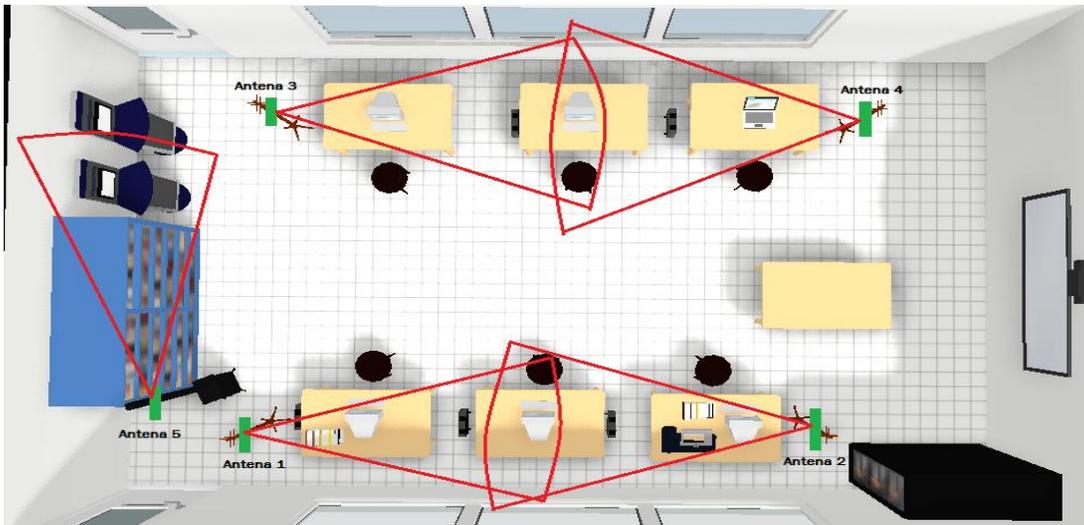


Figura 3 17

Con esta forma, también se tiene un total de 5 módulos lectores y aun así queda un estante sin ser controlado.

Las 2 opciones presentadas, con 5 módulos lectores, incrementarían el costo económico del proyecto; por lo que se fueron descartadas estas opciones, pudiéndose realizar el mismo funcionamiento, con la solución propuesta en el apartado 3.1.2.1.

Como resumen de la zona cubierta en el salón de telemática, se muestra la Figura 3.18

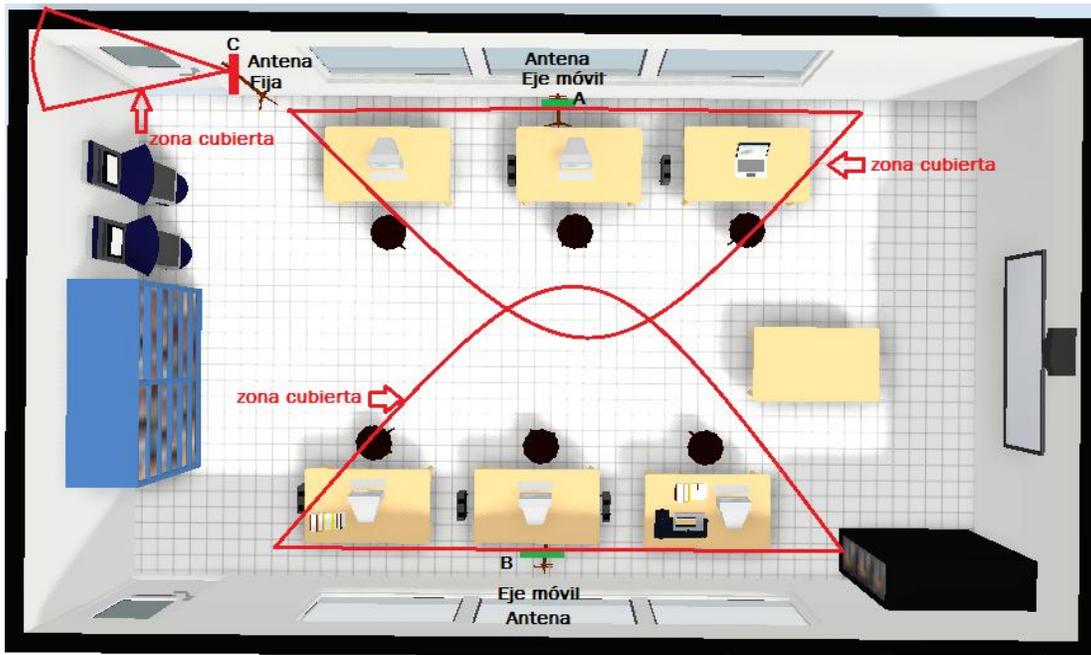


Figura 3 18

3.1.2.1.4 Almacén de bodega 1:

Justificación: Se decidió colocar un lector fijo en la posición D mostrada en la Figura 3.19, la antena radiaría en forma tal, que se cubra la zona de la puerta de acceso al salón.

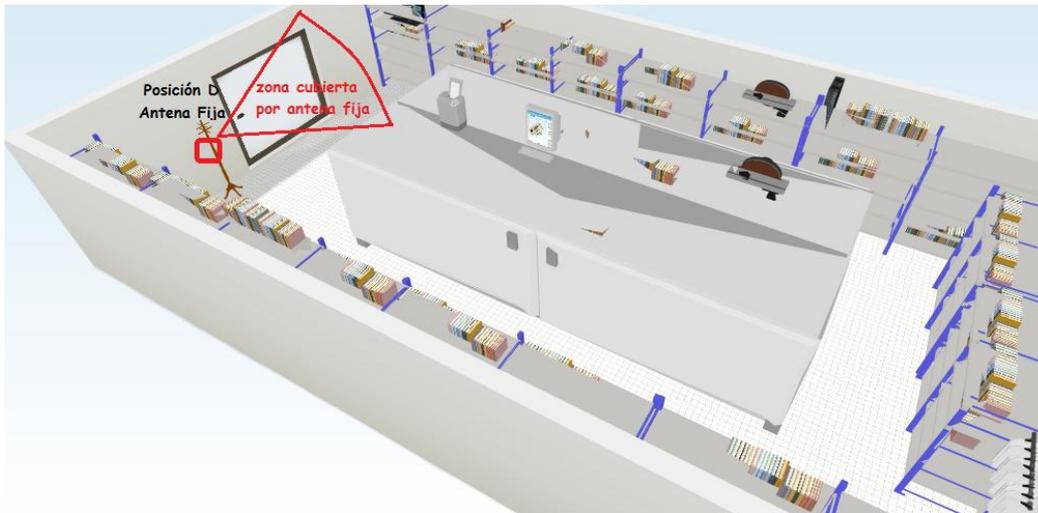


Figura 3 19

Se propone colocar aquí, debido a que con esta posición, se lograría controlar la entrada y salida de los elementos con tag's asignadas al salón y de esta forma resguardar, todos los artículos. Teniendo presente que es la única salida del local, fácilmente todo lo que allí se encuentra puede estar en la base de datos del sistema, lo que permitiría detectar a este, cuando algo no se encuentre en el local, mencionar que la puerta de acceso mide 2.10 metros de alto por 1 metro de ancho, la descripción de las pruebas, se detalla a continuación.

- **Descripción de las pruebas, para antena con eje fijo.**

- ✓ Posición D

Los pasos expuestos para una de las posiciones de la antena A (por ejemplo la A3) se repitieron para la posición D.

➤ Plano horizontal

Las lecturas de abertura horizontal de la posición D, se resumen en la Tabla 3.7.

Distancia en línea vista (cms)	Posición D (niveles obtenidos dB)					
	115		200		240	
Abertura Horizontal (cms)	Dcha.	Izq.	Dcha.	Izq.	Dcha.	Izq.
60	-36	-32	-31	-33	-32	-34
50	-32	-30	-38	-34	-31	-33

10	-32	-31	-37	-35	-32	-31
----	-----	-----	-----	-----	-----	-----

Tabla 3.7

➤ Plano Vertical

La Tabla 3.8 muestra las lecturas de abertura vertical.

Distancia línea vista (cms) Abertura vertical (cms)	Posición D (niveles obtenidos dB)					
	115		200		240	
	Arriba	Abajo	Arriba	Abajo	Arriba	Abajo
70	-32	-39	-37	-31	-34	-32
100	-33	-37	-38	-32	-37	-35
40	-32	-35	-34	-38	-39	-38
80	-33	-36	-33	-33	-36	-39
55	-36	-38	-32	-32	-33	-35

Tabla 3.8

- **Resultado de las pruebas, para antena con eje fijo**

Los resultados, serían prácticamente los mismos mostrados en las Figuras 3.14, 3.15 y 3.16 de este documento.

3.1.2.1.5 Proyecciones Futuras de otras instalaciones del Edificio de EIE, para control de Seguridad de equipos de laboratorio, con prototipo del sistema RFID.

Las Figuras 3.20 y 3.21 nos muestran otros salones del edificio de EIE, que mantienen equipo de valor; por lo que como proyecciones futuras, se proponen para brindarles, un control de seguridad, con el sistema RFID.

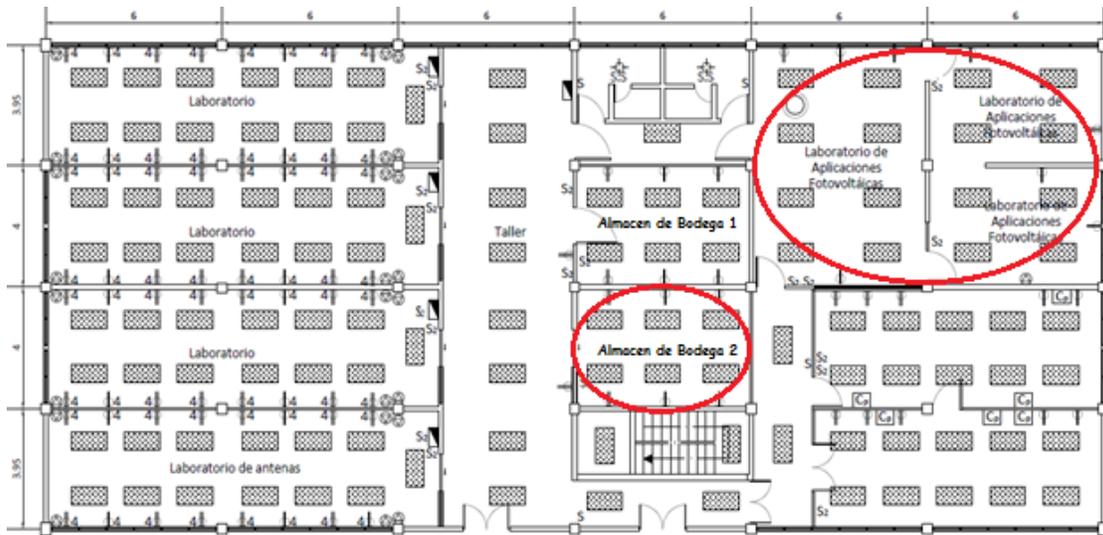


Figura 3.20: Primera planta de Edificio de EIE

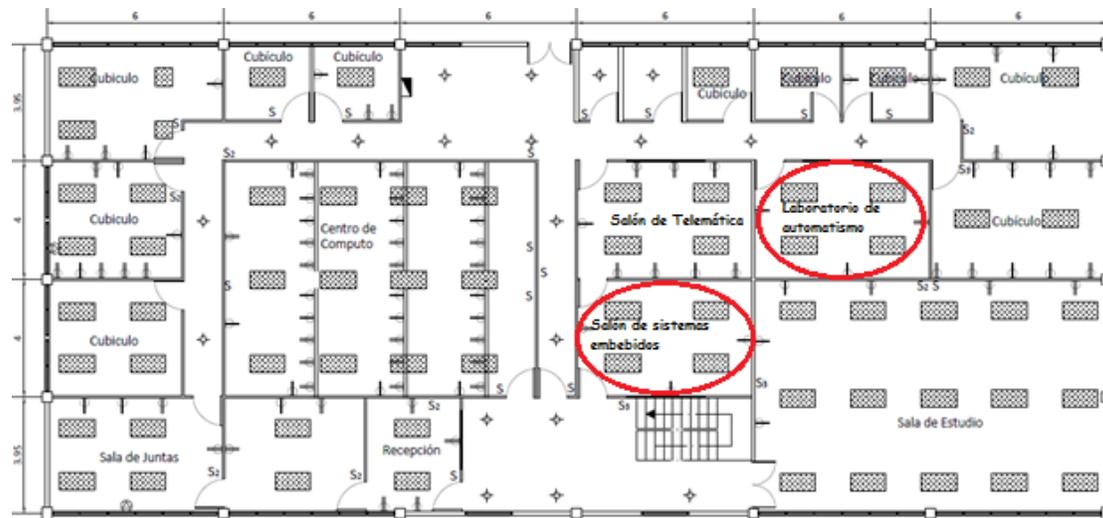


Figura 3.21: Segunda planta de Edificio de EIE

Como un breve análisis, la figura 3.22, nos muestra el salón de almacén de bodega 2, que es prácticamente idéntico en su contextura de local y en el equipo que es mantenido, dentro de él, al salón de bodega 1, ya analizado en el apartado 3.1.2.1.4 de este documento.

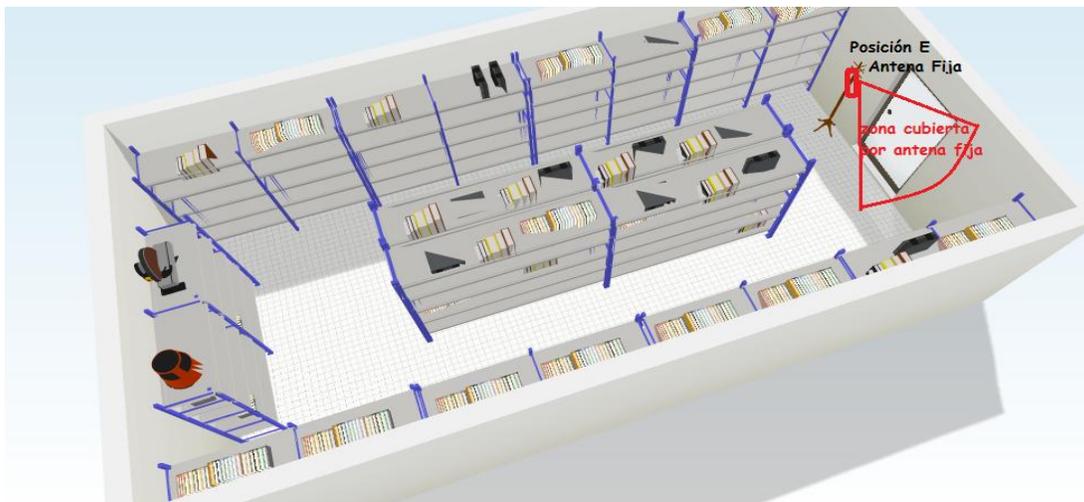


Figura 3.22: Salón de bodega 2

Por lo que con el módulo lector fijo en la posición indicada, se resguardaría, todo el equipo que hay dentro de este.

Muy parecido también resulta ser, el salón de sistemas embebidos, al salón de telemática ya analizado en el apartado 3.1.2.1

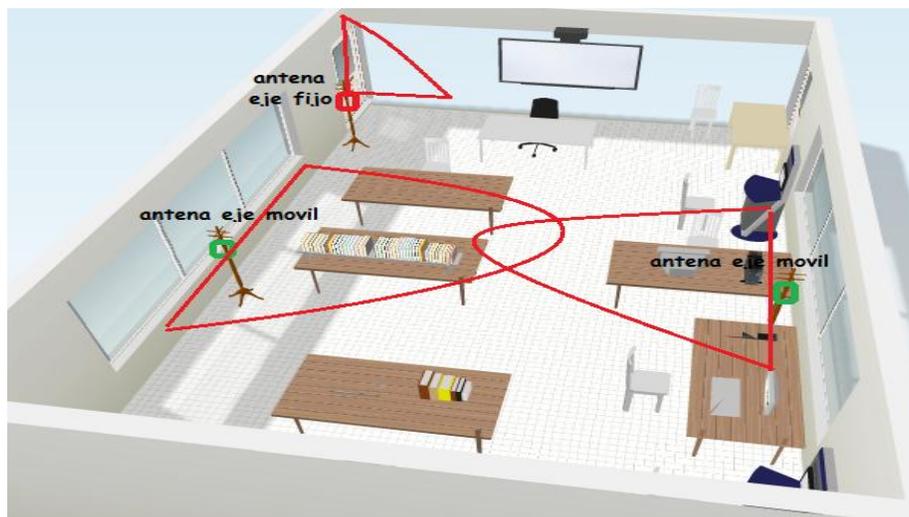


Figura 3.23: Salón de sistemas embebidos

Con la misma propuesta de 2 módulos móviles y 1 fijo, se podría mantener un control de seguridad de todo lo que hay dentro de este, lo mismo resulta, para el laboratorio de control automático, mostrado en la Figura 3.24.

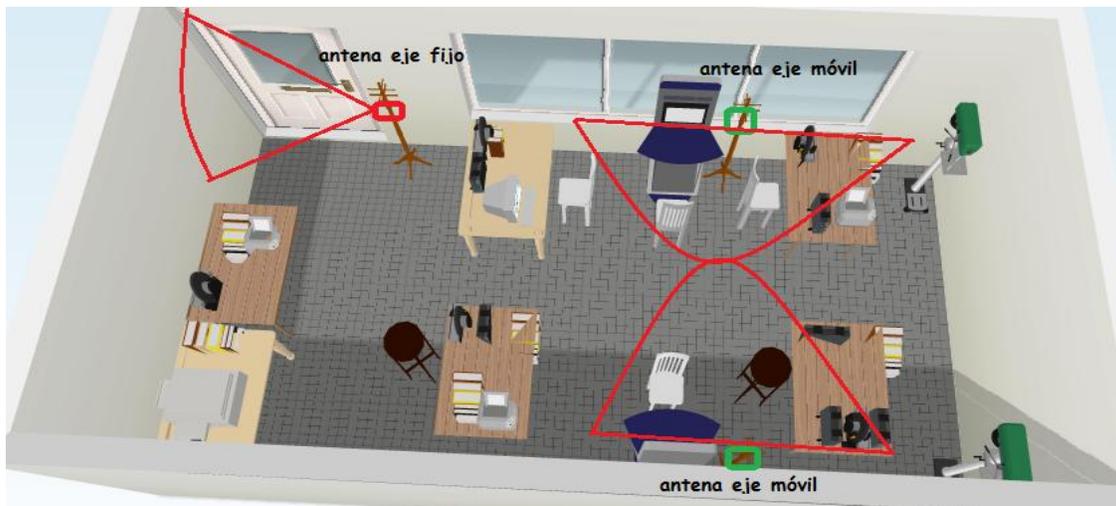


Figura 3.24: Laboratorio de Control Automático

Por último, un bosquejo del laboratorio de aplicaciones fotovoltaicas, es mostrado en la Figura 3.25, este local si es mucho más amplio de los analizados hasta este momento, no se guarda, tanto equipo como en los anteriores; pero tiene bastante acceso a ventanas que dan a la calle, por lo que una propuesta de seguridad para este salón, se elevaría en economía.



Figura 3.25: Laboratorio de aplicaciones fotovoltaicas

3.1.2.2 Inventario de equipos de laboratorio

Justificación: Para los equipos de los que se desea conocer su ubicación o cualquier otra característica de este, se propone un sistema manual con el módulo lector y antena, es decir colocar sobre una base, el sistema RFID en general y movilizarlo por cualquier persona con la mano, ya que sería la forma más económica de optimizar los recursos, porque con un solo sistema construido de esta forma, podrá servir, para cualquier aérea del edificio de EIE.

Limitantes: No se acoplo el sistema en esta forma, para hacer fácil su movimiento; por motivos económicos, sino que, lo que se hizo fue acercar módulo lector con la antena a artículos lo más cerca posible para su detección, una breve representación de esto es observada, en la Figura 3.24.

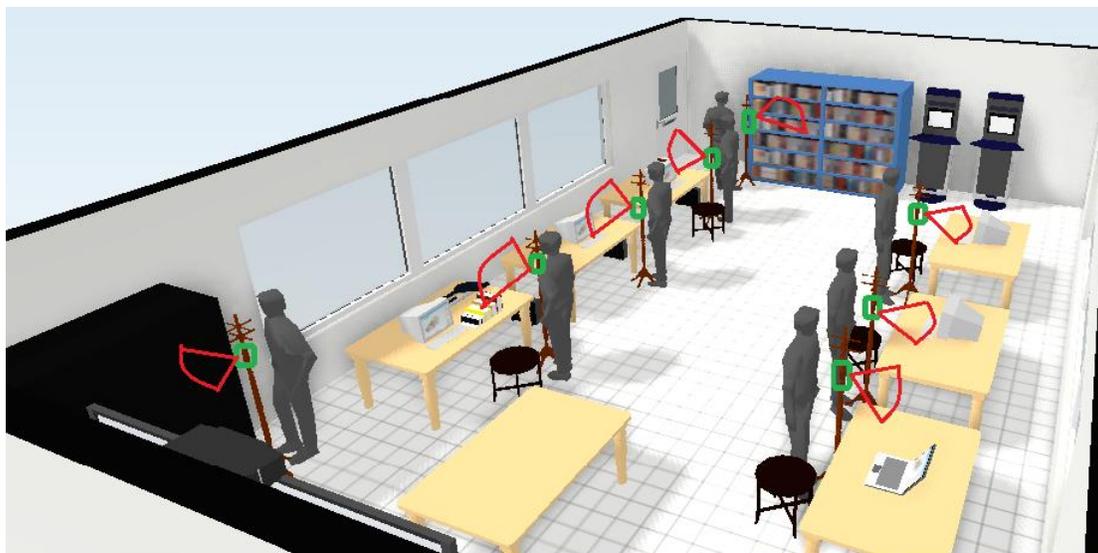


Figura 3.26

- **Descripción de las pruebas, del sistema RFID con uso manual.**

La tabla 3.9, muestra lecturas tomadas a artículos con tag's, a distancias cercanas (a distancias debajo de 50 cms, no se alcanzaba a observar la medida arrojada por el analizador de espectro; pero el sistema si la detectaba, como se verá a continuación en los resultados de pruebas).

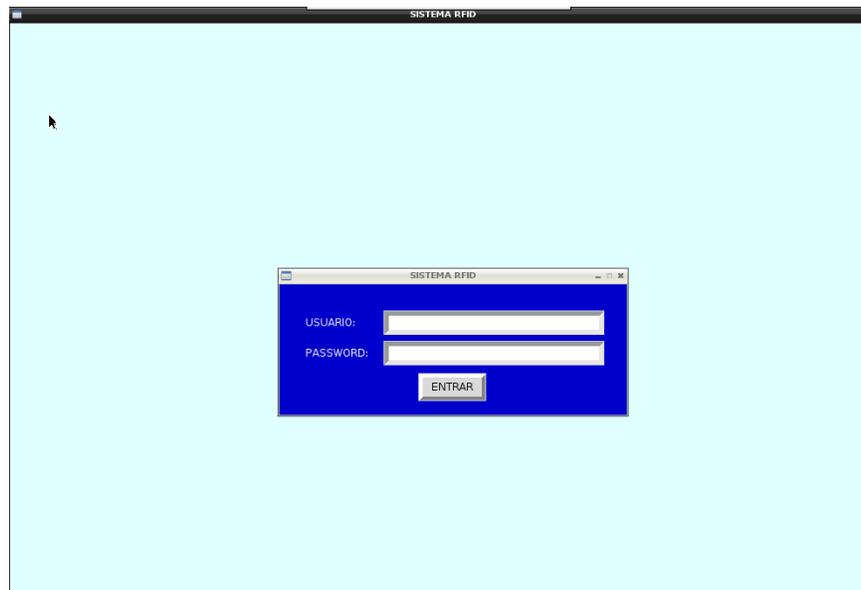
Distancias en línea vista (cms)	5	10	50
	Niveles obtenidos (dB)		
	---	---	-13

Tabla 3.9

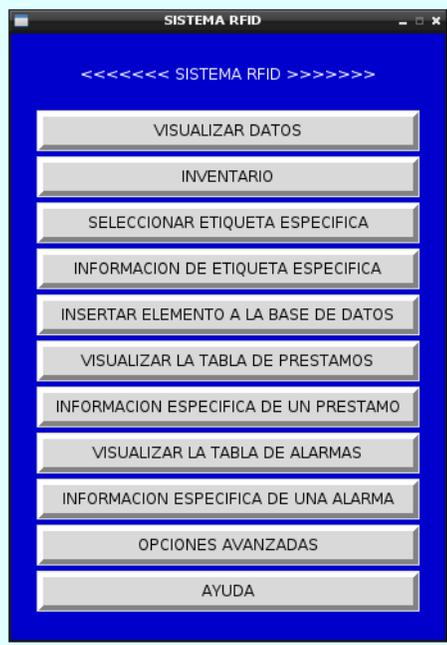
- **Resultado de las pruebas, del sistema RFID con uso manual.**

Pruebas de interfaz usuario-sistema.

El sistema prototipo posee dos tipos de interfaces para poder interactuar entre las funciones que puede realizar y el usuario, estas son muy importante ya que muestra de forma agradable las acciones que el sistema pueda ofrecer, el primer tipo de interface es la mostrada en la figura siguiente, esta se encuentra alojada en el procesador principal y requiere que el usuario este cerca del mismo



a)



b)



c)

Figura 3.27: Interface gráfica principal: a) Registro de usuario para el ingreso del mismo al sistema, b) Menú principal de opciones o funciones, c) Menú de opciones avanzadas, el cual muestra funciones para la modificación física del funcionamiento del mismo.

El segundo tipo de interface que posee el sistema es una tipo WEB, como se muestra en la siguiente figura, esta posee las mismas funciones que la interface principal o que se encuentra en el procesador principal, la ventaja que posee es que no es necesario que el usuario se encuentre cerca al sistema, ya que la comunicación entre usuario y sistema se realiza por medio de la web, otorgándole una gran libertad al mismo,



a)

SISTEMA RFID

[VISUALIZAR BASE DE DATOS](#)
[INVENTARIOS](#)
[SELECCIONAR ETIQUETA ESPECIFICA](#)
[INFORMACION DE ETIQUETA ESPECIFICA](#)
[INSERTAR ELEMENTOS A LA BASE DE DATOS](#)
[VISUALIZAR LA TABLA DE PRESTAMOS](#)
[INFORMACION ESPECIFICA DE UN PRESTAMO](#)
[VISUALIZAR LA TABLA DE ALARMAS](#)
[INFORMACION ESPECIFICA DE UNA ALARMA](#)
[OPCIONES AVANZADAS](#)

Un sistema RFID es un sistema de identificación por radiofrecuencia (por sus siglas en inglés), este sistema lleva un registro documental de los bienes pertenecientes a una persona o comunidad, esto quiere decir que el sistema es capaz de llevar un inventario de los elementos así como el control de los préstamos de los mismos y un sistema de control de alertas para los elementos ausentes que estén registrados, este sistema es conveniente a la hora de la realización de un inventario ya que lo efectúa en tiempo real, realizando un escaneo por radiofrecuencia, ahorrando así el esfuerzo del usuario a la hora de hacer un control de los elementos.

Este tipo de sistema se compone de varias partes, un sistema de mando el cual es un host para que el usuario pueda tener un total control sobre el sistema, una interfaz amigable al mismo, y información pertinente de los elementos en el inventario, un módulo RFID el cual es quien realiza el escaneo por radiofrecuencia con ayuda de una antena externa, para el presente caso el sistema opera en la banda de frecuencia UHF (la cual es de 300MHz a 900MHz); y la parte más importante es una etiqueta, la cual se le adhiere al elemento a controlar, esta etiqueta posee una antena y un chip con la información necesaria para que el módulo RFID la diferencie de las demás, esta información es llamada ID y es único para cada etiqueta.

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

b)



c)

Figura 3.28: Interface gráfica WEB: a) Registro de usuario para el ingreso del mismo al sistema, b) Menú principal de opciones o funciones, c) Menú de opciones avanzadas, el cual muestra funciones para la modificación física del funcionamiento del mismo.

Dichas interfaces fueron probadas con interacciones físicas con usuarios, y el sistema en funcionamiento, en las instalaciones de EIE de la UES, estos ensayos mostró aceptación por parte de las personas que manipularon el sistema, dándole una interface sistema-usuario amigable al mismo.

Inventario.

Uno de los procesos más importante en el funcionamiento del sistema es la función de inventario, dicha función muestra al usuario cuales elementos se encuentran en la zona de control y advierte o muestra una alarma si falta un elemento, el mismo posee dos formas de realizarlo, a petición del usuario el cual decide cuándo y cuantos se deben realizar, y de forma automática la cual el sistema decide cuantos realizar, este se detiene cuando no se detecta un elemento registrado, así mismo estos se dividen en tres tipos, inventario sencillo el cual el lector RFID genera un escaneo verificando si los elementos a controlar se encuentran en el lugar, y si existe algunos que no estén y que no se tenga permiso de su ausencia o préstamo, este generara una alarma tanto en la misma interface como un mensaje de texto a usuarios específicos; inventario con RSSI, esta realiza lo mismo que lo anterior pero con el agregado de que muestra la frecuencia a la cual se realizó el escaneo y su valor de RSSI del mismo; y el inventario continuo o automático el cual realiza el mismo proceso que el inventario sencillo solo que dicho proceso es recursivo esto quiere decir que realizar inventarios tras inventarios hasta que el sistema detecte que falta un elemento y proceda a ejecutar la función de alarma (la cual es mandara un mensaje de texto al teléfono celular de usuarios específicos).

A continuación se muestra un ejemplo del funcionamiento del sistema, con el proceso de inventario sencillo, mostrando primero un inventario donde todos los elementos se encuentran en el sitio, dicho inventario se realizó con las dos interfaces del sistema,

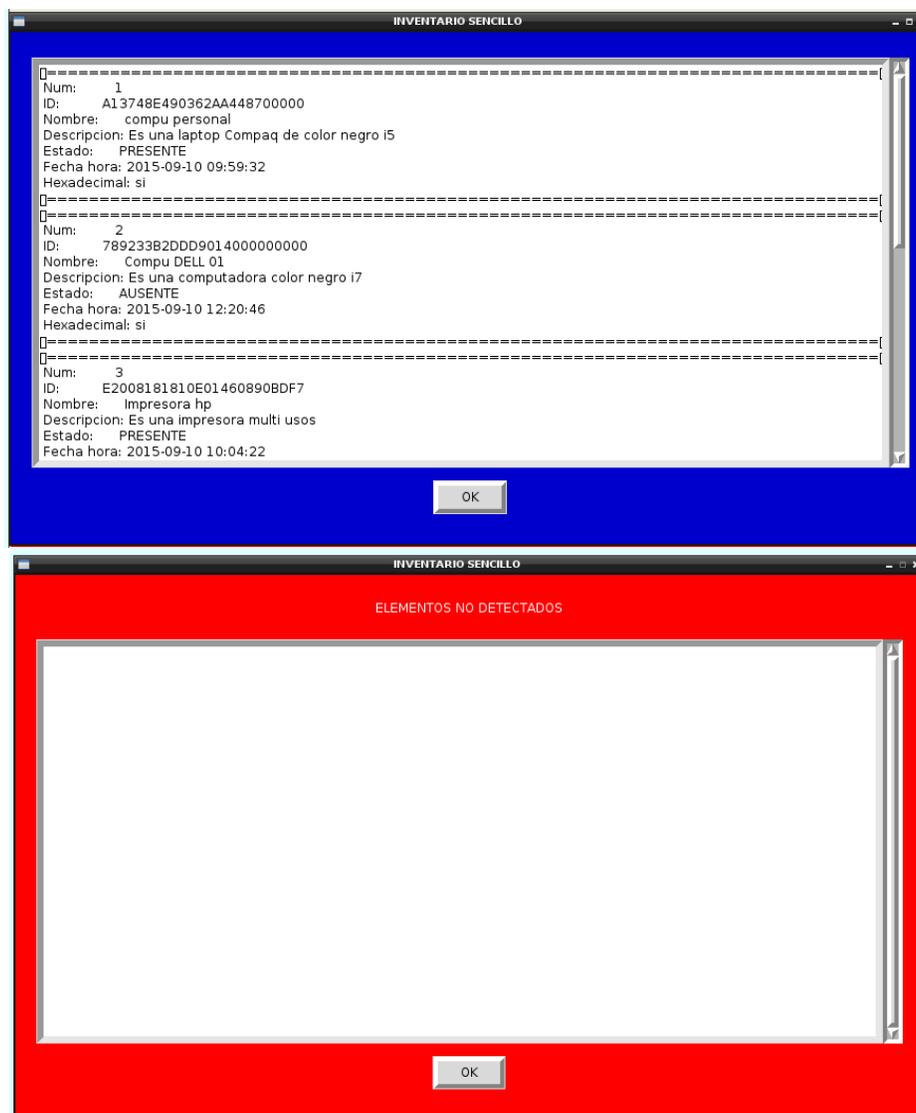


Figura 3.29: Ventana de interface principal de los resultados de la función de inventario sencillo, en este muestra que se encuentran todos los elementos a controlar en el sitio.

INVENTARIO

num	Nombre	ID	Descripcion	ESTADO	Fecha y Hora	HEX
1	compu personal	A13748E490362AA448700000	Es una laptop Compaq de color negro i5	PRESENTE	10 de Septiembre de 2015 a las 09:59	si
2	Compu DELL 01	789233B2DDD9014000000000	Es una computadora color negro i7	PRESENTE	10 de Septiembre de 2015 a las 10:02	si
3	Impresora hp	E2008181810E01460890BDF7	Es una impresora multi usos	PRESENTE	10 de Septiembre de 2015 a las 10:04	si
4	Compu DELL 02	E2008181810E01460990B1B5	Computadora i7 de 8 ram	PRESENTE	10 de Septiembre de 2015 a las 10:08	si
5	Router cisco 2600	987654321105000000000000	Se ocupa para realizar practicas de laboratorio	PRESENTE	10 de Septiembre de 2015 a las 10:10	si
6	Router cisco 2800	987586850000000000000000	Se ocupa para realizar practicas de laboratorio	PRESENTE	10 de Septiembre de 2015 a las 10:25	si

ELEMENTOS NO DETECTADOS

NO EXISTE ALARMAS!

Figura 3 30: Ventana de interface WEB de los resultados de la función de inventario sencillo, en este muestra que se encuentran todos los elementos a controlar en el sitio.

Las imágenes anteriores muestran los resultados de la realización de un inventario sencillo, y el resultado del mismo en la dos interfaces del sistema, mostrando que todos los elementos a controlar se encuentran en el lugar, pero en la siguiente figura se muestra la reacción y la ejecución de las alarmas cuando el sistema no detecta un elemento,

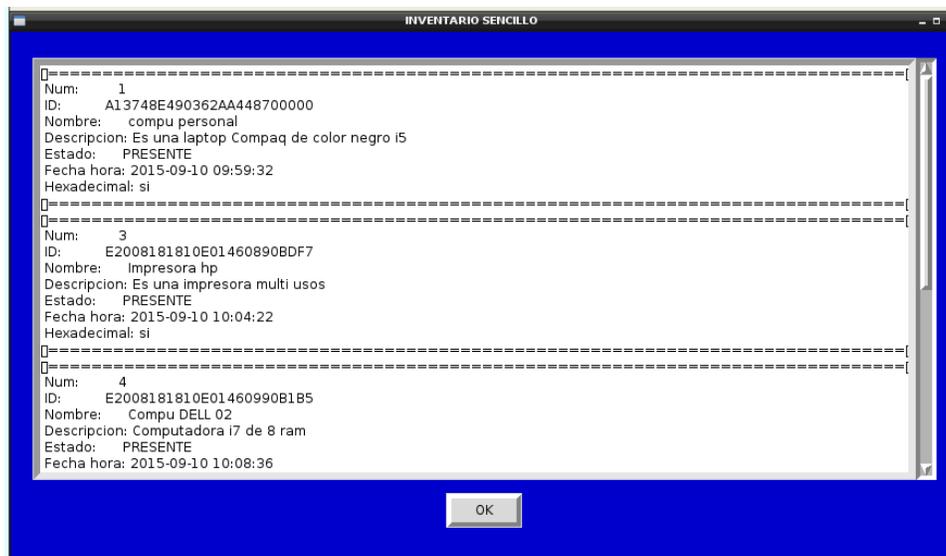




Figura 3.31: Ventana de interface principal de los resultados de la función de inventario sencillo, en esta muestra que en el lugar falta un elemento.

INVENTARIO						
num	Nombre	ID	Descripcion	ESTADO	Fecha y Hora	HEX
1	compu personal	A13748E490362AA448700000	Es una laptop Compaq de color negro i5	PRESENTE	10 de Septiembre de 2015 a las 09:59	si
3	Impresora hp	E2008181810E01460890BDF7	Es una impresora multi usos	PRESENTE	10 de Septiembre de 2015 a las 10:04	si
4	Compu DELL 02	E2008181810E01460990B1B5	Computadora i7 de 8 ram	AUSENTE	10 de Septiembre de 2015 a las 10:08	si
5	Router cisco 2600	98765432110500000000000000	Se ocupa para realizar practicas de laboratorio	PRESENTE	10 de Septiembre de 2015 a las 10:10	si
6	Router cisco 2800	98758685000000000000000000	Se ocupa para realizar practicas de laboratorio	PRESENTE	10 de Septiembre de 2015 a las 10:25	si

ELEMENTOS NO DETECTADOS

num	Nombre	ID	Descripcion	ESTADO	Fecha y Hora	HEX
2	Compu DELL 01	789233B2DDD901400000000000	Es una computadora color negro i7	AUSENTE	10 de Septiembre de 2015 a las 12:23	si

Figura 3.32: Ventana de interface WEB de los resultados de la función de inventario sencillo, en esta muestra que en el lugar falta un elemento.

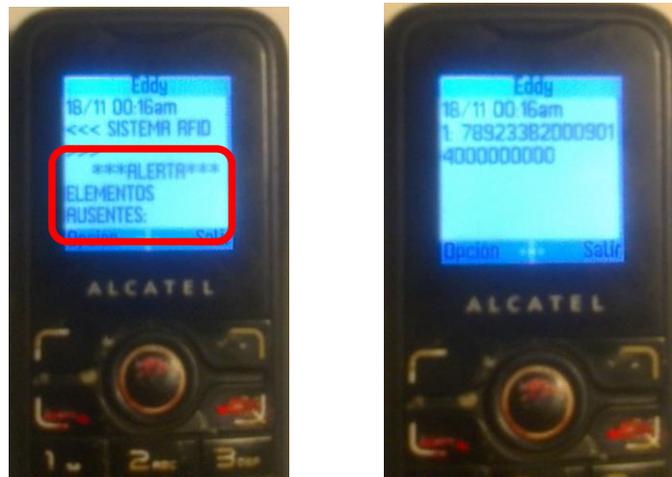


Figura 3.33: Mensaje de texto de alarmas del sistema prototipo RFID dirigidos a un usuario específico, al realizar un inventario.

Las imágenes anteriores muestran la reacción del sistema cuando se detecta que falta un elemento en el escaneo o inventario efectuando alarmas tanto en el mismo sistema y como en mensajes de texto dirigidos a teléfonos celulares de usuarios específicos, otorgándole una ventaja sobre los existentes en el mercado.

3.1.3 Presupuesto.

El costo de la elaboración del sistema prototipo de seguridad RFID, se presenta en la siguiente tabla, se detalla todos los componentes del sistema así como también su valor monetario de cada uno, mostrando la inversión total para la realización del mismo.

NOMBRE	COSTO
Modulo lector UHF RFID AS3992, fuente de poder de 3.3V – 4.55A, antena de 5dbi	\$219.43
U-EC6 C8051F Emulator Programmer w/10pin JTAG Interface Adapter Plate Turn C2	\$25.00
Arduino MEGA 2560	\$25.00
Sim900 cuatro bandas GSM/GPRS Shield Para Arduino uno/mega/leonardo	\$30.00
Arduino UNO	\$10.00
RASPBERRY PI 2 - Model B. 1GB RAM, Quad Core CPU (Worldwide Free Shipping)	\$40.42
Uhf915m-antpccr 8dbi UHF RFID Antena 915mhz Pasiva Rfid tabular Antena Mmcx	\$61.88
10 Piezas Rp-sma Mujer Pin Macho final lanzamiento de Jack Montaje Pcb versión larga Conector De Rf	\$10.24
Mmcx enchufe ra A Rp-sma Plug Mujer Pin ángulo recto Pigtail Cable Rg316 50cm	\$5.99
25 Etiquetas RFID UHF pasivas	\$24.75
TOTAL (Sin cobro de envío)	\$452.71

Tabla 3.10

CONCLUSIONES

- ❖ Las aplicaciones RFID facilitan muchas opciones para controlar el movimiento de dispositivos y elementos varios, y con los complementos adecuados, pueden brindar opciones de seguridad importantes para empresas o instituciones que administran gran cantidad de inventarios
- ❖ Las prestaciones de este tipo de sistemas, pueden mejorarse mediante el fortalecimiento de los sistemas de antenas asociados a ellos
- ❖ Una evolución propia de estas aplicaciones, son las tecnologías tipo Smart Card, Chip Card, o Integrated Circuit Card, las cuales potencian aún más sus ventajas, y aunque no se han discutido en el presente documento, se recomienda su estudio en trabajos posteriores.
- ❖ Otra evolución que se recomienda estudiar, son las que utilizan las frecuencias Microondas (2.45 GHz, 5.8 GHz), las cuales evitan las interferencias derivadas desde otras aplicaciones GSM (telefonía móvil por ejemplo), y la versatilidad de la creación de antenas a esa frecuencia
- ❖ Los procesos manuales de toma de inventarios, en especial cuando se disponen de gran cantidad de elementos, toman mucho tiempo y con soluciones como la presente, se logra disminuir significativamente el tiempo en cuestión, con el consiguiente ahorro para las empresas y/o instituciones.

BIBLIOGRAFÍA:

- [1]. Nombre: “La guía definitiva de django: desarrolla aplicaciones Web de forma rápida y sencilla”
Autor: Saul García M.
Versión: 1.8.0
Año de edición: 2015
- [2]. Nombre: TTL UART/USB Long Range UHF RFID reader Module (ISO18000-6C EPC G2) UART Protocol
Autor: LinkSprite Technologies, Inc.
Año de edición: 2011
- [3]. Nombre: UHF RFID Reader Module User Manual
Autor: LinkSprite Technologies, Inc.
Año de edición: 2012
- [4]. Nombre: ARDUINO Curso práctico de formación.
Autor: Oscar Torrente Artero
Editorial: Alfaomega
Año de edición: 2013
- [5]. Nombre: Arduino Cookbook
Autor: Michael Margolis
Editorial: O`reilly
Año de edición: 2011
- [6]. Nombre: An Introduction to Tkinter
Autor: David Beazley
Editorial: Copyrigh©
Año de edición: 2008
- [7]. Nombre: Guía de Usuario de Arduino
Autor: Rafael Enrique Herredor
Año de edición: 2009
- [8]. Nombre: Arduino: Manual de programación
Autor: José Manuel Ruiz Gutiérrez
Año de edición: 2007
- [9]. Nombre: Tutorial de PHP y MySQL Completo
Autor: José Antonio Rodríguez
Año de edición: 2000
- [10]. Nombre: Manual básico de creación de página Web
Autor: Universidad de Murcia
- [11]. Nombre: Manual básico de HTML: Creación y estructura de páginas Web
Autor: Carlos Eduardo Anibarro Zelaya

Año de edición: 2001

- [12]. Nombre: Curso: Python para principiantes
Autor: python, tm
Editorial: Eugenia Bahit
Año de edición: 2012

- [13]. Nombre: Lenguaje HTML
Editorial: Grupo Eidos
Edición: 1.0.0
Año de edición: 2000

- [14]. Artículo: Controla tu placa arduino con raspberry pi y python
Autor: Andrea Stagi
Año de edición: 2012

- [15]. Nombre: Python para todos
Autor: Raúl González Duque
Editorial: Creative Commons Reconocimiento 2.5 España

- [16]. Nombre: MySql Tutorial
Editorial: tutorialspoint.com

- [17]. Nombre: Python Tutorial
Editorial: tutorialspoint.com

- [18]. Nombre: Comandos básicos de MYSQL
Autor: Miguel Ángel Sanz Santos
Editorial: Fac. CC. Geológicas – UCM
Año de edición: 2002-2003

- [19]. Nombre: GUI Programming using Tkinter
Autor: Cuauhtemoc Carbajal
Editorial: ITESM CEM
Año de edición: 2003

- [20]. Nombre: Tkinter 8.5 reference: a GUI for python
Autor: John W. Shipman
Editorial: NEW MEXICO TECH
Año de edición: 2003

- [21]. Nombre: Antennas
Autor: J. D. Kraus
Editorial: New Delhi: Tata McGraw-Hill
Año de edición: 1988.

- [22]. Nombre: Microstrip Antenna Design Handbook
Autor: P. B. Ramesh Garg
Editorial: London: Artech House
Año de edición: 2001.

- [23]. Nombre: Antenna Theory Analysis and Design Third Edition
 Autor: C. A. Balanis
 Editorial: New Jersey: Wiley Interscience
 Año de edición: 2005.
- [24]. Nombre: RFID Handbook
 Autor: K. Finkenzeller
 Editor: Wiley
 Año de edición: 2010
- [25]. Nombre: *Identificación por Radiofrecuencia: Fundamentos y Aplicaciones*
 Autor: J. R. C. O. G. J. M. C. J. B. Dante I. Tapia
 Editor: Salamanca, España
 Año de edición: 2007
- [26]. Nombre: *RFID Tecnología de Identificación por Radiofrecuencia y sus principales aplicaciones*
 Autor: Cosejería de fomento
 Año de edición: 2007
- [27]. Nombre: *Sistema de Control de Acceso y Monitoreo con la Tecnología RFID para el Departamento de Sistemas de la Universidad Politécnica Salesiana*
 Autor: Z. V. V. Vergara
 Editor: Guayaquil
 Año de edición: 2013
- [28]. Nombre: *Uso y Aplicaciones de las Etiquetas RFID*
 Autor: D. C. Martin
 Editorial: Pamplona
 Año de edición: 2010
- [29]. Nombre: Diseño y construcción del subsistema RFID para un expositor inteligente.
 Autor: José Manuel Jiménez Payá
 Año de edición: 2008
- [30]. Nombre: Diseño de antenas UHF para aplicaciones RFID
 Autor: José Vicente Hernández Ripoll
 Año de edición: 2009
- [31]. Nombre: Sistema de reparto de recursos en un sistema FID con lectores interferentes y tags móviles.
 Autor: Álvaro Campillo Soler
 Año de edición: 2013
- [32]. Nombre: Diseño de antenas y caracterización de Interferencias entre lectores RFID
 Autor: José Daniel Guzmán Guzmán
 Año de edición: 2009
- [33]. Nombre: Análisis de modelos de propagación e interferencia de la tecnología RFID pasiva de UHF para aplicación en la identificación vehicular.

Autor: Ing. Roberto Orozco Vega
Año de edición: 2011

- [34]. Nombre: Análisis de antenas de UHF para aplicaciones de RFID
Autores: José Antonio Hernández González y José Mauricio Montalbán Pérez
Año de edición: 2010
- [35]. Nombre: Estudio, diseño y simulación de un sistema RFID basado en EPC
Autor: José María
- [36]. Título de Tesis: Sistema RFID, su evolución, aplicaciones actuales e interoperatividad.
Autores: Gilberto Azcúnaga Vargas y Carlos Armando Cornejo García
Año de edición: 2008
- [37]. Nombre: RFID: Tecnología, aplicaciones y perspectivas
Autor: whitepaper series
Año de edición: 2010
- [38]. Nombre: *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*
Autor: L. M. R. S. Daniel Alejandro Cadena Moran
Editor: Sangolqui-Ecuador
Año de edición: 2011
- [39]. Título: How to program Cottonwood the RFID reader with USB and UART firmware
Dirección: <http://learn.linksprite.com/rfid/how-to-program-the-rfid-reader/>
- [40]. Título: LinkSprite Long Distance RFID reader (ISO18000-6C EPC G2) Communication Protocol
Dirección: <http://learn.linksprite.com/rfid/linksprite-long-distance-rfid-reader-iso18000-6c-epc-g2-communication-protocol/>
- [41]. Título: Tutorial of Arduino Cottonwood UHF RFID Reader.
Dirección: <http://learn.linksprite.com/rfid/tutorial-of-arduino-cottonwood-uhf-rfid-reader/>
- [42]. Título: Link Sprite RFID
Dirección: <http://learn.linksprite.com/category/rfid/>
- [43]. Título: L. M. Blazquez, «Sistemas de identificación por Radiofrecuencia,».
Dirección: <http://www.it.uc3m.es>
- [44]. Título: Libera, «RFID: Tecnología, Aplicaciones y Perspectivas,» 2010.
Dirección: <http://www.libera.net>

ANEXO A: MANUAL DE USUARIO.

El sistema se compone de dos partes primordiales, el hardware que consiste en las diferentes partes físicas o módulos que componen el módulo de control; y el software el cual posee la forma de ejecutar los comandos u órdenes para el funcionamiento de cada módulo así mismo la interface de con el usuario para el control del funcionamiento de todos los procesos del sistema.

A.1 MANUAL DE USUARIO: HARDWARE

Los diferentes bloques o módulos que componen el sistema físicamente, se conectan de forma específica entre sí para tener una comunicación bilateral eficiente, así mismo algunos módulos necesitan ser activados físicamente para su funcionamiento, a continuación se describirán puntos importantes sobre el funcionamiento u acoplamiento de los diferentes elementos físicos que compone el sistema en su totalidad.

✚ CONEXIÓN ENTRE LA INTERFACE RFID Y EL PROCESADOR PRINCIPAL DEL SISTEMA.

La interface RFID corresponde a el modulo lector AS3992 y el arduino MEGA 2560, así mismo el procesador principal es una raspberry pi B+, en la siguiente figura se muestra la forma de conexión entre el módulo AS3992 y el arduino MEGA 2560.

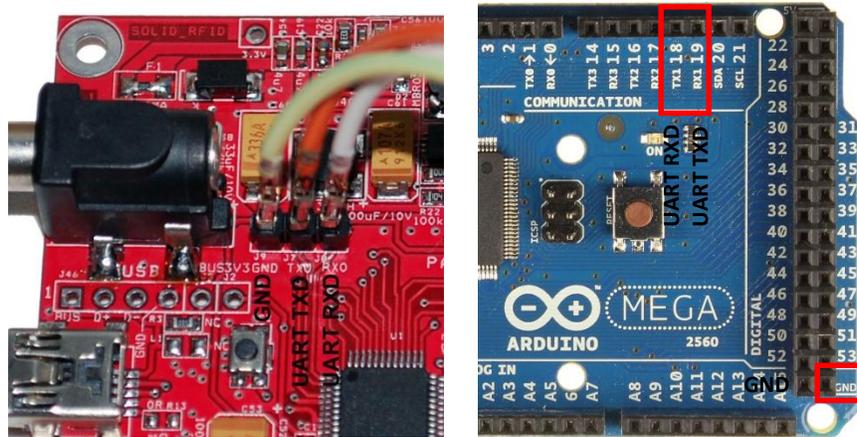


Figura A.1: Conexión entre el módulo lector RFID AS3992 y el arduino MEGA 2560.

En la figura anterior se muestra la forma en que se deben de conectar el modulo lector RFID AS3992 y el arduino MEGA 2560 (se debe tener presente que el módulo AS3992 que la forma en que se conecta dependerá del tipo de FIRMWARE⁴⁶ que fue instalado en él, así mismo la forma en que se le da energía) el FIRMWARE que posee el lector AS3992 para el funcionamiento del sistema es el controlador UART, gracias a esto es necesaria una fuente de 3.3V y 4A, y así conectar los puertos UART con los pines del arduino MEGA 2560, el puerto UART TXD con el pin 18 (RX1) y el puerto UART RXD con el pin 19 (TX1), así mismo el puerto GND del módulo AS3992 se conecta con uno de los tantos pines GND del arduino MEGA.

⁴⁶ <http://learn.linksprite.com/rfid/how-to-program-the-rfid-reader/>

Para conocer si el módulo RFID AS3992 se encuentra encendido o en funcionamiento, como se muestra en la siguiente figura, el led de color rojo de power debe de encender en forma intermitente, si no es así el lector RFID se encuentra apagado.

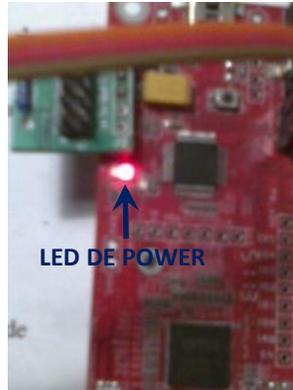


Figura A.2: Led de encendido o power del módulo RFID AS3992.

Así mismo el arduino MEGA 2560 es alimentado y se comunica con la Raspberry pi B+ por medio de su puerto USB, por ende las interfaces graficas del sistema, las cuales se encuentran en el procesador principal, se comunica con el arduino MEGA 2560 por medio del mismo puerto.

CONEXIÓN ENTRE LA INTERFACE GSM Y EL PROCESADOR PRINCIPAL DEL SISTEMA.

La interface GSM corresponde al módulo arduino UNO y al arduino SIM 900, el arduino shield SIM 900 se conecta encima del arduino UNO, encajando cada uno de sus pines, como se muestra en la siguiente figura.



Figura A.3: Conexión entre el arduino UNO y el arduino shield SIM 900 (GPRS Shield V1.1(B)).



Figura A.4: Puentes en el shield SIM 900 para elegir los pines de comunicación TX y RX.

Los puentes están en posiciones de GRX=3 y GTX=2 como se ve en la figura A.4 y utilizar la librería GSM.h (en el arduino UNO), PRECAUCIÓN en la placa las conexiones de los pines serie están cruzados! GTX=RX 2 Arduino y GRX=TX 3 Arduino. El shield GSM necesita una alimentación de una fuente externa de 5V y 2A, esto es necesario ya que lo necesita para alimentar la antena GSM del shield y se comunice con el sistema GSM comercial, esta alimentación se conecta en los pines VIN y GND del shield GSM SIM 900, así mismo es necesario una tarjeta SIM celular, no importa de qué compañía sea ya que el módulo GSM posee las 4 banda celulares abiertas.

LEDs (color)	Estatus	Desripcion
PWR (red)	ON	Encendido del Shield GPRS/GSM
	OFF	Apagado del Shield GPRS/GSM
NET (verde)	64ms On /800ms Off	SIM900 no ha registrado una red de telefonía movil
	64ms On /3000ms Off	SIM900 ha registrado en una red de telefonía movil
	64ms On /300ms Off	Se establecio comunicación GPRS
	OFF	SIM900 no se está funcionando o no esta activado

Tabla A. 1: Tabla de funcionamiento del GPRS Shield V1.1 (B)⁴⁷ arduino, SIM900.

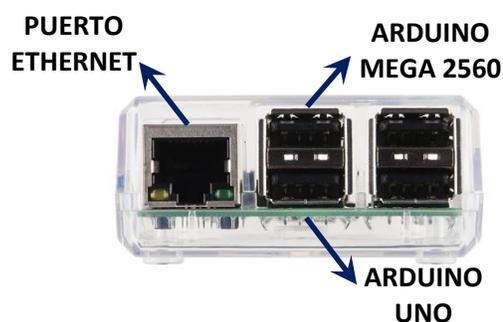


Figura A.5: Vista de los puertos USB y Ethernet de Raspberry PI B+ y posición de conexión de los módulos arduino MEGA 2560 y arduino UNO.

A.2 MANUAL DE USUARIO: SOFTWARE

El sistema de seguridad utilizando tecnología RFID, utiliza dos sistemas de interacción con el usuario, los cuales son:

El primero es una interface de usuario dentro del sistema de control, dándole así un control total al usuario sobre todo el sistema, esta interface fue creada en el lenguaje de programación PYTHON utilizando una librería llamada **TKINTER** (es un binding de la biblioteca grafica Tlc/Tk para el lenguaje de programación PYTHON, siendo así un estándar en la programación del lenguaje PYTHON).

El segundo es una interface web, para que el usuario pueda tener un control a distancia del módulo, esto conlleva a un sistema más eficiente y adecuado a su funcionamiento ya que el usuario no debe

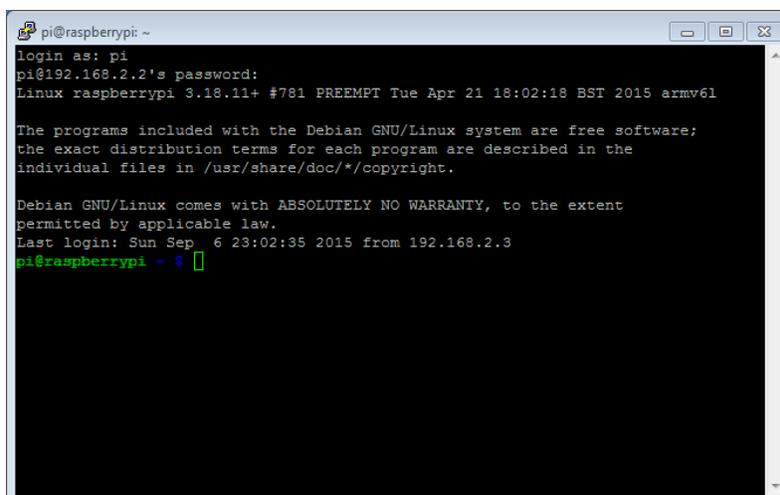
⁴⁷ <https://andrologiciels.wordpress.com/arduino/gsm/sim-900-ebay/>

de estar presente para su control, esta interface fue creada en el lenguaje de programación **DJANGO**, es un framework de desarrollo web de código abierto, escrito en el lenguaje de programación **PYTHON**, y con la ayuda o conexión con un servidor web **HTTP APACHE**, el cual es un servidor web **HTTP** de código abierto y el soporte del lenguaje de programación **HTML**, le otorgan al usuario una interacción con el sistema más eficiente e ordenada, para el procesamiento de las funciones del mismo.

El procesamiento de los datos en el sistema, está basado en el sistema de gestión de base de datos relacional, multihilo y multiusuario **MySQL**, el cual está basado en el lenguaje de programación **SQL**; le otorga al sistema un control de los datos internos del sistema, la interacción entre el sistema de gestión de base de datos **MySQL** y el sistema es controlado por el lenguaje de programación **PYTHON**, es el cerebro del sistema ya que gracias a este lenguaje de programación es controlado cada parte, proceso y función de todo el sistema, así como también la interacción del usuario con el mismo sistema, esto crea un ambiente eficiente y ordenado para la interacción con el usuario.

Para una mejor comprensión sobre el sistema se describirá el funcionamiento del mismo, se debe tener en cuenta que ya que las dos interfaces graficas controlan las mismas funciones y procesos en el sistema, por ende las dos interfaces con el usuario se describirán al mismo tiempo:

Para la conexión entre el usuario y la interface gráfica alojada en el sistema, se necesita una conexión ya sea ethernet o wifi, para la conexión entre el sistema de control que está alojado en una Raspberry B+ y un host para la interacción con el usuario, se ha utilizado una conexión con un servidor **SSH**, dicha conexión se realiza por medio del cliente **SSH** llamado **PUTTY**, este realiza una conexión iterativa con la raspberry B+ como se observa a continuación:



```
pi@raspberrypi: ~
login as: pi
pi@192.168.2.2's password:
Linux raspberrypi 3.18.11+ #781 FREEMPT Tue Apr 21 18:02:18 BST 2015 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

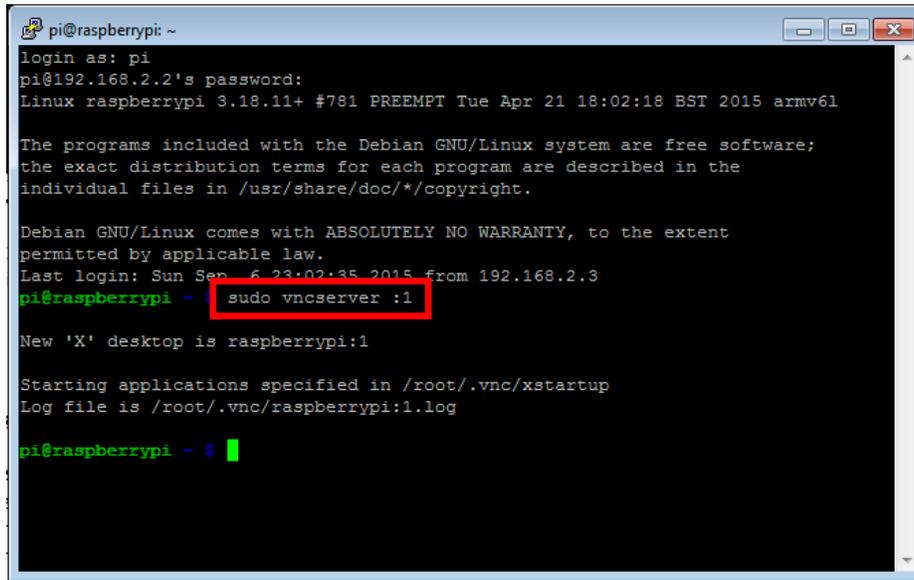
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Sep  6 23:02:35 2015 from 192.168.2.3
pi@raspberrypi - $
```

Figura A.6: Conexión con la Raspberry B+ por medio de un cliente SSH llamado PUTTY.

En la figura A.6 se observa la conexión realizada por el cliente **SSH** **PUTTY**, para esta conexión el usuario necesita conocer la IP que contiene la Raspberry, e ingresarla al sistema de **PUTTY**, para la comunicación con la Raspberry B+ y el Host el usuario debe ingresar los datos siguientes, para una comunicación bilateral:

Login: **pi**
Password: **raspberry**

Luego se necesita activar un servidor VNC, VNC (Virtual Network Computing por sus siglas en inglés, o escritorio remoto), es un programa basado en una estructura cliente-servidor el cual permite tomar el control del host servidor remotamente a través de un host cliente, es posible compartir la pantalla de una maquina con cualquier sistema operativo del host servidor con respecto al del cliente, tener en cuenta que el servidor VNC se encuentra en la Raspberry y el cliente VNC se encuentra en el Host del usuario.

A terminal window titled 'pi@raspberrypi: ~' showing the process of starting the VNC server. The terminal output includes login information for user 'pi' on a Raspberry Pi running Linux raspberrypi 3.18.11+. It displays the Debian GNU/Linux license notice and the last login time. The command 'sudo vncserver :1' is entered and highlighted with a red box. The output shows that a new 'X' desktop is created for user 'pi' on host 'raspberrypi:1', and applications are started from the file '/root/.vnc/xstartup'. The log file is '/root/.vnc/raspberrypi:1.log'. The terminal ends with the prompt 'pi@raspberrypi - \$' and a green cursor.

```
pi@raspberrypi: ~
login as: pi
pi@192.168.2.2's password:
Linux raspberrypi 3.18.11+ #781 PREEMPT Tue Apr 21 18:02:18 BST 2015 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Sep  6 23:02:35 2015 from 192.168.2.3
pi@raspberrypi ~$ sudo vncserver :1
New 'X' desktop is raspberrypi:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/raspberrypi:1.log

pi@raspberrypi ~$ █
```

Figura A.7: Activación del servidor VNC en la Raspberry B+.

Al activar el servidor VNC se necesita también activar el cliente VNC que realizara la conexión, el cual se realizara en el Host del usuario, y así realizar la comunicación con el sistema de control del sistema de seguridad RFID, e igual que la conexión con PUTTY el cliente VNC requiere la IP asignada a la Raspberry.

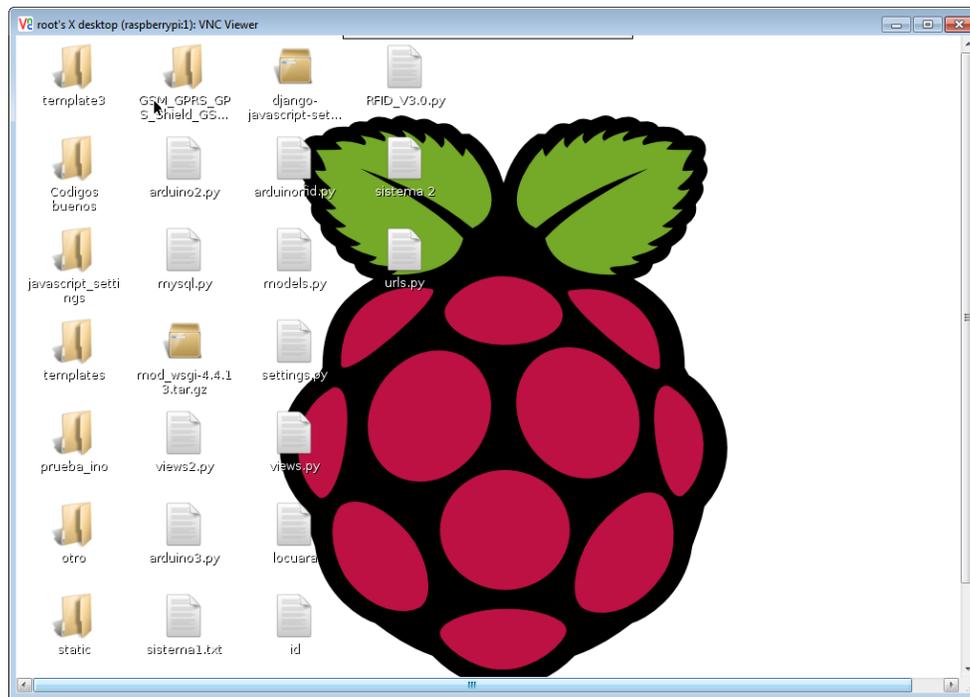


Figura A.8: Conexión con el escritorio remoto o VNC, con la Raspberry B+.

Al realizar dicha conexión el usuario debe iniciar la interface del sistema de seguridad RFID. Luego para la activación de la interface web, el usuario debe de ingresar a la carpeta donde se aloja el código del sistema, esta carpeta se llama **rfidsites**, para el ingreso de dicha carpeta el usuario debe introducir el comando **cd** y el nombre de la carpeta, este comando debe de ingresarse en PUTTY como se observa en la figura A.9, luego el usuario debe de ingresar el comando **python manage.py runserver 192.168.2.2:8080**, para iniciar el servidor DJANGO, dentro de la carpeta rfidsites, siendo así 192.168.2.2 la IP de la Rspberry y :8080 es el puerto que el usuario pretenda tener comunicación entre la Raspberry y el Host del usuario, se recomienda usar el puerto 8080 ya que este es el puerto http, si el usuario omite la IP y el puerto de comunicación, el servidor DJANGO correrá con la IP 127.0.0.1 en el puerto 8000, para que el sistema y el usuario no tengan problemas de comunicación se recomienda que al iniciar el servidor DJANGO ingrese la IP con la que se quiere comunicar (de preferencia la IP de la Raspberry) y el puerto al cual se quieran comunicar, y por último el usuario ingresara la IP y el puerto de comunicación que se le otorgo al servidor DJANGO, en el buscador WEB, como se observa en la figura A.10.

```
pi@raspberrypi: ~/rfidsite
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Sep  6 23:02:35 2015 from 192.168.2.3
pi@raspberrypi ~$ sudo vncserver :1

New 'X' desktop is raspberrypi:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/raspberrypi:1.log

pi@raspberrypi ~$
pi@raspberrypi ~$
pi@raspberrypi ~$
pi@raspberrypi ~$ cd rfidsite/
pi@raspberrypi ~/rfidsite$
pi@raspberrypi ~/rfidsite$ python manage.py runserver 192.168.2.2:8080
Performing system checks...

System check identified no issues (0 silenced).
September 07, 2015 - 08:02:23
Django version 1.8, using settings 'rfidsite.settings'
Starting development server at http://192.168.2.2:8080/
Quit the server with CONTROL-C.
```

Figura A.9: Activación del servidor DJANGO.

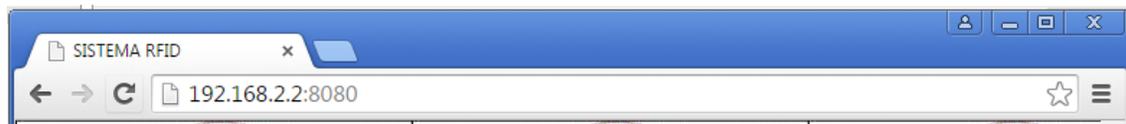


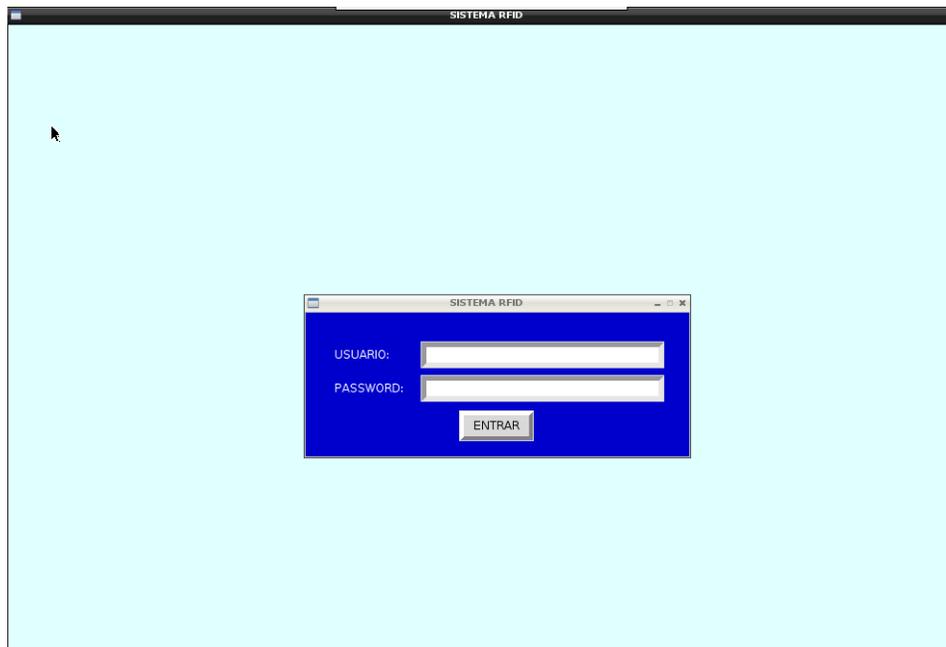
Figura A.10: Conexión con la interface gráfica WEB o con el servidor DJANGO desde el Host del usuario.

A continuación se describirá el funcionamiento de la interfaces gráficas para la iteración con el usuario:

Al ingresar al sistema, este le pide al usuario identificarse o autenticarse, esto lo realiza ingresando un usuario y un password, como se observa en la figura A.11 a y b, si al sistema no se ha ingresado ningún usuario en la base de datos de usuarios entonces utilizar los datos:

USUARIO: **sysrfid**
PASSWORD: **sysrfid**

Este es el usuario por defecto que contiene el sistema, para ingresar cuando el sistema no tiene registrado ningún usuario o se necesite un ingreso por una puerta trasera del sistema, se le advierte al usuario utilizar al usuario por defecto con moderación ya que este no se puede eliminar del sistema.



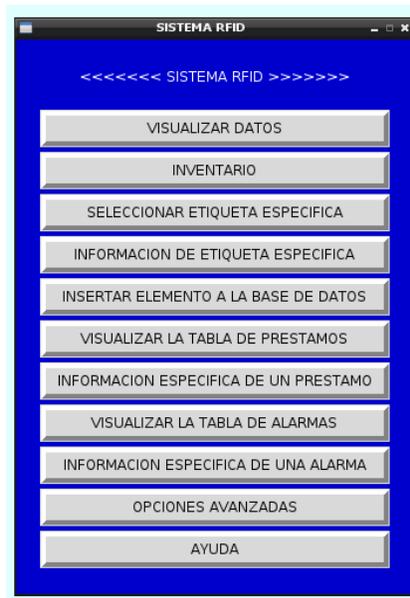
a)



b)

Figura A.11: Autenticación del usuario con el sistema de seguridad RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Después de que el usuario se autentique con el sistema, ingresara al menú principal del mismo, en este menú se encuentran las funciones básicas del sistema, para una mayor comprensión dichas funciones se describirán a continuación:



a)



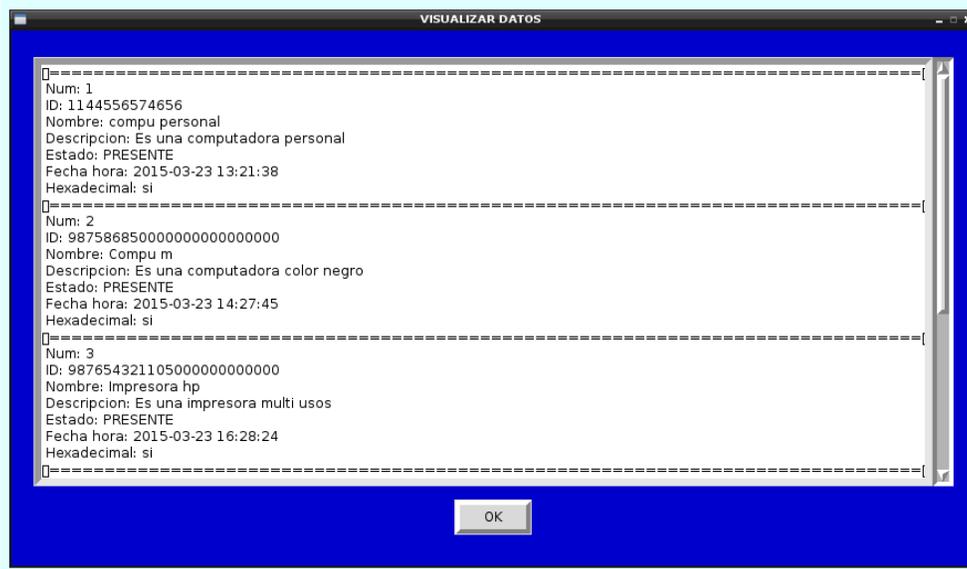
b)

Figura A.12: Menú principal del sistema, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

- **VISUALIZAR DATOS.**

El sistema muestra al usuario todos los datos relacionados con los elementos registrados en la base de datos, dicha información es:

- **ID:** Es el número de identificación o de inventario que se le ha otorgado al elemento, este número proviene de la etiqueta RFID adherida al elemento, este número debe ser de 24 caracteres hexadecimales máximo.
- **Nombre:** El nombre que el usuario le otorga al elemento para diferenciarlo.
- **Descripción:** Datos extras sobre el elemento que el usuario piense que son importantes agregarlos, para tener una mejor identificación del elemento.
- **Estado:** Es una etiqueta textual que se le otorga al elemento para conocer la situación del mismo, esta etiqueta divide en tres tipos:
 - **PRESENTE:** Le informa al usuario que el elemento se encuentra físicamente en el rango de escaneo del sistema.
 - **PRESTADO:** Le informa a usuario que el elemento ha salido del rango de escaneo del sistema con permiso.
 - **AUSENTE:** Le informa al usuario que el elemento o se detecta en el rango de escaneo del sistema, y que no tiene permisos de salir del mismo.
- **Fecha y Hora:** Le informa la hora y la fecha en que el elemento fue ingresado a la base de datos principal del sistema.



a)



b)

Figura A. 13: Visualizar base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Al presionar el botón OK en la interface situada en el módulo de control, el sistema le preguntara al usuario si desea visualizar de nuevo la base de datos, si el usuario acepta se visualizara de nueva la base de datos, pero si no acepta el usuario el sistema regresara al menú principal; y al presionar el botón MENU PRINCIPAL de la interface web el sistema regresara al menú principal.

- **INVENTARIOS.**

El sistema tiene dos formas de realizar escaneos por radiofrecuencias, las cuales se ven en la figura A.14:



a)



b)

Figura A.14: Tipos de escaneos de radio frecuencia que puede realizar el sistema a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

- **[A] REALIZAR INVENTARIOS Y AL NO DETECTAR UN ELEMENTO REGISTRADO EL SISTEMA GENERA ALARMAS.**

Un inventario es un registro documental de los bienes pertenecientes a una persona o a varias personas, el sistema genera un pulso a una frecuencia determinada, en el rango UHF, más específicamente en el rango de 800-990MHz, luego las etiquetas RFID que estén en el rango del lectura del sistema, responden al pulso con la información de su ID, si el sistema no detecta a algunos elementos el sistema genera alarmas, la primera alarmas es registrada en la base de datos de alarmas para que el sistema tenga un registro de la misma, la segunda alarmas es enviada por un mensaje de texto por medio de la comunicación celular GGSM/SMS, este mansaje de texto de alarma contiene los ID de los elementos que no fueron detectados por el escaneo, el mensaje de texto va dirigido a un usuario determinado por el sistema.

El sistema posee la capacidad de generar tres tipos de inventarios, como se observa en la figura A.15:



a)

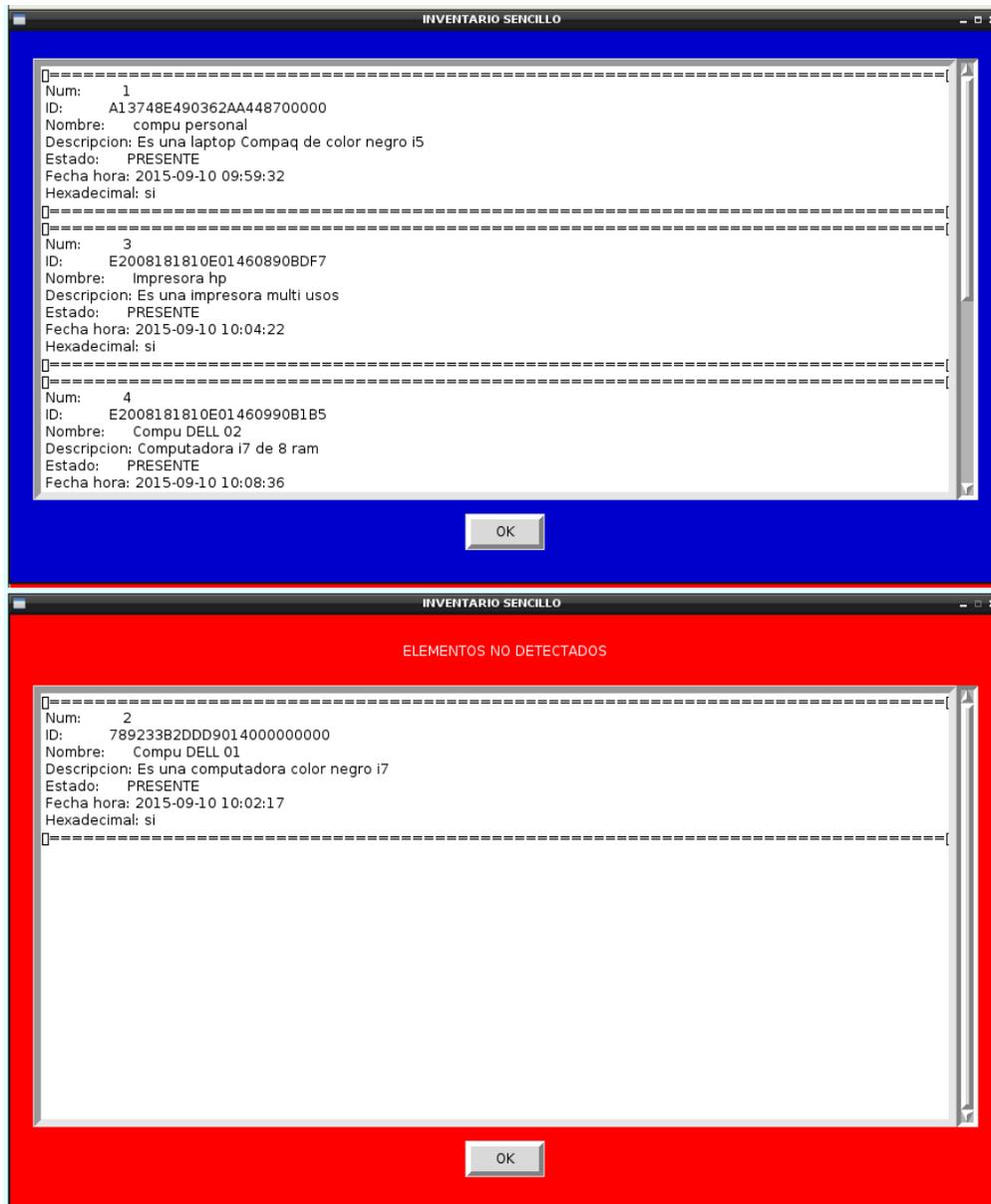


b)

Figura A. 15: Tipos de inventarios que el sistema puede generar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

⇒ **INVENTARIO SENCILLO.**

El resultado de la ejecución de la subfuncion Inventario sencillo mostrando el resultado cuando el sistema no detecta algunos elementos, como se observa en la siguiente figura:



a)

INVENTARIO

num	Nombre	ID	Descripcion	ESTADO	Fecha y Hora	HEX
1	compu personal	A13748E490362AA448700000	Es una laptop Compaq de color negro i5	PRESENTE	10 de Septiembre de 2015 a las 09:59	si
3	Impresora hp	E2008181810E01460890BDF7	Es una impresora multi usos	PRESENTE	10 de Septiembre de 2015 a las 10:04	si
4	Compu DELL 02	E2008181810E01460990B1B5	Computadora i7 de 8 ram	AUSENTE	10 de Septiembre de 2015 a las 10:08	si
5	Router cisco 2600	987654321105000000000000	Se ocupa para realizar practicas de laboratorio	PRESENTE	10 de Septiembre de 2015 a las 10:10	si
6	Router cisco 2800	987586850000000000000000	Se ocupa para realizar practicas de laboratorio	PRESENTE	10 de Septiembre de 2015 a las 10:25	si

ELEMENTOS NO DETECTADOS

num	Nombre	ID	Descripcion	ESTADO	Fecha y Hora	HEX
2	Compu DELL 01	789233B2DDD9014000000000	Es una computadora color negro i7	AUSENTE	10 de Septiembre de 2015 a las 12:23	si

b)

Figura A.16: Resultado de la ejecución de la subfuncion inventario sencillo, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

⇒ INVENTARIO CON RSSI.

El resultado de la ejecución de la subfuncion Inventario con RSSI mostrando el resultado cuando el sistema no detecta algunos elementos, como se observa en la siguiente figura, la ejecución de la subfuncion es igual al de la de inventario sencillo con la diferencia que muestra la información de la frecuencia en que se efectuó el escaneo y el valor de RSSI del mismo:

INVENTARIO CON RSSI

Valor de RSSI:
-30 dBm. [] [] [] []

Frecuencia de operacion:
867500 kHz. []

[]===== []

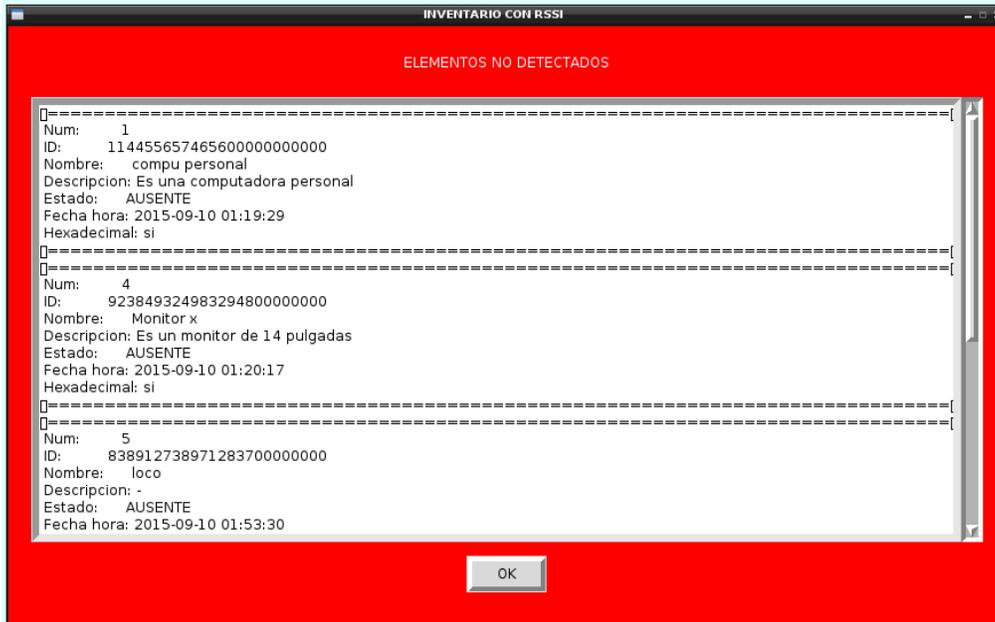
Num: 2
ID: 987586850000000000000000
Nombre: Compu m
Descripcion: Es una computadora color negro
Estado: PRESENTE
Fecha hora: 2015-09-10 03:07:48
Hexadecimal: si

[]===== []

[]===== []

Num: 3
ID: 987654321105000000000000
Nombre: Impresora hp
Descripcion: Es una impresora multi usos
Estado: PRESENTE
Fecha hora: 2015-09-10 03:28:26
Hexadecimal: si

[]===== []



a)

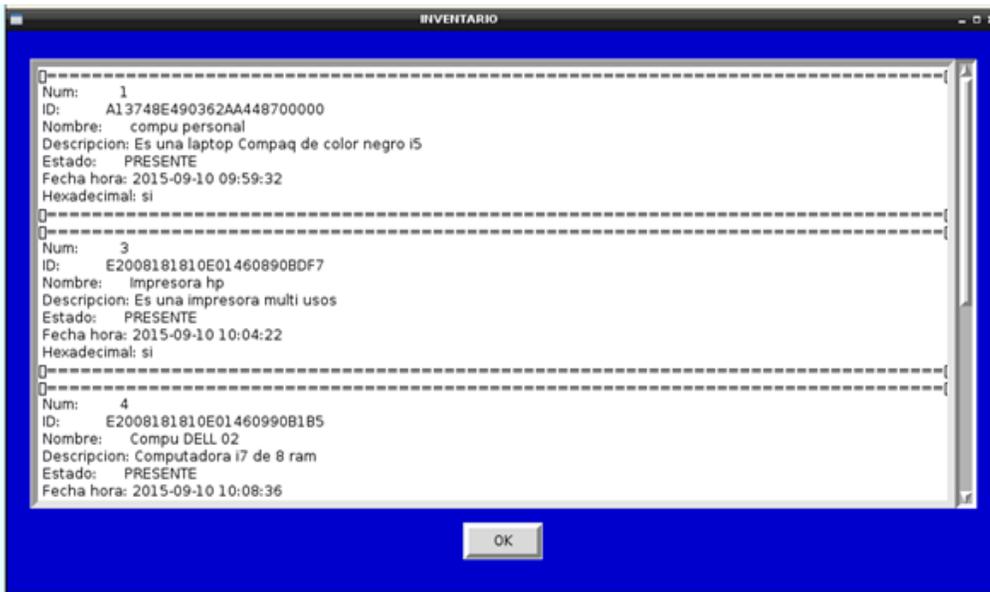


b)

Figura A.17: Resultado de la ejecución de la subfuncion inventario con RSSI, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

⇒ **INVENTARIO CONTINUO.**

El resultado de la ejecución de la subfuncion Inventario continuo mostrando el resultado cuando el sistema no detecta algunos elementos, como se observa en la siguiente figura, la ejecución de la subfuncion es igual al de la de inventario sencillo con la diferencia que los inventarios se realizan automáticamente con un intervalo de tiempo de diferencia, y se detiene al generar una alarma al no detectar un elemento en la zona de control:



a)



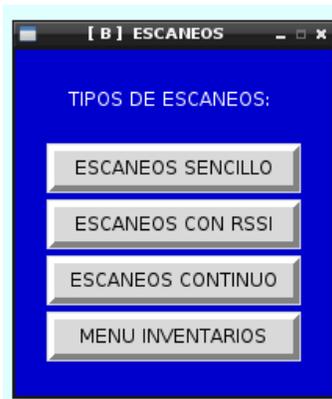
b)

Figura A. 18: Resultado de la ejecución de la subfunción inventario continuo, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

➤ **[B] REALIZAR ESCANEOS Y AL DETECTAR ELEMENTOS REGISTRADOS EL SISTEMA GENERA ALARMAS.**

El sistema genera un escaneo a una frecuencia determinada, utilizando el mismo proceso que la función de inventario, con la diferencia que se genera la alarma cuando el sistema detecta un elemento registrado, esto conlleva a tener un control de las salidas de los elementos de la zona a controlar, generando la alarma cuando un elemento sale sin permiso.

El sistema posee la capacidad de generar tres tipos de escaneos, como se observa en la figura A.19:



a)



b)

Figura A.19: Tipos de escaneos que el sistema puede generar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

⇒ ESCANEEO SENCILLO.

El resultado de la ejecución de la subfunción Inventario sencillo mostrando el resultado cuando el sistema no detecta algunos elementos, como se observa en la siguiente figura:



a)

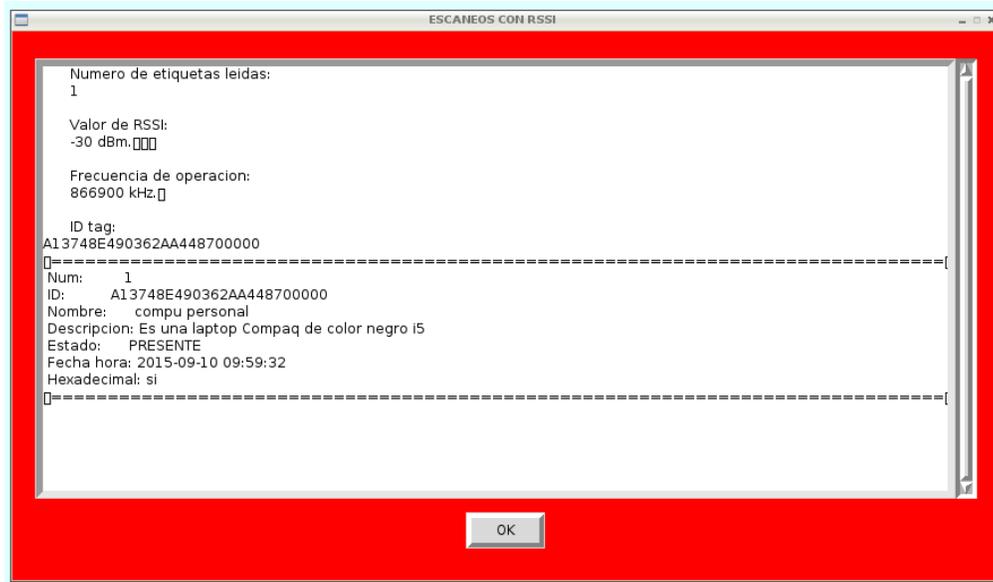


b)

Figura A.20: Resultado de la ejecución de la subfuncion escaneo sencillo, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

⇒ ESCANEEO CON RSSI.

El resultado de la ejecución de la subfuncion Escaneo con RSSI mostrando el resultado cuando el sistema no detecta algunos elementos, como se observa en la siguiente figura, la ejecución de la subfuncion es igual al de la de escaneo sencillo con la diferencia que muestra la información de la frecuencia en que se efectuó el escaneo y el valor de RSSI del mismo:



a)

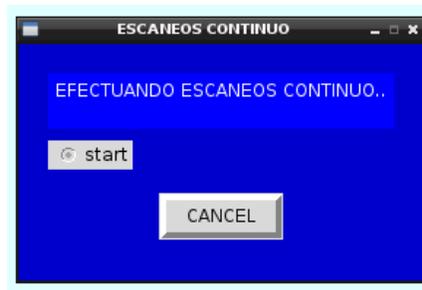


b)

Figura A.21: Resultado de la ejecución de la subfunción escaneo con RSSI, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

⇒ **ESCANEAO CONTINUO.**

El resultado de la ejecución de la subfunción Escaneo continuo mostrando el resultado cuando el sistema no detecta algunos elementos, como se observa en la siguiente figura, la ejecución de la subfunción es igual al de la de escaneo sencillo con la diferencia que los escaneos se realizan automáticamente con un intervalo de tiempo de diferencia, y se detiene al generar una alarma y detectar una salida de un elemento sin permiso:



a)

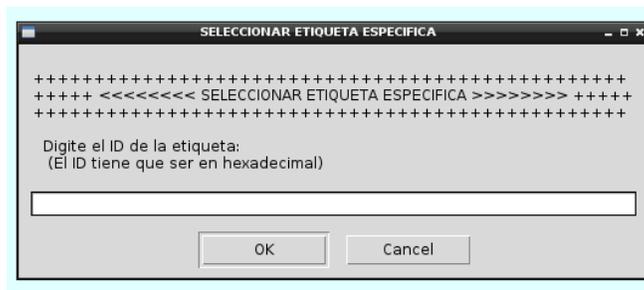


b)

Figura A.22: Resultado de la ejecución de la subfunción escaneo continuo, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

- **SELECCIONAR ETIQUETA ESPECIFICA.**

El sistema es capaz de buscar una etiqueta específica físicamente, el sistema genera un escaneo o un pulso a una frecuencia determinada, buscando una etiqueta RFID que contenga el ID que el usuario ha ingresado en la función.



a)



b)

Figura A.23: Seleccionar etiqueta específica, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema le pide al usuario que ingrese el ID de la etiqueta RFID a buscar, el ID debe de ser en formato hexadecimal y como máximo de 24 dígitos, el sistema manda la ordena al módulo RFID, el cual guarda el ID en una pequeña locación de memoria, luego genera un escaneo, y cada etiqueta que responde su ID es comparado con el ID que el usuario ha ingresado a la función, al terminar el módulo RFID responde al sistema con la afirmación o la negación de haber encontrado la etiqueta específica.

NOTA: El usuario debe tener en mente que la memoria del módulo RFID es muy pequeña, cada vez que el sistema ejecuta la función de SELECCIONAR ETIQUETA ESPECIFICA el ID se guarda en la memoria del módulo RFID pero si el sistema ejecuta otra función, dicha memoria del módulo RFID se borra, o si el sistema ejecuta otra vez la función de seleccionar etiqueta específica el contenido de la memoria del módulo RFID se intercambia con el nuevo ID a buscar.



a)

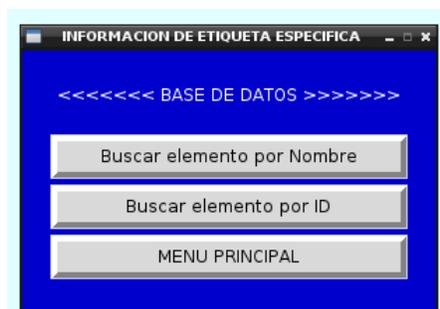


b)

Figura A.24: Advertencia del sistema al usuario de la existencia o no de la etiqueta por ende del elemento, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El resultado de esta función muestra al usuario la existencia física del elemento en el rango del escaneo del módulo RFID. Por último el sistema le pregunta al usuario si desea seleccionar otra etiqueta específica, si el usuario acepta el sistema reinicia la función para que pueda buscar otra etiqueta, pero si el usuario no acepta el sistema termina la función y regresa al menú principal del sistema.

- **INFORMACION DE ETIQUETA ESPECÍFICA.**



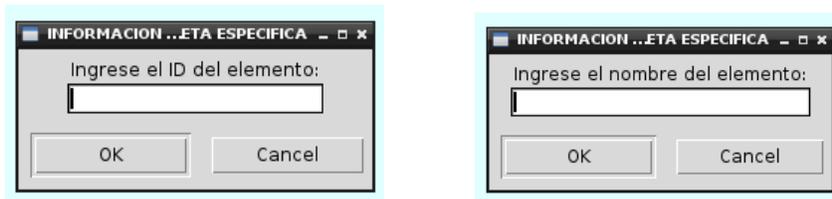
a)



b)

Figura A.25: Menú de la función información de la etiqueta específica, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función es capaz de mostrarle al usuario la información específica de un elemento registrado, se le pide al usuario elegir la forma en que se buscara la información, la primera forma es por medio de su ID, el sistema le pide al usuario ingresar el ID del elemento a buscar, este ID debe de ser en formato hexadecimal y como máximo debe de contener 24 dígitos, y la segunda forma en que el sistema busca la información del elemento es por el NOMBRE del elemento a buscar, el sistema advierte al usuario si no ha ingresado la información, si el sistema encuentra la información del elemento en la base de datos la información será mostrada al usuario y de igual manera si la información del elemento no es encontrada, el sistema lo advertirá que la etiqueta no ha sido encontrada.

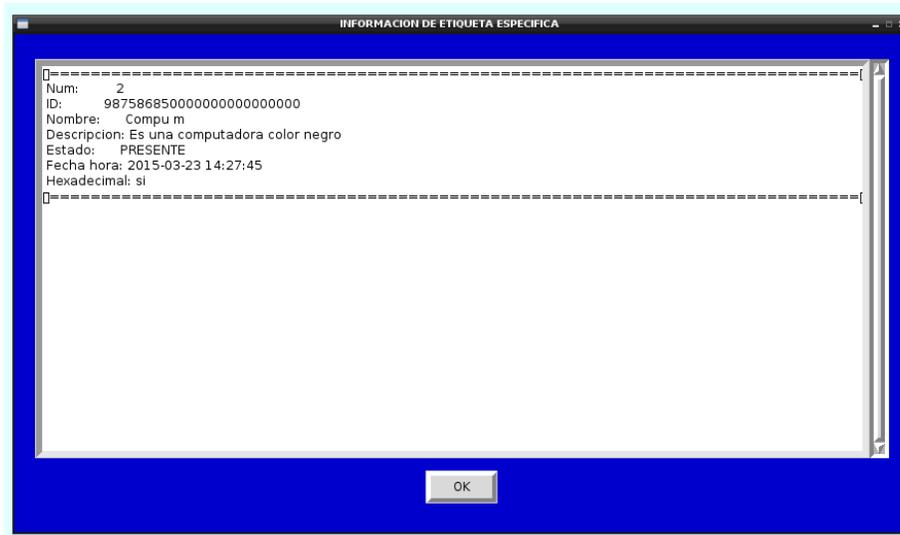


a)

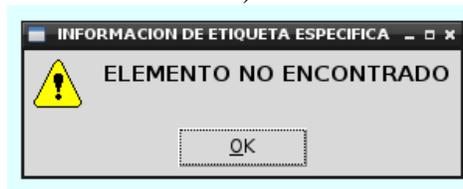


b)

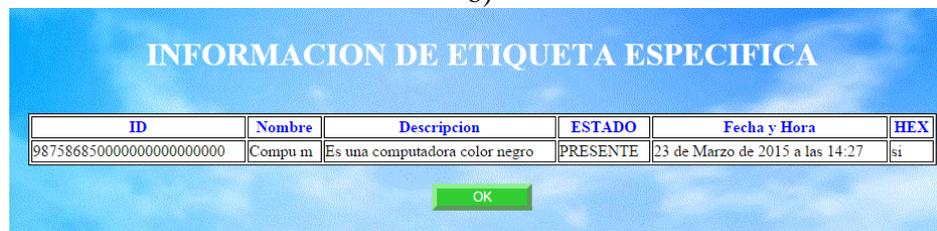
Figura A.26: Formas es que el sistema busca la información específica del elemento, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



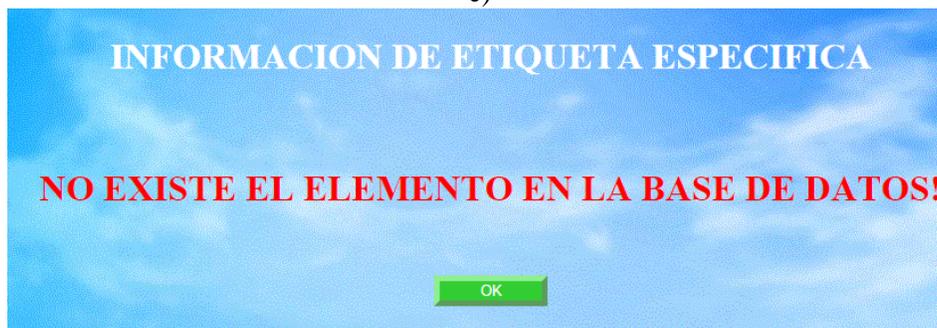
a)



b)



c)

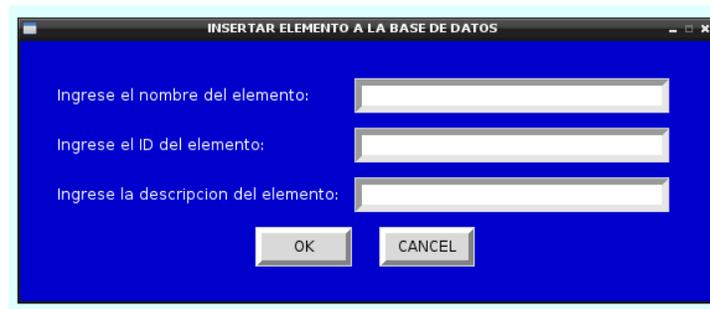


d)

Figura A.27: Resultado de la búsqueda de la información de una etiqueta específica, a) El elemento fue encontrado, interface gráfica situada en el módulo de control, b) El elemento fue no encontrado, interface gráfica situada en el módulo de control, c) El elemento

Por ultimo al presionar el botón OK, de la ventana que se observa en la figura A.27, la función regresa al inicio y le pregunta la forma en que el sistema buscara la información del elemento.

- **INCERTAR ELEMENTO A LA BASE DE DATOS.**



a)



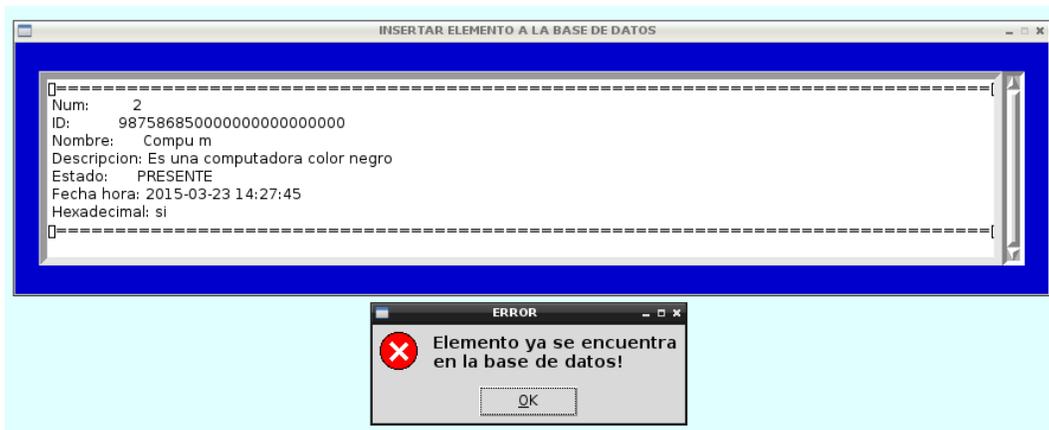
b)

Figura A.28: Ventana de la función de insertar elemento a la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

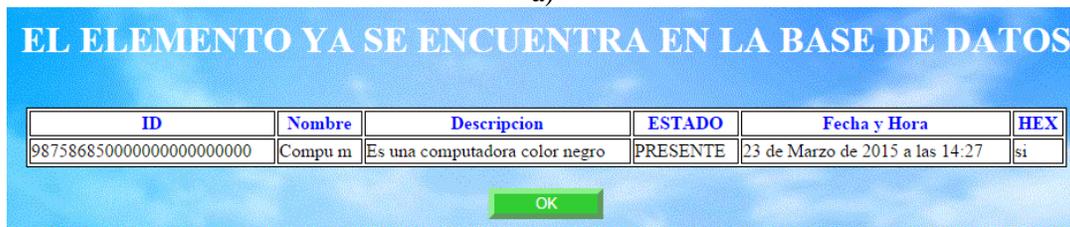
La función despliega al usuario una ventana para que ingrese la información pertinente al elemento a registrar, dicha información es:

- **ID:** Número de identificación que se le otorgara al elemento, es el mismo número de identificación que posee la etiqueta RFID adherida al elemento, dicho número es formado por 24 números en formato hexadecimal.
- **NOMBRE:** Es el nombre que el usuario le asigna al elemento para poder diferenciarlo de los demás.
- **Descripción:** Son datos extras que el usuario decida que son pertinentes para especificar el elemento.

Al ingresar los datos el sistema los verifica, y le advertirá al usuario si ha cometido algún error al ingresar dichos datos, debe tener presente que los datos de ID y el NOMBRE del elemento son datos obligatorios, y el sistema advierte si esos datos no han sido ingresados, pero los datos de la DESCRIPCION no son obligatorios el usuario es libre de ingresar o no, dicha información; luego que se ha ingresado la información verifica si el elemento ha sido ingresado anteriormente, evitando así copias innecesarias del mismo elemento, después que el sistema ha verificado que la información del elemento no ha sido ingresada anteriormente, la función procede a escribir o registrar la información en la base de datos principal, luego la función muestra la información del elemento al usuario afirmando que se escribió correctamente la información en la base de datos principal, pero si ha ocurrido algún error al registrar la información en la base de datos el sistema advertirá.



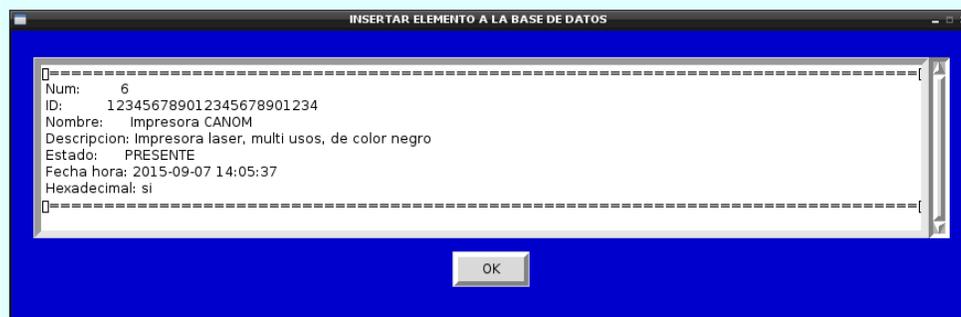
a)



b)

Figura A.29: Advertencia de que ya existe el elemento que el usuario quiere registrar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Luego que el sistema advierte al usuario que el elemento ya ha sido registrado anteriormente, y ya se encuentra en la base de datos, si se presiona el botón OK la función regresara a su inicio pidiendo de nuevo que se ingrese los datos necesarios para registrar un elemento en la base de datos principal.



a)



b)

Figura A.30: Resultado de la función de insertar elemento a la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

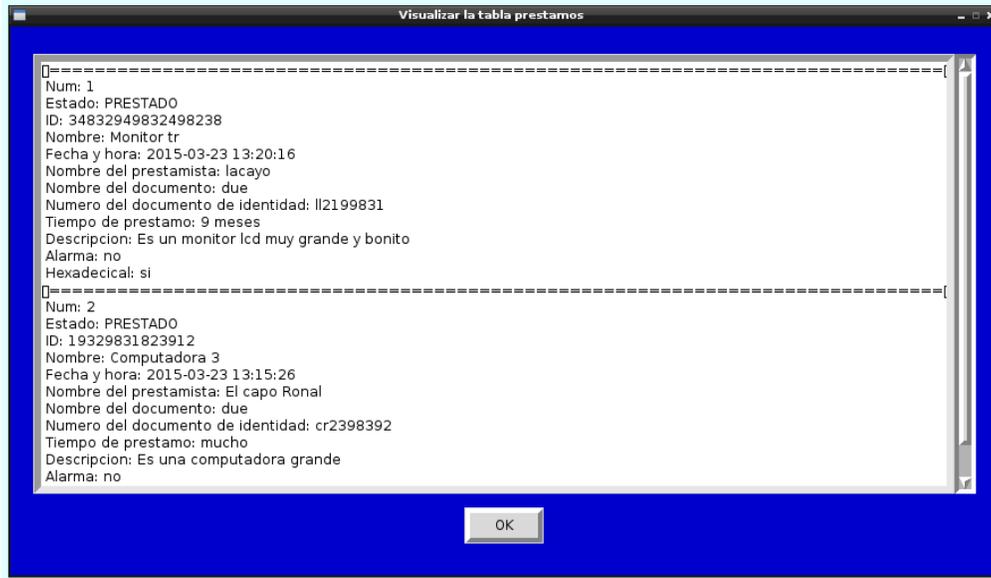
Y por último cuando el sistema ha registrado correctamente la información del elemento, como se observa en la figura A.30, el usuario presiona el botón OK, el sistema le pregunta si quiere ingresar un nuevo elemento, si el usuario acepta la función se reinicia y pide los datos del nuevo elemento, pero si no acepta la función termina y regresa al menú principal.

- **VISUALIZAR LA TABLA DE PRÉSTAMOS.**

El sistema muestra al usuario todos los datos relacionados con los elementos registrados en la base de datos de préstamos, el contenido de esta base de datos hace referencia a los elementos prestados, dicha información es:

- **Estado:** Es una etiqueta textual que se le otorga al elemento para conocer la situación del mismo, en este caso la etiqueta es PRESTADO, informado que el elemento ha sido prestado.
- **ID:** Es el número de identificación o de inventario que se le ha otorgado al elemento, este número proviene de la etiqueta RFID adherida al elemento, este número debe ser de 24 caracteres hexadecimales máximo.
- **Nombre:** El nombre que el usuario le otorga al elemento para diferenciarlo.
- **Fecha y Hora:** Le informa la hora y la fecha en que el elemento fue ingresado a la base de datos de préstamos del sistema, por ende el tiempo en que fue prestado el elemento.
- **Nombre de quien realizo el préstamo:** Es el nombre de la persona que realiza el préstamo del elemento.
- **Nombre del documento de identidad de quien realiza el préstamo:** Es el nombre de algún documento que demuestre la identidad de la persona que realiza el préstamo.

- **Número del documento de identidad de quien realiza el préstamo:** Es un número que lo relacione con el documento de identidad que proporciono la persona que realiza el préstamo.
- **Tiempo de préstamo:** Es el tiempo que el usuario decida pertinente que dure el préstamo del elemento.
- **Descripción:** Datos extras sobre el elemento que el usuario piense que son importantes agregarlos, para tener una mejor identificación del elemento prestado.



a)

VISUALIZAR BASE DE DATOS DE PRESTAMOS

Estado	ID	Nombre	Fecha y Hora	Nombre de quien realiza el prestamo	Nombre del documento de quien realiza el prestamo	Numero del documento de quien realiza el prestamo	Tiempo que durara el prestamo	Descripcion	Alarma	HEX
PRESTADO	34832949832498238	Monitor tr	23 de Marzo de 2015 a las 13:20	lacayo	due	ll2199831	9 meses	Es un monitor lcd muy grande y bonito	no	si
PRESTADO	01272374833122731	Computadora 3	8 de Septiembre de 2015 a las 08:08	El capo Ronal	due	cr2398392	mucho	Es una computadora grande	no	si
PRESTADO	123456789000	Leti	8 de Septiembre de 2015 a las 10:59	Saul Rosa	due	he8327483	dos semanas	Es una impresora de color negro	no	si

MENU PRINCIPAL

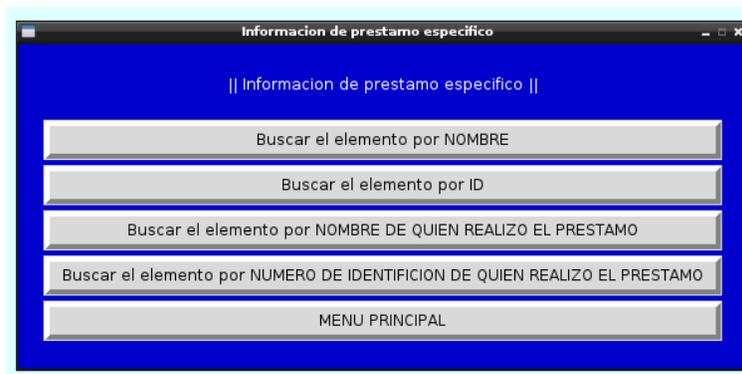
b)

Figura A.31: Visualizar base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Al presionar el botón OK en la interface situada en el módulo de control, el sistema le preguntara si desea visualizar de nuevo la base de datos de préstamos, si el usuario acepta se visualizara de nueva

la base de datos, pero si no acepta el usuario el sistema regresara al menú principal; y al presionar el botón MENU PRINCIPAL de la interface web regresara al menú principal.

- **INFORMACION ESPECÍFICA DE UN PRESTAMO.**



a)

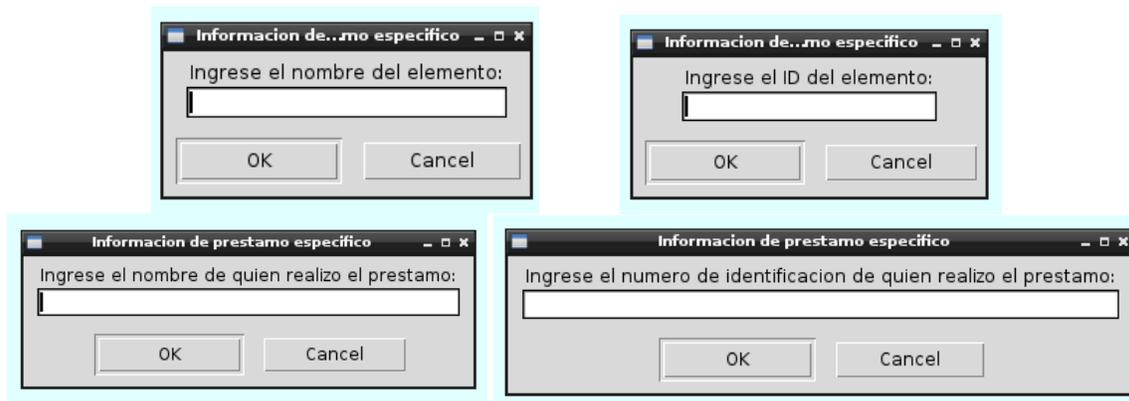


b)

Figura A.32: Menú de la función información de un préstamo específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función es capaz de mostrarle al usuario la información específica de un elemento registrado en la base de datos de préstamos, la función le pide al usuario elegir la forma en que el sistema buscara la información, la primera forma es por medio de su ID, el sistema le pide al usuario ingresar el ID del elemento a buscar, este ID debe de ser en formato hexadecimal y como máximo debe de contener 24 dígitos, la segunda forma en que el sistema busca la información del elemento específico es por el NOMBRE del elemento a buscar, la tercera forma es buscar por el NOMBRE DE QUIEN REALIZO EL PRESTAMO este es el nombre de la persona que efectuó el préstamo y el usuario debe tener en mente que esta información puede contener un solo nombre como todo un el nombre de la persona (nombre y apellidos) esto depende del usuario que lo registro y por ende así se debe de buscar; y la cuarta forma es con el NUMERO DEL DOCUMENTO DE IDENTIDAD

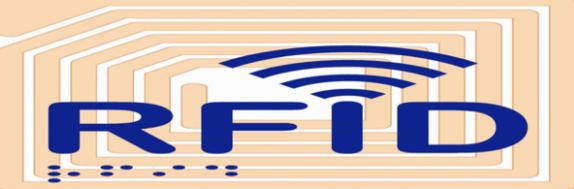
de la persona quien realizo el préstamo, este documento, puede ser cualquier documento que identifique a la persona, pero se debe buscar la información de préstamo con el número del documento con que fue registrado dicho préstamos y dicha persona; si el sistema encuentra la información del elemento en la base de datos de préstamos dicha información será mostrada al usuario y de igual manera si la información del elemento no es encontrada, el sistema lo advertirá.



a)



INFORMACION DE PRESTAMOS ESPECIFICO



Buscar el elemento por NOMBRE

Buscar el elemento por ID

Buscar el elemento por NOMBRE DE QUIEN REALIZO EL PRESTAMO

Buscar el elemento por NUMERO DE IDENTIFICACION DE QUIEN REALIZO EL PRESTAMO

MENU PRINCIPAL

BUSCAR PRESTAMO POR ID

INGRESE EL ID DEL ELEMENTO DEL PRESTAMO:

BUSCAR

CANCEL

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

INFORMACION DE PRESTAMOS ESPECIFICO



Buscar el elemento por NOMBRE

Buscar el elemento por ID

Buscar el elemento por NOMBRE DE QUIEN REALIZO EL PRESTAMO

Buscar el elemento por NUMERO DE IDENTIFICACION DE QUIEN REALIZO EL PRESTAMO

MENU PRINCIPAL

BUSCAR EL ELEMENTO POR EL NOMBRE DE QUIEN REALIZO EL PRESTAMO

INGRESE EL NOMBRE DE LA PERSONA QUE REALIZO EL PRESTAMO:

BUSCAR

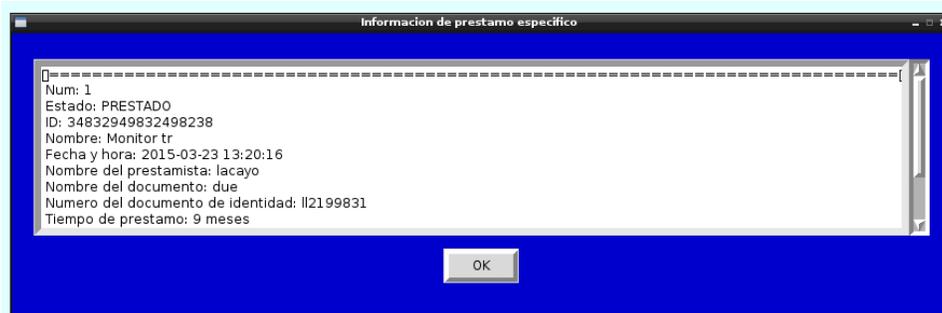
CANCEL

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

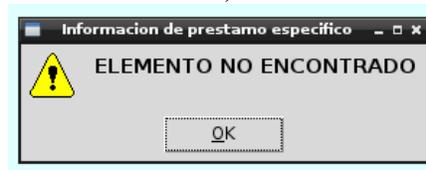


b)

Figura A.33: Formas es que el sistema busca la información específica del préstamo, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



a)



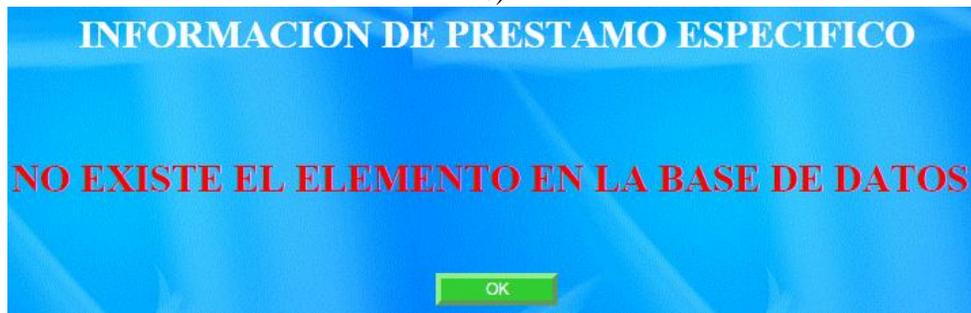
b)

INFORMACION DE PRESTAMO ESPECIFICO

Estado	ID	Nombre	Fecha y Hora	Nombre de quien realiza el prestamo	Nombre del documento de quien realiza el prestamo	Numero del documento de quien realiza el prestamo	Tiempo que durara el prestamo	Descripcion	Alarma	HEX
PRESTADO	34832949832498238	Monitor tr	23 de Marzo de 2015 a las 13:20	lacayo	due	112199831	9 meses	Es un monitor lcd muy grande y bonito	no	si

OK

c)



d)

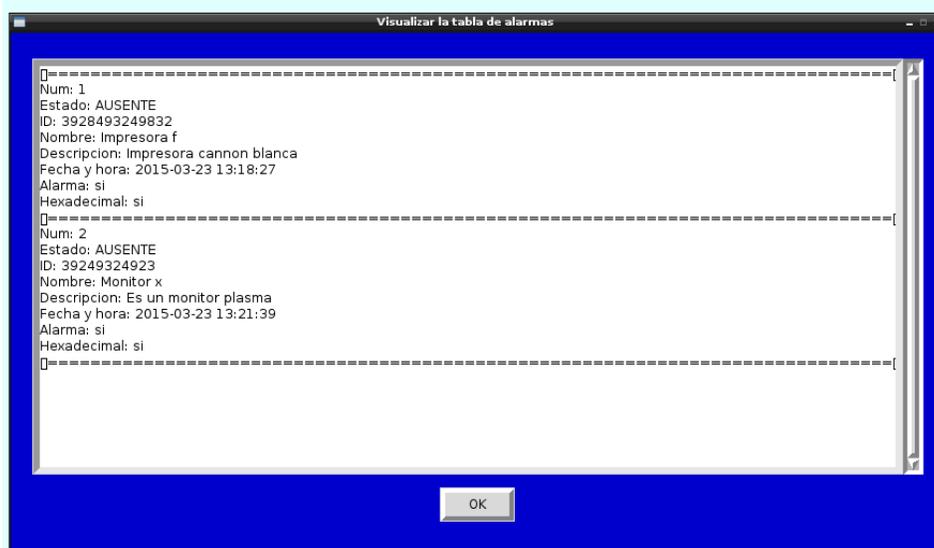
Figura A.34: Resultado de la búsqueda de la información de un préstamo específico, a) El préstamo fue encontrado, interface gráfica situada en el módulo de control, b) El préstamo fue no encontrado, interface gráfica situada en el módulo de control, c) El préstamo fue encontrado, interface WEB y d) El préstamo fue no encontrado, interface WEB.

Por ultimo al presionar el botón OK, de la ventana que se observa en la figura A.34, la función regresa al inicio y le pregunta la forma en que el sistema buscara la información del préstamo específico.

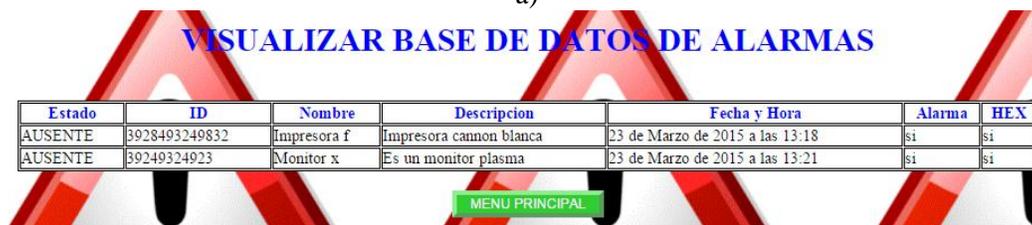
- **VISUALIZAR LA TABLA DE ALARMAS.**

El sistema muestra al usuario todos los datos relacionados con los elementos registrados en la base de datos de alarmas, la información que se hace referencia es la información de los elementos que no están presentes físicamente en el rango de escaneo del sistema, que no se encuentra en préstamo y que ha generado alarmas, dicha información es:

- **Estado:** Es una etiqueta textual que se le otorga al elemento para conocer la situación del mismo, en este caso la etiqueta es AUSENTE, informado que el elemento no se encuentra físicamente.
- **ID:** Es el número de identificación o de inventario que se le ha otorgado al elemento, este número proviene de la etiqueta RFID adherida al elemento, este número debe ser de 24 caracteres hexadecimales máximo.
- **Nombre:** El nombre que el usuario le otorga al elemento para diferenciarlo.
- **Descripción:** Datos extras sobre el elemento que el usuario piense que son importantes agregarlos, para tener una mejor identificación del elemento ausente.
- **Fecha y Hora:** Le informa la hora y la fecha en que el elemento fue ingresado a la base de datos de alarmas del sistema, por ende el tiempo en que no fue detectado por el sistema.



a)

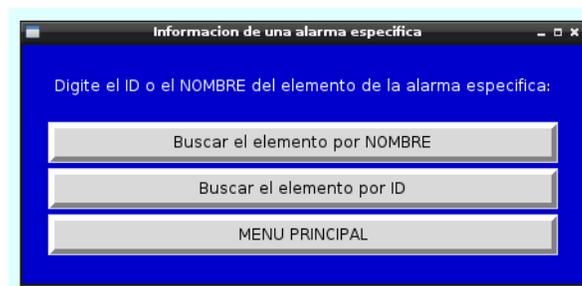


b)

Figura A.35: Visualizar base de datos de alarmas, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Al presionar el botón OK en la interface situada en el módulo de control, el sistema le preguntara al usuario si desea visualizar de nuevo la base de datos de alarmas, si el usuario acepta se visualizara de nueva la base de datos de alarmas, si no acepta el sistema regresara al menú principal; y al presionar el botón MENU PRINCIPAL de la interface web el sistema regresara al menú principal.

- **INFORMACION DE UNA ALARMA ESPECÍFICA.**



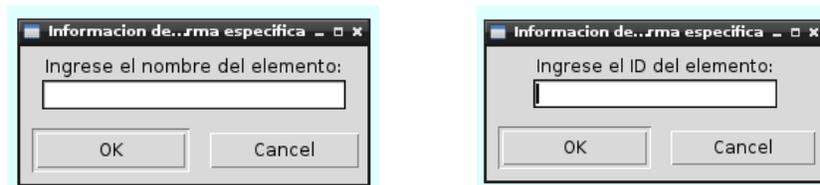
a)



b)

Figura A.36: Menú de la función información de una alarma específica, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función es capaz de mostrarle al usuario la información específica de un elemento registrado en la base de datos de alarmas, la función le pide elegir la forma en que el sistema buscara la información, la primera es por medio de su ID, el sistema le pide ingresar el ID del elemento a buscar, este ID debe de ser en formato hexadecimal y como máximo debe de contener 24 dígitos, y la segunda en que el sistema busca la información del elemento específico es por el NOMBRE del elemento a buscar, si el sistema encuentra la información del elemento en la base de datos será mostrada al usuario y de igual manera si la información no es encontrada.

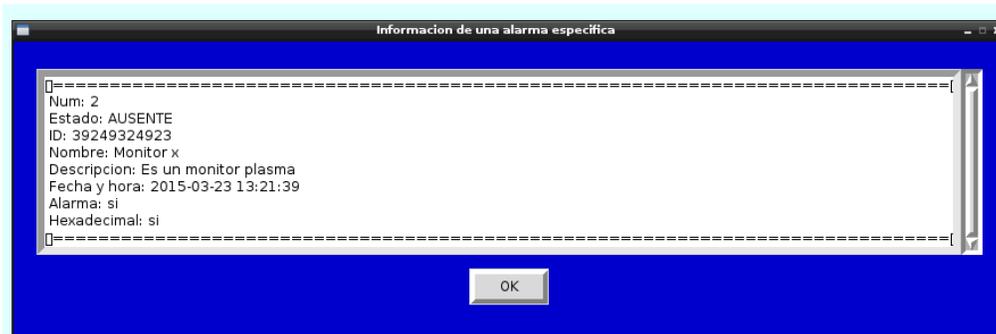


a)



b)

Figura A.37: Formas en que el sistema busca la información específica de una alarma, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



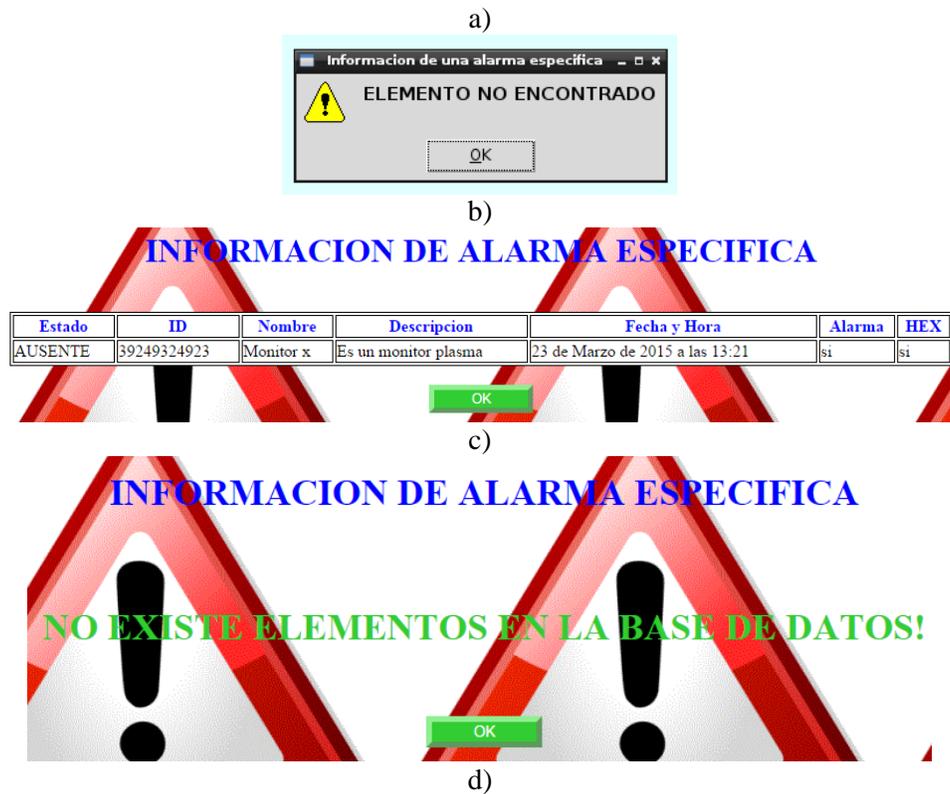


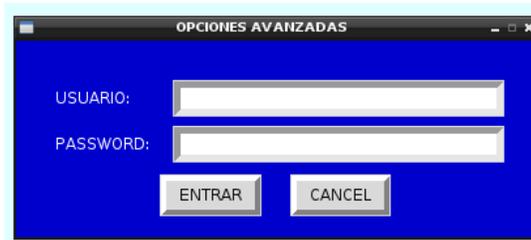
Figura A.38: Resultado de la búsqueda de la información de una alarma específica, a) El elemento fue encontrado, interface gráfica situada en el módulo de control, b) El elemento fue no encontrado, interface gráfica situada en el módulo de control, c) El elemento fue encontrado, Interface gráfica WEB, d) El elemento fue no encontrado, Interface gráfica WEB.

Por ultimo al presionar el botón OK, de la ventana que se observa en la figura A.38, la función regresa al inicio y le pregunta la forma en que el sistema buscara la información del elemento.

- **OPCIONES AVANZADAS.**

El menú de opciones avanzadas contiene las funciones capaces de modificar las funciones físicas del sistema, así como también de eliminar, modificar y agregar información en las bases de datos principal, de préstamos, de alarmas y de usuarios del sistema, por ende el usuario debe tener cuidado.

Para que un usuario pueda ingresar a este menú necesita tener permisos del sistema, el sistema le pide una autenticación para ingresar. En el sistema contiene dos tipos de usuarios, el primero es un usuario normal que solo tiene permisos de ingresar y operar solo las funciones del menú principal, y los segundos tipos de usuarios tienen permisos de súper usuarios estos son los que tienen permiso para ingresar al menú de opciones avanzadas e interactuar con sus funciones.



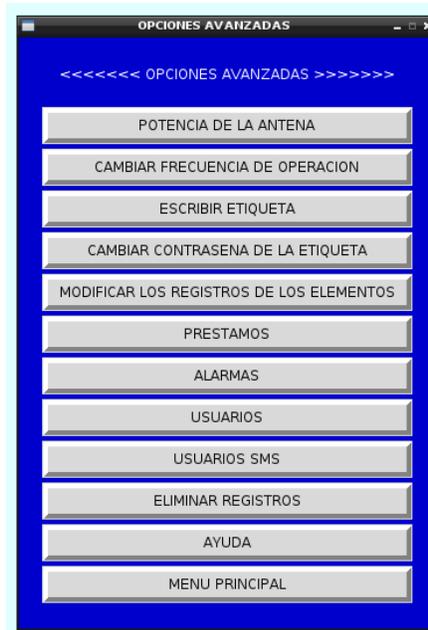
a)



b)

Figura A.39: Autenticación del usuario para ingresar al menú de opciones avanzadas, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Al ingresar el Usuario y Password de un usuario el sistema verificara si dicho usuario tiene permisos para poder ingresar al menú de opciones avanzadas, si tiene dichos permisos el sistema lo dejara ingresar al menú de opciones avanzadas, pero si no, el sistema no lo dejara ingresar al menú.



a)

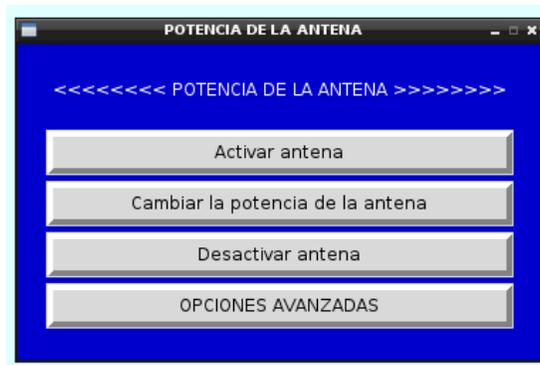


b)

Figura A.40: Menú de opciones avanzadas, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

A continuación se describirá las funciones pertenecientes al menú de opciones avanzadas:

- **POTENCIA DE LA ANTENA.**



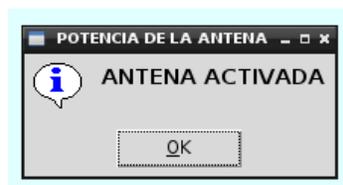
a)



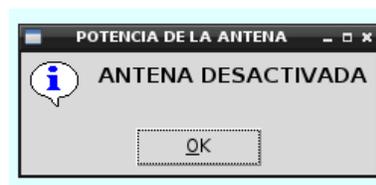
b)

Figura A.41: Menú de la función potencia de la antena, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función de POTENCIA DE LA ANENA, es capaz de modificar el funcionamiento de la antena externa del módulo RFID, como activarla, desactivarla y modificar la potencia de la antena, el usuario debe tener en mente que al iniciar el sistema la antena externa se encuentra desactivada; para que el sistema pueda alcanzar mayor distancia de escaneo la antena externa debe de estar activada.



a)



b)



c)



d)

Figura A. 42: Mensajes de información sobre el estado de la antena cuando el usuario elige sobre activar o desactivar la antena, a) ANTENA ACTIVADA Interface gráfica situada en el módulo de control, b) ANTENA DESACTIVADA Interface gráfica situada en el módulo de control, c) ANTENA ACTIVADA Interface gráfica WEB, d) ANTENA DESACTIVADA Interface gráfica WEB.

Físicamente el usuario puede visualizar si la antena está o no activada, en el módulo RFID como se observa en la figura A.43, el modulo posee un led (cerca del led de power de dicho modulo como se observa en la figura A.43) que señala si la antena se encuentra activada o desactivada, cuando la antena se encuentra desactivada cuando el led se encuentra apagado (figura A.43 a) y la antena se encuentra activada cuando el led se encuentra encendido (figura A.43 b).

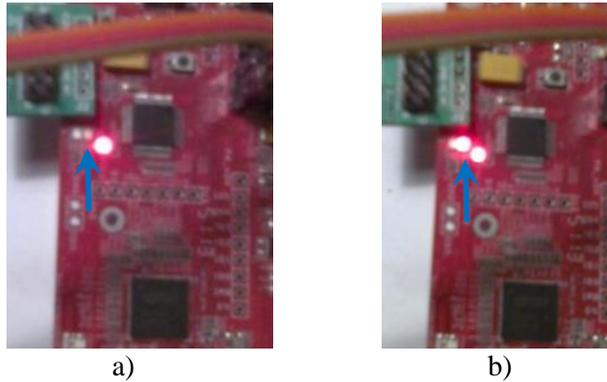
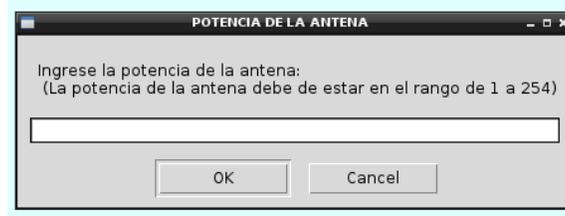


Figura A.43: Visualización del módulo RFID a) Antena desactivada, b) Antena activada.

Al presionar el botón de Cambiar la potencia de la antena, el sistema le permite al usuario cambiar o modificar la potencia de la antena externa del sistema, la potencia de la antena debe de estar en el rango de 1 a 254, y en formato decimal.

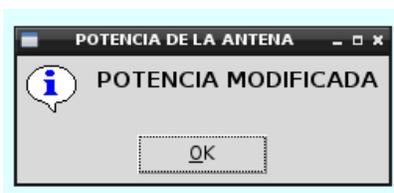


a)



b)

Figura A.44: Ventana para ingresar la nueva potencia de la antena, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



a)



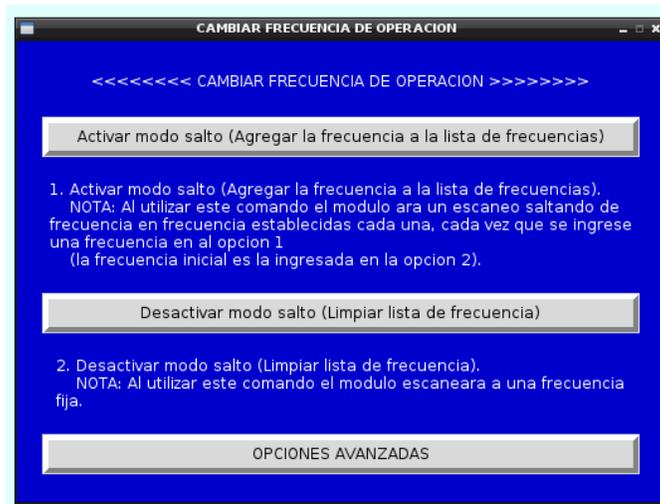
b)

Figura A.45: Mensaje de modificación de la potencia de la antena, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Por último si el usuario presiona el botón de OPCIONES AVANZADAS, el sistema saldrá del menú de potencia de la antena y regresara al menú de opciones avanzadas.

- **CAMBIAR FRECUENCIA DE OPERACIÓN.**

El sistema tiene la capacidad de modificar la frecuencia de operación o la frecuencia a la cual el sistema efectúa los escaneos, estos escaneos se realizan a una frecuencia determinada, en el rango de 800,000KHz a 995,000kHz, la banda en que se efectúa es la banda de UHF, la cual es la que trabaja el módulo RFID.



a)



b)

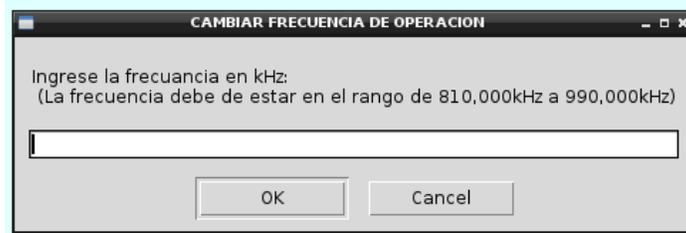
Figura A.46: Menú de la función modificar la frecuencia de operación, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema posee dos formas de realizar los escaneos, estos se basan en un proceso principal llamado MODO SALTO, este proceso se basa en que se realizan los escaneos saltando de frecuencia en frecuencia, esto quiere decir que cada vez que el módulo RFID realiza un escaneo lo realiza con una frecuencia diferente a la anterior; el usuario debe de tener presente, el módulo RFID contiene una lista de frecuencias con la cual efectúa el modo salto, dicha lista se puede modificar y al iniciar el módulo RFID contiene una lista por defecto, las frecuencias son 866300kHz y

866900kHz, cuando se presiona el botón ACTIVAR MODO SALTO el sistema le pide ingresar la frecuencia, debe ingresarla en formato decimal y en kHz, si el usuario comete un error al ingresar la frecuencia el sistema lo advertirá, luego el sistema agregara dicha frecuencia a la lista de frecuencia que posee el módulo RFID con el método de modo salto.

Cuando el usuario presiona el botón DESACTIVAR MODO SALTO el sistema pide ingresar la frecuencia, luego el sistema borrara la lista de frecuencia que tendrá por defecto o anteriormente luego se insertara dicha frecuencia, el sistema realizara los escaneos a una frecuencia fija; pero si el usuario ingresa una nueva frecuencia con la función ACTIVAR MODO SALTO la frecuencia se agregara a la lista de frecuencia del modo salto, teniendo en cuenta que cada vez que un usuario activa la función de Desactivar modo salto, la lista se borrara y se iniciara una nueva lista de frecuencias teniendo como la primera en esa lista la frecuencia ingresada con la función de DESACTIVAR MODO SALTO.

NOTA: Cada vez que el módulo RFID se energiza se eliminara la frecuencia o la lista de frecuencia creada por el usuario y el modulo funcionara con la lista de frecuencias por defecto en el modo salto (las frecuencias de dicha lista son 866300kHz y 866900kHz).



A screenshot of a Windows-style dialog box titled "CAMBIAR FRECUENCIA DE OPERACION". The dialog contains the text: "Ingrese la frecuencia en kHz: (La frecuencia debe de estar en el rango de 810,000kHz a 990,000kHz)". Below the text is a text input field. At the bottom of the dialog are two buttons: "OK" and "Cancel".

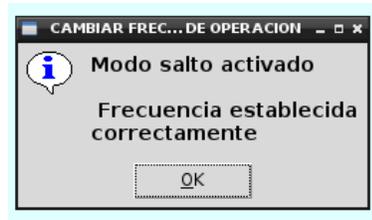
a)



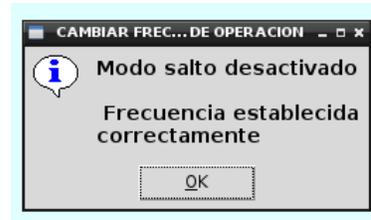
A screenshot of the main application interface. On the left is a vertical menu with a blue header "CAMBIAR LA FRECUENCIA DE OPERACION" and several green buttons: "POTENCIA DE LA ANTENA", "CAMBIAR FRECUENCIA DE OPERACION", "ESCRIBIR ETIQUETA", "CAMBIAR CONTRASENA DE LA ETIQUETA", "MODIFICAR LOS REGISTROS DE LOS ELEMENTOS", "PRESTAMOS", "ALARMAS", "ELIMINAR REGISTROS DE LOS ELEMENTOS", and "MENU PRINCIPAL". To the right of the menu is a large blue area with a circuit pattern background. It features a red and purple antenna icon at the top. Below the icon, the text reads: "ACTIVAR MODO SALTO (Agregar la frecuencia a la lista de frecuencias) (La frecuencia debe de estar en el rango de 810,000kHz a 990,000kHz)". At the bottom of this area are three buttons: a text input field, "Cambiar frecuencia", and "CANCEL". At the very bottom of the screen is a grey bar with the text "SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA".

b)

Figura A.47: Ventana para ingresar la frecuencia de operación al sistema, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



a)



b)



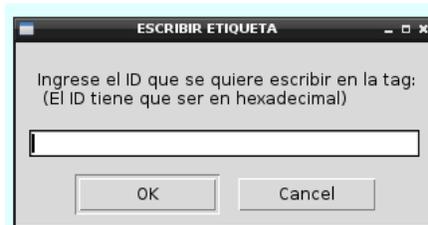
c)



b)

Figura A.49: Ventana para ingresar el password de la etiqueta RFID para poder modificar su ID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función pide que se ingresen el password de la etiqueta RFID, este debe de contener 8 dígitos máximo en formato hexadecimal, si el usuario ha cometido un error al ingresar el password el sistema le advertirá de su error, la función pide que se ingrese el nuevo ID o el que se escribirá en la etiqueta, el nuevo ID debe de contener como máximo 24 dígitos en formato hexadecimal, si el usuario ingresa de forma errónea se le advertirá de su error.



a)



b)

Figura A.50: Ventana para ingresar el nuevo ID de la etiqueta RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El usuario debe tener muy presente que para ejecutar la función ESCRIBIR ETIQUETA, anteriormente debe de haber ejecutado la función SELECCIONAR UNA ETIQUETA ESPECIFICA, al ejecutar esta función el módulo RFID guarda en su memoria el ID de la etiqueta que fue seleccionada, y la función ESCRIBIR ETIQUETA ocupa dicho ID para encontrar y escribir la etiqueta RFID, por ende estas dos funciones debe de ejecutarse una después de la otra, si no es así el ID guardado en la memoria del módulo RFID se borraría y la función de ESCRIBIR ETIQUETA no podrá ejecutarse ya que no sabrá a que etiqueta deberá escribir el nuevo ID, y el sistema le advertirá al usuario de que la etiqueta no fue localizada.



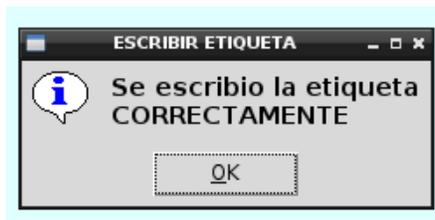
a)



b)

Figura A.51: Advertencias al usuario sobre los errores de no haber encontrado la etiqueta a escribir y/o que el password ingresado por el usuario no es el password de la etiqueta RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El usuario debe tener en cuenta que el proceso de la función de escribir etiqueta solo se puede ejecutar en etiquetas capaces de modificar su ID, ya que también existen etiquetas que no se puede modificar dicho ID; y las etiquetas que si se pueden modificar su ID tiene un password por defecto, el cual es **00 00 00 00**, este password le permite al usuario modificar el ID de la etiqueta RFID.



a)

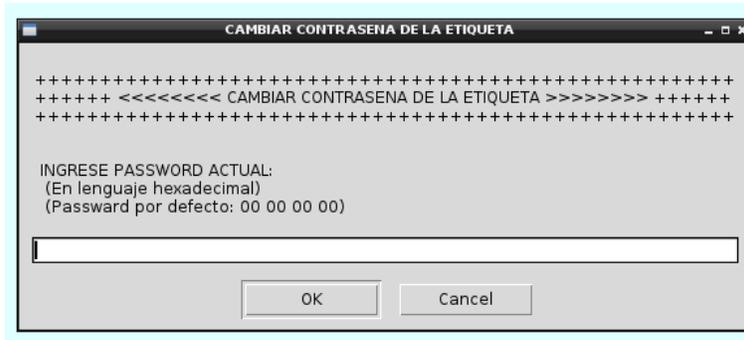


b)

Figura A.52: Resultado de escribir una etiqueta RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

- **MODIFICAR CONTRASEÑA DE LA EITQUETA.**

El sistema es capaz de modificar el password de la etiqueta, este password le permite al usuario tener el permiso de modificar el ID de dicha etiqueta, el password debe de contener como máximo 8 dígitos en formatos hexadecimales, si el usuario comete un error al ingresar dicha información el sistema advertirá de su error.



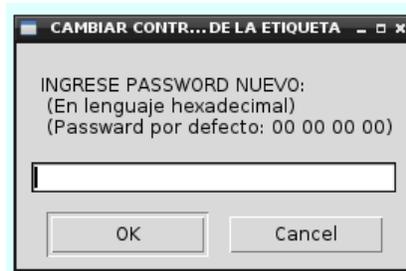
a)



b)

Figura A.53: Ventana para ingresar el password actual que posee la etiqueta a modificar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Luego el sistema pide ingresar el nuevo password, y como el anterior este password debe de contener 8 dígitos en formato hexadecimal, si el usuario comete un error al ingresar los datos el sistema le advertirá de su error.



a)



b)

Figura A.54: Ventana para ingresar el password nuevo que posee la etiqueta a modificar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El usuario debe tener muy presente que para ejecutar la función CAMBIAR LA CONTRASEÑA DE UNA ETIQUETA, anteriormente debe de haber ejecutado la función SELECCIONAR UNA ETIQUETA ESPECIFICA, al ejecutar esta función el módulo RFID guarda en su memoria el ID de la etiqueta que fue seleccionada, y la función CAMBIAR LA CONTRASEÑA DE UNA ETIQUETA ocupa dicho ID para encontrar y modificar el password de la etiqueta, por ende estas dos funciones debe de ejecutarse una después de la otra, si no es así el ID guardado en la memoria del módulo RFID se borrará y la función de CAMBIAR LA CONTRASEÑA DE UNA ETIQUETA no podrá ejecutarse ya que no sabrá a que etiqueta deberá escribir el nuevo password, y el sistema le advertirá al usuario de que la etiqueta no fue localizada.



a)

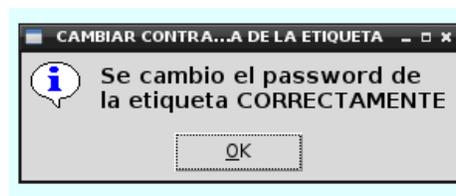




b)

Figura A.55: Advertencias al usuario sobre los errores de no haber encontrado la etiqueta y/o que el password ingresado por el usuario no es el password de la etiqueta RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El usuario debe tener en cuenta que el proceso de la función de cambiar la contraseña de la etiqueta solo se puede ejecutar en etiquetas capaces de modificar su ID; y las etiquetas que si se pueden modificar su información, tienen un password por defecto, el cual es **00 00 00 00**, este password le permite al usuario modificar el ID y su password de la etiqueta RFID.



a)



b)

Figura A.56: Resultado de modificar el password una etiqueta RFID, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

- **MODIFICAR LOS REGISTROS DE LOS ELEMENTOS.**

El sistema es capaz de modificar todos los registros de los elementos en la base de datos principal, el usuario puede modificar el ID, el NOMBRE y la DESCRIPCION del elemento específico, debe tener en mente que el ID que se encuentra en el registro es el que posee la etiqueta físicamente, por ende si se modifica el ID del registro del elemento el sistema nunca notificara que se encuentra el elemento en los escaneos ya que el ID de la etiqueta y e ID del elemento registrado son diferentes.



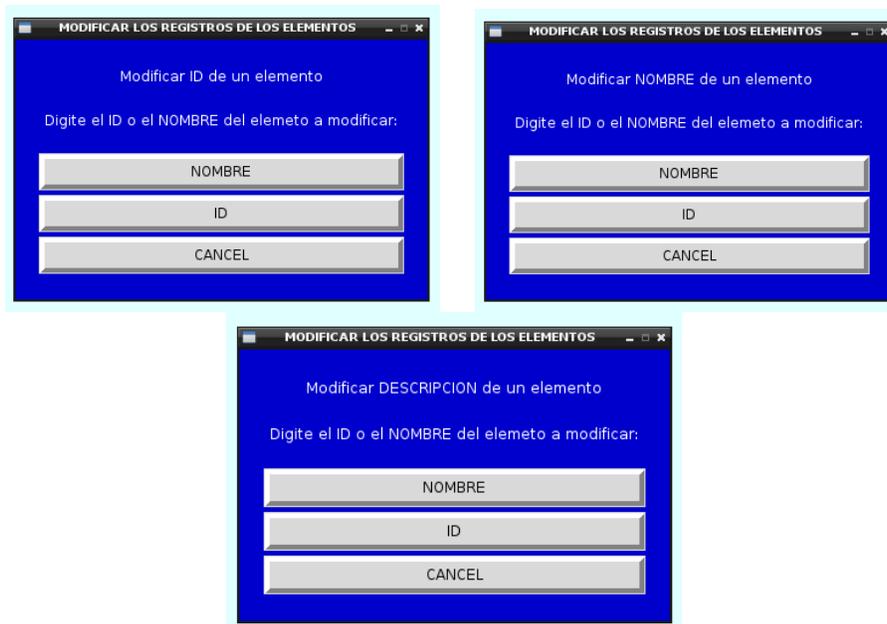
a)



b)

Figura A.57: Menú de la función de modificar los registros de los elementos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Al elegir el registro que se modificara de la información del elemento, el sistema preguntara la forma que buscara la información del elemento, estas dos formas son por el ID o por el NOMBRE del elemento registrado, si existe dicho elemento en la base de datos el sistema prosigue a preguntarle al usuario sobre el nuevo ID, NOMBRE o DESCRIPCION del elemento, la función advertirá al usuario si ha ingresado erróneamente la información del elemento ya sea los nuevos ID, NOMBRE o DESCRIPCION de los elementos a modificar, así como también de los nombres e ID de los elementos para que el sistema busque la información específica.



a)

MODIFICAR LOS REGISTROS DE LOS ELEMENTOS

Modificar NOMBRE de un elemento
Modificar ID de un elemento
Modificar DESCRIPCION de un elemento
OPCIONES AVANZADAS

MODIFICAR ID DE UN ELEMENTO.

DIGITE EL ID O EL NOMBRE DEL ELEMENTO A MODIFICAR:

NOMBRE
ID
CANCEL

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

MODIFICAR LOS REGISTROS DE LOS ELEMENTOS

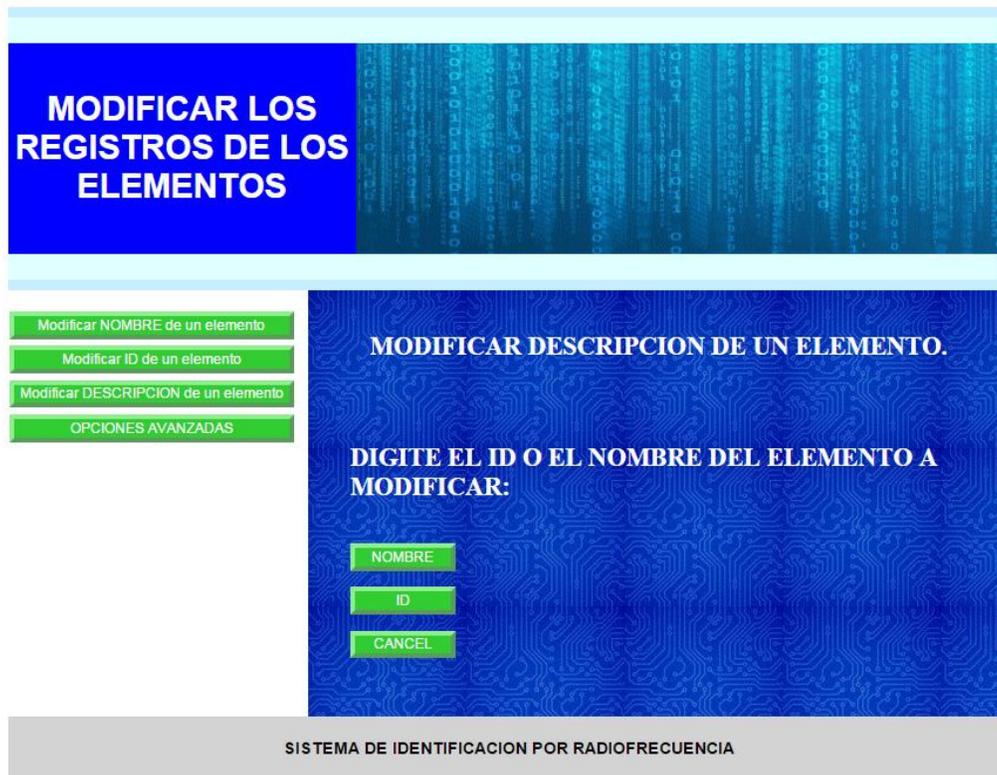
Modificar NOMBRE de un elemento
Modificar ID de un elemento
Modificar DESCRIPCION de un elemento
OPCIONES AVANZADAS

MODIFICAR NOMBRE DE UN ELEMENTO.

DIGITE EL ID O EL NOMBRE DEL ELEMENTO A MODIFICAR:

NOMBRE
ID
CANCEL

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

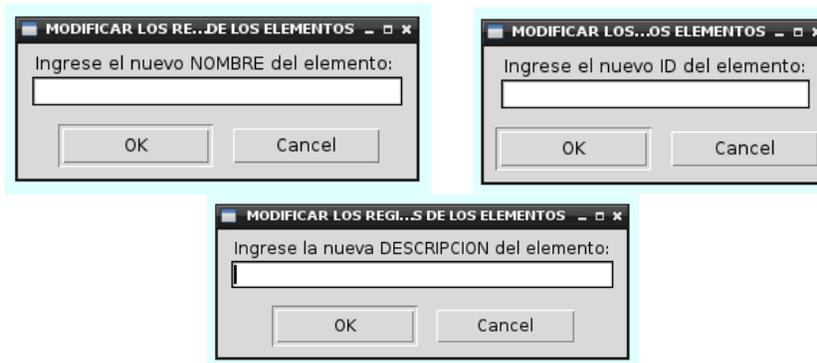


b)

Figura A.58: Opciones de búsqueda del elemento para modificar sus registros, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



Figura A.59: Formas en que el sistema busca la información específica del elemento a modificar, Interface gráfica situada en el módulo de control.



a)

Figura A.60: Ventanas para ingresar ya sea el nuevo ID, el nuevo NOMBRE o el nuevo DESCRIPCION del elemento, Interface gráfica situada en el módulo de control.

MODIFICAR LOS REGISTROS DE LOS ELEMENTOS

Modificar NOMBRE de un elemento
Modificar ID de un elemento
Modificar DESCRIPCION de un elemento
OPCIONES AVANZADAS

MODIFICAR NOMBRE DE UN ELEMENTO.

INGRESE EL NOMBRE DEL ELEMENTO A MODIFICAR:

INGRESE EL NUEVO NOMBRE DEL ELEMENTO A MODIFICAR:

Modificar
CANCEL

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA



Figura A.61: Ejemplo de las ventanas que utiliza la interfaz WEB para modificar los registros de los elementos, se presenta las ventanas para modificar el nombre del elemento y ser elegido en la base de datos ya sea por su nombre actual o por su ID.

Luego que el usuario haya ingresado la nueva información la función buscará la información del elemento y la modificará, si el proceso se efectuó correctamente el sistema mostrará dicha información al usuario, pero si se produjo un error el sistema advertirá al usuario sobre dicho error.



a)

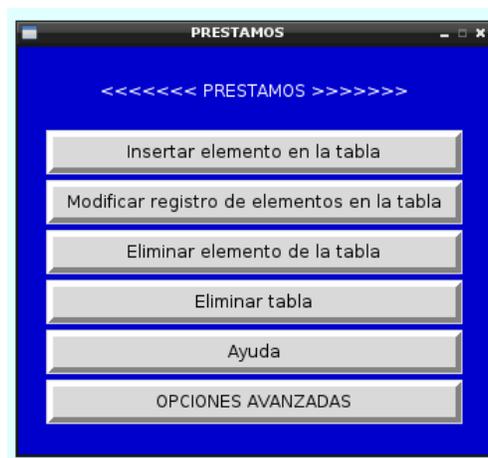


b)

Figura A.62: Resultados de la modificación de los registros de un elemento en específico (en este caso se modificó su NOMBRE), a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Al presionar el botón OK el sistema reiniciará la función dándole oportunidad al usuario de modificar otro registro de un elemento específico.

- **PRESTAMOS.**



a)



b)

Figura A.63: Menú de la función PRESTAMOS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema posee funciones de préstamos, estas funciones modifican, eliminan e ingresan los registros de los elementos que se encuentran en préstamo, a continuación se describirán el funcionamiento de los procesos que abarca la función de préstamos:

⇒ **INSERTAR ELEMENTO EN LA TABLA.**

Insertar elemento en la tabla

Ingrese ID del elemento:

Ingrese nombre de quien lo presta:

Ingrese nombre del documento de identidad de quien lo presta:

Ingrese numero del documento de quien lo presta:

Ingrese la duracion del prestamo:

Ingrese la descripcion del elemento:

OK CANCEL

a)

PRESTAMOS

Insertar elemento en la tabla

Modificar registro de elementos en la tabla de prestamos

Eliminar elemento de la tabla

Eliminar tabla

OPCIONES AVANZADAS

INSERTAR PRESTAMOS A LA BASE DE DATOS DE PRESTAMOS

Ingrese ID del elemento:

Ingrese nombre de quien lo presta:

Ingrese nombre del documento de identidad de quien lo presta:

Ingrese numero del documento de quien lo presta:

Ingrese la duracion del prestamo:

Ingrese la descripcion del elemento:

Insertar

CANCEL

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

b)

Figura A.64: Ventana para ingresar los datos necesarios para un nuevo elemento en la tabla prestamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

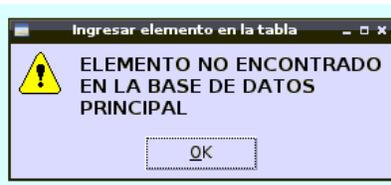
La función despliega una ventana para que el usuario ingrese los datos necesarios para registrar un nuevo elemento en la base de datos de préstamos, estos datos son:

- **ID:** Número de identificación que se le otorgara al elemento, es el mismo número de identificación que posee la etiqueta RFID adherida al elemento.
- **Nombre de la persona que realiza el préstamo:** Es el nombre de la persona que realizara el préstamo.
- **Nombre del documento de identidad de la persona quien realiza el préstamo:** Es el nombre de algún documento que corrobore la identidad del apersona que realiza el préstamo.
- **Número del documento de identidad de la persona quien realiza el préstamo:** Es el número de algún documento de identidad de la persona que realiza el préstamo.

- **Duración del préstamo:** Es el tiempo que el usuario establezca que sea conveniente que dure el préstamo.
- **Descripción:** Son datos extras que el usuario decida que son pertinentes para especificar el elemento.

El usuario debe tener en mente que los datos de ID, nombre de la persona que realizara el préstamo, nombre del documento de identidad de la persona que realizara el préstamo y numero del documento de identidad de la persona que realizara el préstamo, son datos obligatorios, el sistema advertirá al usuario al ingresar los datos erróneamente; los datos duración del préstamo y descripción, no son datos obligatorios por ende si el usuario no los ingresa el sistema lo permitirá y se registrara el elemento en la base de datos de préstamos.

El usuario debe tener presente que los únicos elementos que se puede registrar en la base de datos préstamos, son los elementos que se encuentran registrados en la base de datos principal, esto quiere decir que los usuarios solo pueden prestar elementos que estén registrados, ya que no se pueden prestar elementos que no se tiene el control documental del mismo.



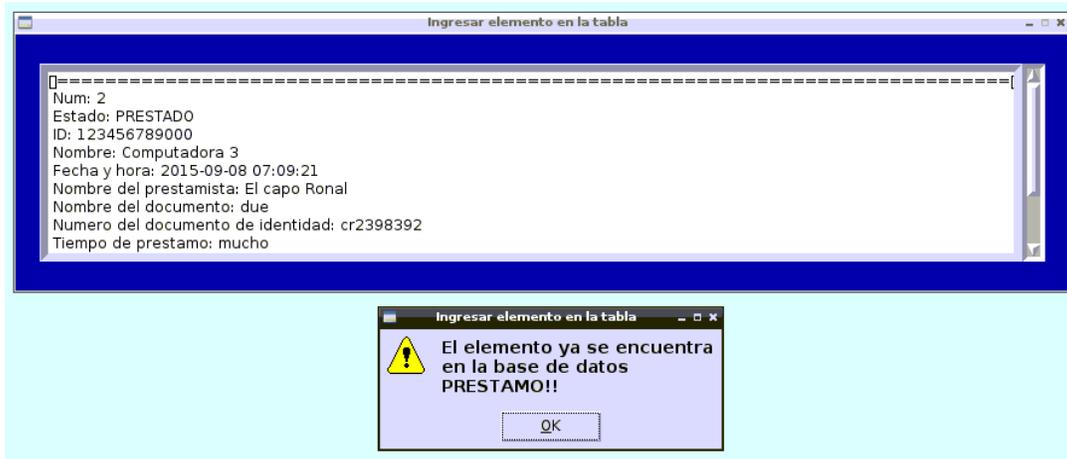
a)



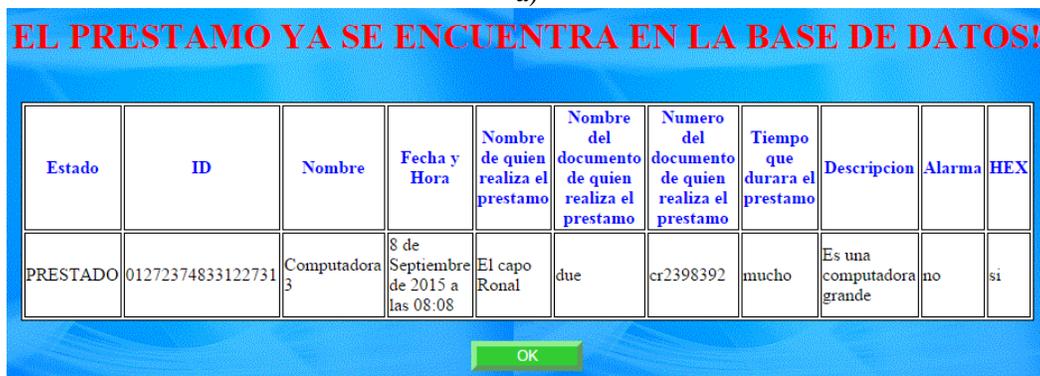
b)

Figura A.65: Ventana de advertencia sobre el préstamo de un elemento que no se encuentra registrado en la base de datos principal, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Se advertirá al usuario si el elemento que se quiera registrar, ya haya sido ingresado anteriormente, mostrando al usuario la información del elemento, y una ventana de advertencia informando al usuario de que el elemento ya ha sido registrado en la base de datos.



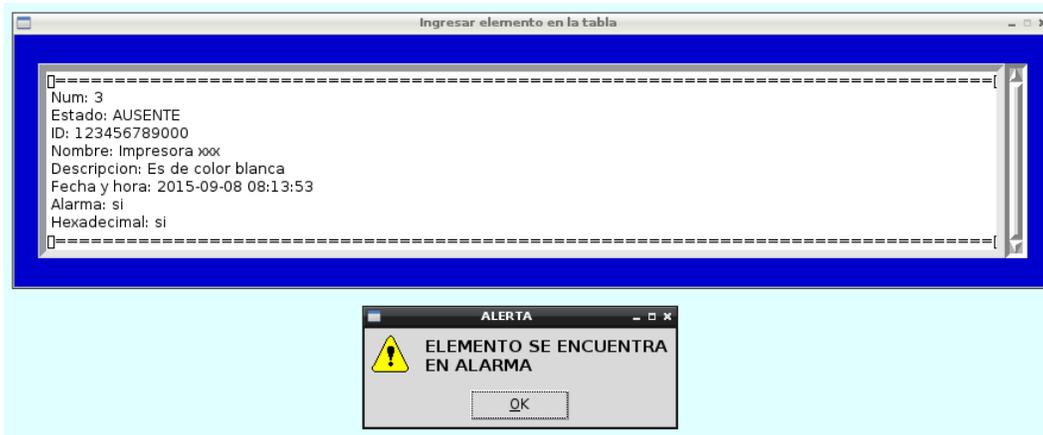
a)



b)

Figura A.66: Ventanas de advertencia al usuario, informando del elemento que ya se encuentra registrado en la base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema también advierte al usuario si la información del elemento se encuentra registrada en la base de datos de alarmas, esto quiere decir que el elemento no se encuentra en el rango de escaneo del módulo RFID y no tiene permiso de préstamo, al dar clic al botón OK el sistema le pregunta si se desea continuar con el registro del elemento, si acepta se borrara la información de la alarma y el elemento se registrara en la base de datos de préstamos, pero si no acepta, la función se reiniciara y no se registrara la información del elemento.

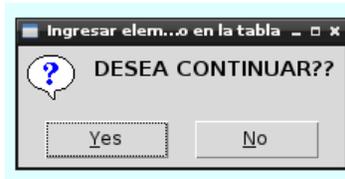


a)



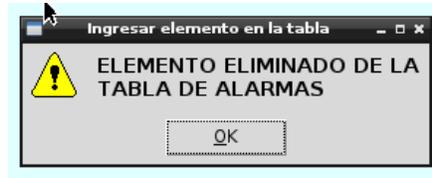
b)

Figura A.67: Ventanas de advertencia al usuario, informando del elemento que se encuentra registrado en la base de datos de alarmas, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB, el sistema interroga al usuario si desea continuar y eliminar la información del elemento en la base de datos de alarmas e ingresarlos en la base de datos de préstamos.



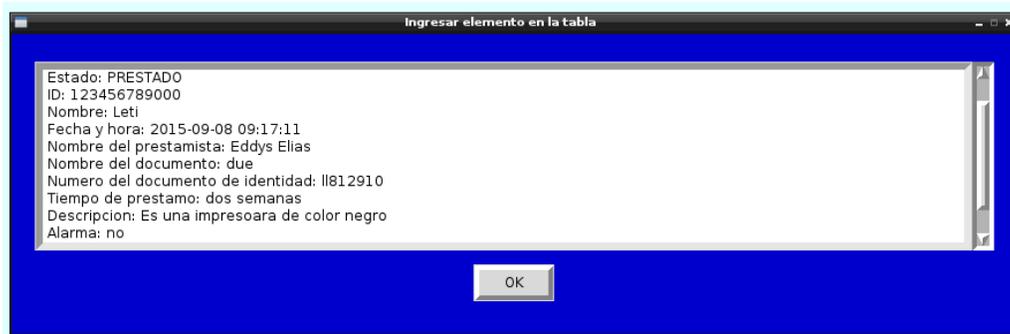
a)

Figura A.68: El sistema interroga al usuario si desea continuar y eliminar la información del elemento en la base de datos de alarmas e ingresarlos en la base de datos de préstamos, a) Interface gráfica situada en el módulo de control.

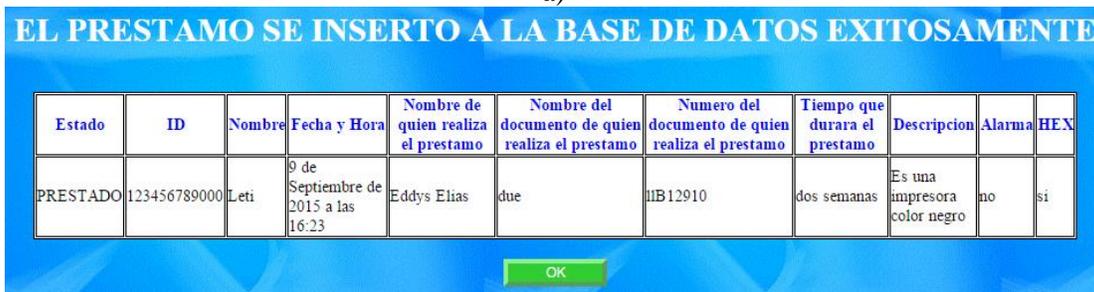


a)

Figura A.69: El sistema advierte al usuario que la información del elemento en la base de datos de alarmas ha sido eliminada para poder ser registrada en la base de datos de préstamos, a) Interface gráfica situada en el módulo de control.



a)

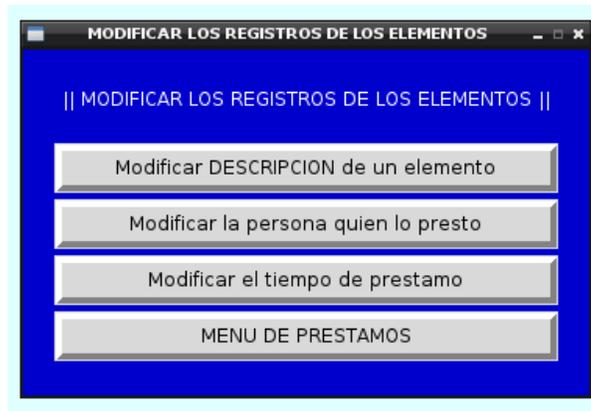


b)

Figura A.70: Resultados de la función al registrar un elemento en la base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Por último el sistema pregunta al usuario si desea ingresar un nuevo elemento, si el usuario acepta, el sistema reiniciara la función, pero si no acepta el sistema terminara la función y regresara al menú de préstamos.

⇒ **MODIFICAR REGISTROS DE ELEMENTOS EN LA TABLA.**



a)

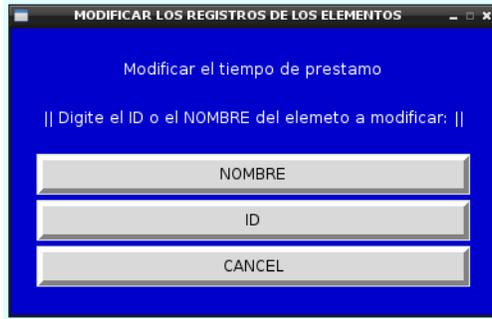
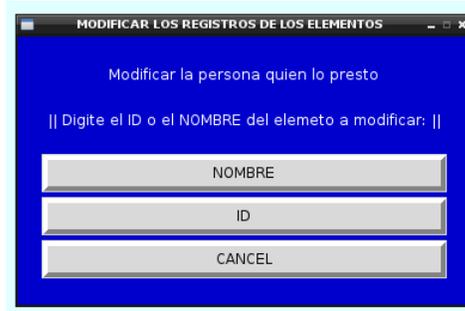
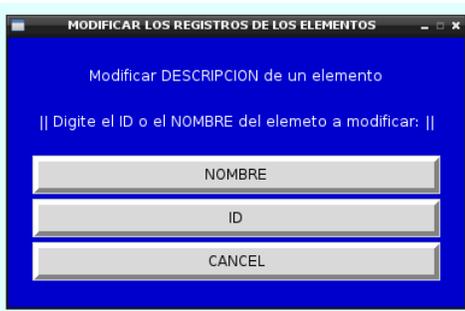


b)

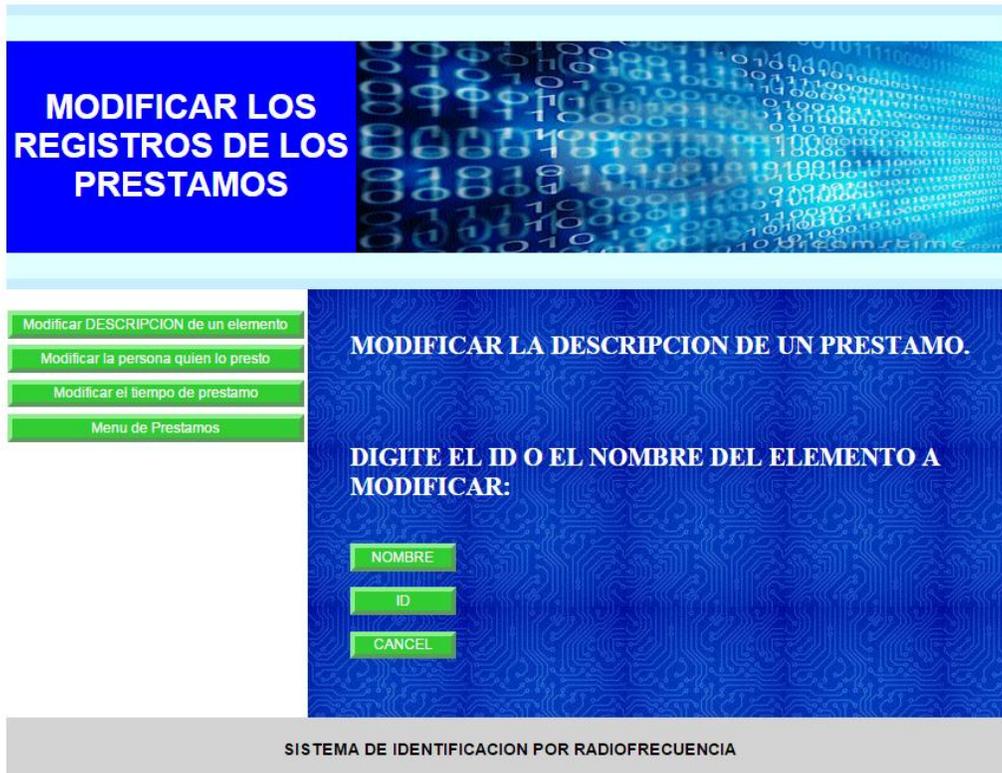
Figura A.71: Menú de la función modificar los registros de los elementos de la base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema es capaz de modificar la información de los registros de la base de datos de préstamos, esto le otorga al usuario un mejor control con la información de los elementos a controlar, también le otorga la oportunidad de modificar o cambiar a la persona que realizara el préstamos cuando el usuario diga ser conveniente; el sistema puede modificar la descripción del elemento, así como también los datos personales de quien realizo el préstamos y el tiempo de que durara el préstamo.

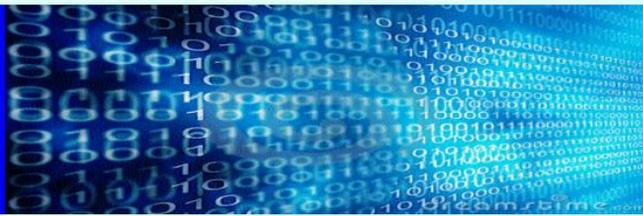
Al elegir el usuario el registro que se modificara de la información del elemento registrado en la base de datos de préstamos, el sistema le preguntara la forma que buscara la información del elemento, estas dos formas son por el ID o por el NOMBRE del elemento registrado, y si existe dicho elemento en la base de datos de préstamos el sistema prosigue a preguntarle al usuario sobre la nueva DESCRIPCION del elemento, información de la persona que realizo el préstamo o el tiempo que durara el préstamo.



a)



MODIFICAR LOS REGISTROS DE LOS PRESTAMOS



Modificar DESCRIPCION de un elemento

Modificar la persona quien lo presto

Modificar el tiempo de prestamo

Menu de Prestamos

MODIFICAR LOS DATOS DE LA PERSONA QUE REALIZO EL PRESTAMO.

DIGITE EL ID O EL NOMBRE DEL ELEMENTO A MODIFICAR:

NOMBRE

ID

CANCEL

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

MODIFICAR LOS REGISTROS DE LOS PRESTAMOS



Modificar DESCRIPCION de un elemento

Modificar la persona quien lo presto

Modificar el tiempo de prestamo

Menu de Prestamos

MODIFICAR EL TIEMPO QUE DURARA EL PRESTAMO.

DIGITE EL ID O EL NOMBRE DEL ELEMENTO A MODIFICAR:

NOMBRE

ID

CANCEL

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

b)

Figura A.72: Opciones de búsqueda del elemento para modificar sus registros en la base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



a)
b)

Figura A.73: Formas en que el sistema busca la información específica del préstamo a modificar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



a)
b)

Figura A.74: Ventanas para ingresar ya sea la nueva DESCRIPCION del elemento o el nuevo tiempo que durara el préstamo, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

MODIFICAR LOS REGISTROS DE LOS PRESTAMOS



- Modificar DESCRIPCION de un elemento
- Modificar la persona quien lo presto
- Modificar el tiempo de prestamo
- Menu de Prestamos

MODIFICAR DESCRIPCION DE UN PRESTAMO.

INGRESE EL NOMBRE DEL ELEMENTO A MODIFICAR:

INGRESE LA NUEVA DESCRIPCION DEL PRESTAMO A MODIFICAR:

Modificar

CANCEL

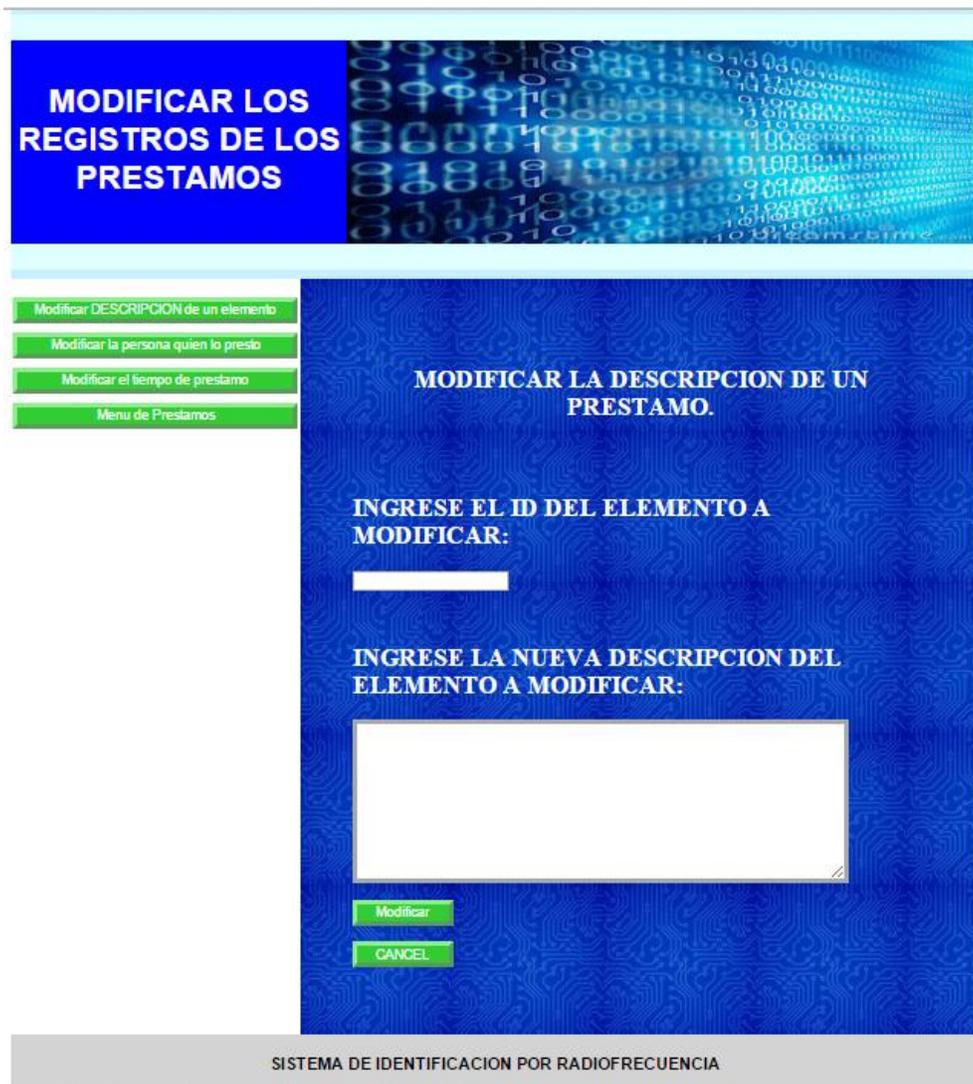
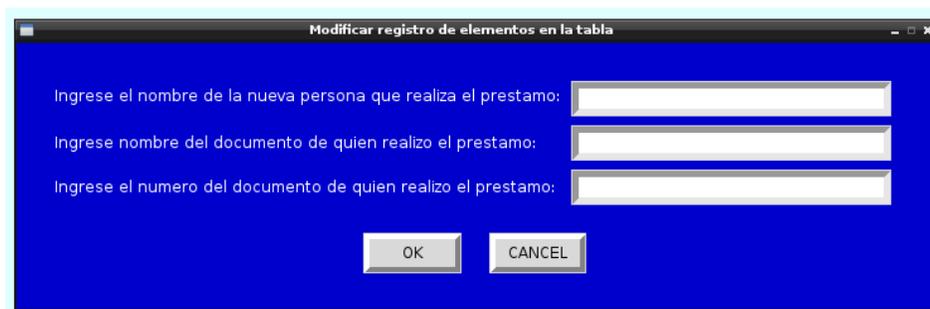


Figura A.75: Ejemplo de las ventanas que utiliza la interfaz WEB para modificar los registros de los elementos en la base de datos de préstamos, se presenta las ventanas para modificar la descripción del elemento y ser elegido en la base de datos ya sea por su nombre actual o por su ID.



a)

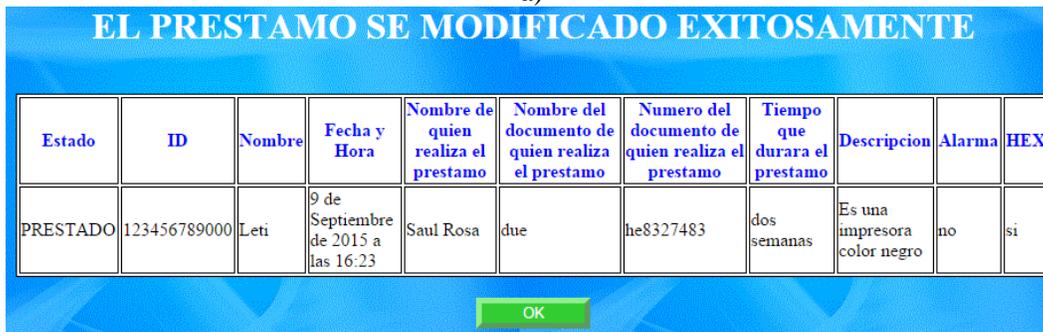
b)

Figura A.76: Ventana para ingresar la nueva información de la persona que realizara o continuara el préstamo del elemento, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El usuario debe tener en cuenta que los datos para la modificación de la información de la persona que realizo el préstamo, todos son obligatorios, y si el usuario cometió un error al ingresar la información o no la ingresó el sistema advertirá de su error. Luego que el usuario haya ingresado el nuevo registro la función lo modificara, si se efectuó correctamente el sistema mostrara dicha información al usuario.



a)

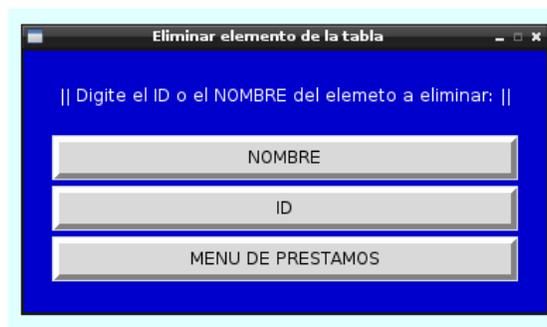


b)

Figura A.77: Resultados de la modificación de los registros de un préstamo en específico (en este caso se modificó los datos personales de la persona quien realizo el préstamo), a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Al presionar el botón OK el sistema reiniciara la función dándole oportunidad al usuario de modificar otro registro de un préstamo específico.

⇒ **ELIMINAR ELEMENTOS DE LA TABLA.**



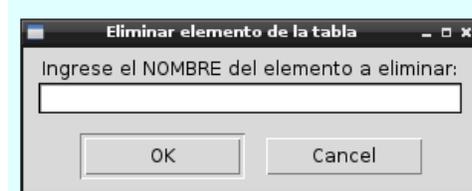
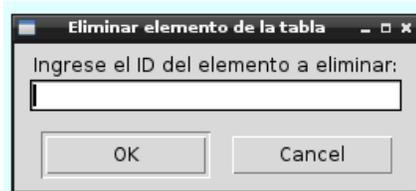
a)



b)

Figura A.78: Opciones de búsqueda para poder eliminar la información de un préstamo específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función es capaz de eliminar la información de un préstamo específico, si el usuario elimina la información de un elemento de la base de datos de préstamos cambiara el estado de la información del mismo en la base de datos principal como PRESENTE; el sistema le otorga al usuario las opciones de búsqueda del elemento teniendo así un control específico de la información que se eliminara.



a)



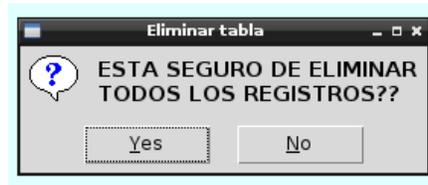
b)

Figura A.79: Formas en que el sistema busca la información específica del préstamo a eliminar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Cuando el usuario haya ingresado ya sea el ID o el NOMBRE de elemento a eliminar el sistema lo buscara y advertirá al usuario si dicho elemento se encuentra registrado, si es así se eliminara toda su información específica del elemento.

⇒ **ELIMINAR TABLA.**

La función es capaz de eliminar en su totalidad la información de los elementos que se encuentran en la base de datos de préstamos, el sistema le pregunta al usuario si verdaderamente desea eliminar toda la información de la base de datos de préstamo, si acepta el sistema eliminara o borrará toda la información de la base de datos, pero si no acepta el sistema regresara al menú de préstamos.



a)

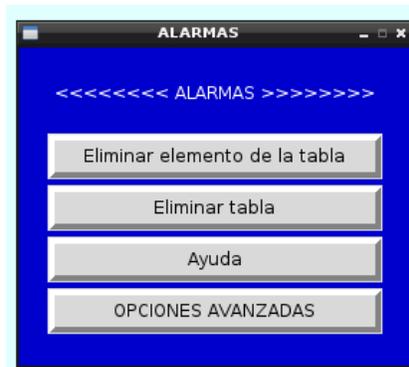


b)

Figura A.80: Forma en que el sistema advierte al usuario de que se eliminara toda la información de la base de datos de préstamos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema advierte al usuario de que eliminara toda la información de la base de datos de préstamos, ya que no se desea que lo realice en forma accidental, perdiendo así un registro adecuado de los elementos que no se encuentran en el rango de distancia del escaneo del módulo RFID pero que se encuentran con permiso de préstamo, eliminando alarmas innecesarias.

- **ALARMAS.**



a)

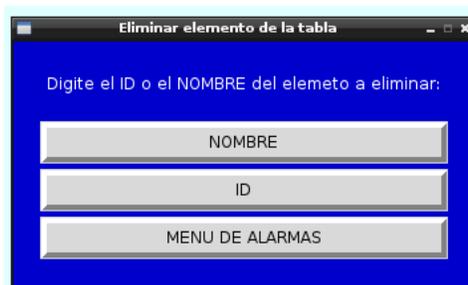


b)

Figura A.81: Menú de las funciones de ALARMAS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema posee funciones de alarmas, esto es muy importante ya que así el usuario tendrá un sistema más eficiente ya que conocerá cuales elementos están fuera del rango de escaneo y no tiene permiso, generando así una alarma advirtiéndole que elemento ha sido hurtado; a continuación se describirán el funcionamiento de los procesos que abarca la función de alarmas:

⇒ **ELIMINAR ELEMENTOS DE LA TABLA.**



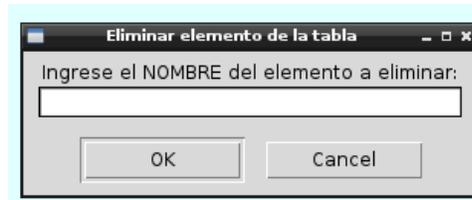
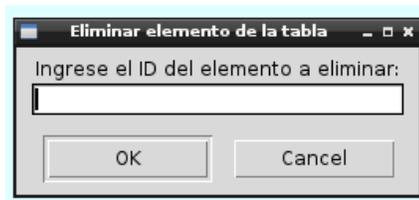
a)



b)

Figura A.82: Opciones de búsqueda para poder eliminar la información de una alarma específica, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función es capaz de eliminar la información de una alarma específica, si el usuario elimina la información de un elemento de la base de datos de alarmas el sistema cambiara el estado de la información del mismo en la base de datos principal como PRESENTE; se le otorga al usuario las opciones de búsqueda del elemento teniendo así un control específico de la información que se eliminara.



a)



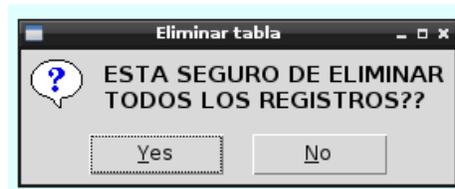
b)

Figura A.83: Formas en que el sistema busca la información específica de la alarma a eliminar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Cuando el usuario haya ingresado ya sea el ID o el NOMBRE de elemento a eliminar el sistema buscara en la base de datos de alarmas y advertirá al usuario si dicho elemento se encuentra registrado, si es así el sistema eliminara toda su información.

⇒ **ELIMINAR TABLA.**

La función es capaz de eliminar en su totalidad la información de los elementos que se encuentran en la base de datos de alarmas, el sistema le pregunta al usuario si verdaderamente desea eliminar toda la información de la base de datos de alarmas, si acepta el sistema eliminara o borrará toda la información de la base de datos de alarmas, pero si no acepta el sistema regresara al menú de alarmas.



a)

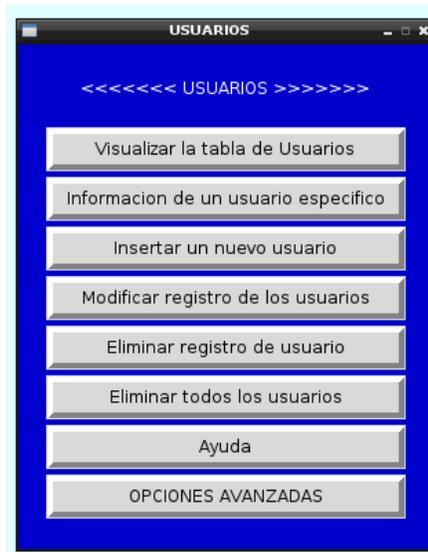


b)

Figura A.84: Forma en que el sistema advierte al usuario de que se eliminara toda la información de la base de datos de alarmas, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema advierte al usuario de que eliminara toda la información de la base de datos de alarmas, ya que no se desea que lo realice en forma accidental, perdiendo así un registro adecuado de los elementos que no se encuentran en el rango de distancia del escaneo del módulo RFID generando así una alarma.

- **USUARIOS.**



a)

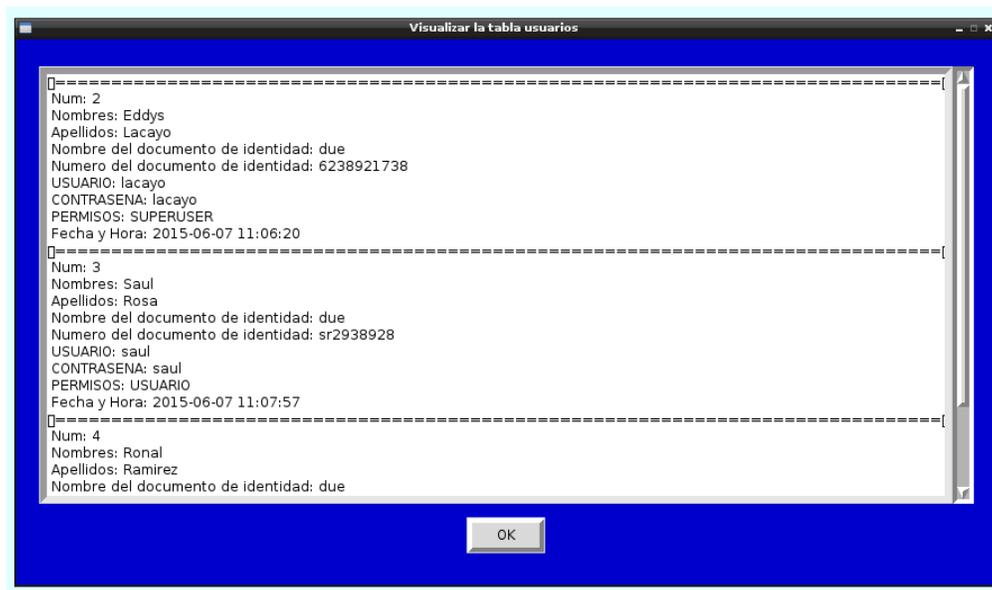


b)

Figura A.85: Menú de las sub funciones de la función de USUARIOS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema posee dos tipos de usuarios, el primero es un usuario natural el cual tiene los permisos básicos del sistema; el segundo tipo de usuario es un súper usuario el cual tiene todos los permisos del sistema, dándole el control total al sistema. A continuación se describirán las sub funciones de la función usuarios:

⇒ **VISUALIZAR LA TABLA DE USUARIOS.**



a)

VISUALIZAR BASE DE DATOS DE USUARIOS							
Nombre	Apellido	Nombre del documento de identidad del usuario	Numero del documento de identidad del usuario	USUARIO	PASSWORD	PERMISOS	Fecha y Hora
Eddys	Lacayo	due	6238921738	lacayo	lacayo	SUPERUSER	7 de Junio de 2015 a las 11:06
Saul	Rosa	due	sr2938928	saul	saul	USUARIO	7 de Junio de 2015 a las 11:07
Ronal	Ramirez	due	jdisdia	ronal	ronal	USUARIO	8 de Junio de 2015 a las 10:50

b)

Figura A.86: Visualización de los datos de todos los USUARIOS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

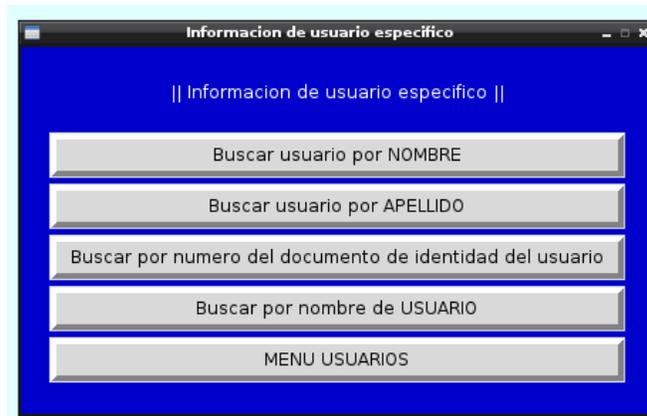
En esta sub función el sistema le muestra al usuario todos los datos relacionados a los usuarios registrados en el sistema, a los dos tipos de usuarios, como se observa en la figura A.86, dicha información de los usuarios es:

- **Nombres:** Son los nombres del usuario.
- **Apellidos:** Son los apellidos del usuario.
- **Nombre del documento de identidad:** El nombre de un documento que corrobore la identidad del usuario.
- **Número del documento de identidad:** El numero de un documento que corrobore la identidad del usuario.
- **USUARIO:** Es un nombre que se le otorga al usuario para que el sistema lo identifique al ingresar al sistema.
- **PASSWORD:** Es una palabra clave que el usuario otorgara para identificarlo con el sistema al querer ingresar a los menús.

- **Permisos:** Son los permisos que el sistema le otorga al usuario para el acceso a diferentes partes del mismo, los permisos son:
 - **USUARIO:** El permiso le otorga al usuario acceso al sistema, pero le permite la interacción solamente con las funciones que abarca el menú principal.
 - **SUPERUSER:** El permiso le otorga al usuario acceso al sistema, le permite el acceso a todo el sistema, le permite la iteración sobre las funciones del menú principal y al de opciones avanzadas.
- **Fecha y Hora:** Le informa la hora y la fecha en que el usuario fue ingresado a la base de datos de usuarios del sistema.

Por último si el usuario ase click en el botón OK, el sistema le pregunta si quiere visualizar de nuevo la base de datos de los usuarios, si acepta el sistema vuelve a mostrar toda la información de la base de datos de usuarios, pero si no acepta el sistema regresara al menú de sub funciones de la función de usuarios.

⇒ **INFORMACION DE UN USUARIO ESPECÍFICO.**



a)



b)

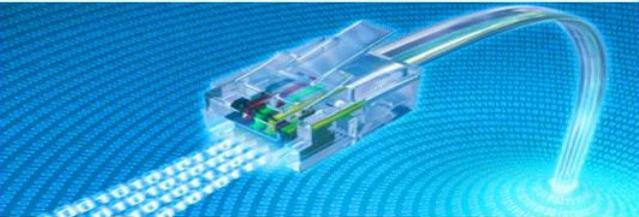
Figura A.87: Menú de la función información de un usuario específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función es capaz de mostrar la información específica de un usuario registrado en la base de datos, pide elegir la forma en que se buscara la información, la primera forma es por el nombre del usuario, esto es por el nombre del usuario según el documento de identificación del mismo; la segunda forma es por el apellido del usuario a buscar, igualmente como la anterior manera es el apellido obtenido por algún documento de identificación del usuario; la tercera forma es con el NUMERO DEL DOCUMENTO DE IDENTIDAD del usuario registrado en el sistema, este documento, puede ser cualquier documento que identifique a la persona; advierte al usuario si no ha ingresado la información necesaria; y la cuarta forma es por medio del nombre de USUARIO, se debe tener el cuidado de que el nombre de USUARIO es diferente del nombre del usuario, el primero es el nombre que se le otorga al usuario para que el sistema lo identifique y el segundo es el nombre del usuario según un documento de identidad del mismo; el sistema le pide ingresar el nombre de USUARIO a buscar; si el sistema encuentra la información será mostrada y de igual manera si no es encontrada en la base de datos, el sistema advertirá que la información del usuario no ha sido encontrado en el sistema.



a)

INFORMACION DE USUARIO ESPECIFICO



- Buscar usuario por NOMBRE
- Buscar usuario por APELLIDO
- Buscar por numero del documento de identidad del usuario
- Buscar por nombre de USUARIO
- Menu Usuarios

BUSCAR POR NOMBRE DEL USUARIO

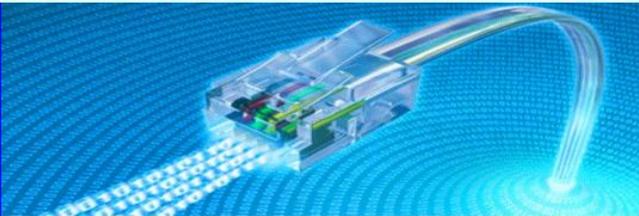
INGRESE EL NOMBRE DEL USUARIO:

BUSCAR

CANCEL

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

INFORMACION DE USUARIO ESPECIFICO



- Buscar usuario por NOMBRE
- Buscar usuario por APELLIDO
- Buscar por numero del documento de identidad del usuario
- Buscar por nombre de USUARIO
- Menu Usuarios

BUSCAR USUARIO POR APELLIDO

INGRESE EL APELLIDO DEL USUARIO:

BUSCAR

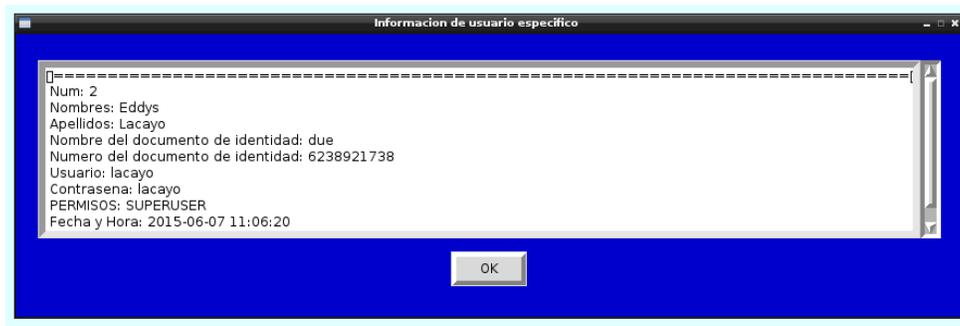
CANCEL

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

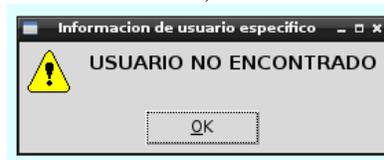


b)

Figura A.88: Formas es que el sistema busca la información específica de un usuario, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



a)



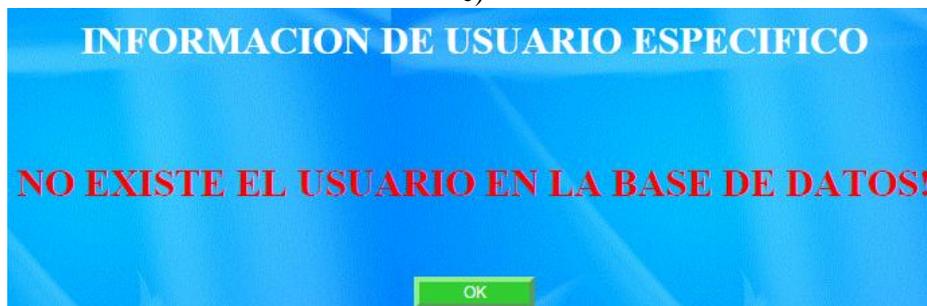
b)

INFORMACION DE USUARIO ESPECIFICO

Nombre	Apellido	Nombre del documento de identidad del usuario	Numero del documento de identidad del usuario	USUARIO	PASSWORD	PERMISOS	Fecha y Hora
Eddys	Lacayo	due	6238921738	lacayo	lacayo	SUPERUSER	7 de Junio de 2015 a las 11:06

OK

c)



d)

Figura A.89: Resultado de la búsqueda de la información de un usuario específico, a) La información del usuario fue encontrada, interface gráfica situada en el módulo de control, b) La información del usuario fue no encontrada, interface gráfica situada en el módulo de control, c) La información del usuario fue encontrada, Interface gráfica WEB, d) La información del usuario fue no encontrada, Interface gráfica WEB.

Por ultimo al presionar el botón OK, de la ventana que se observa en la figura A.89, la sub función regresa al menú de la función USUARIOS y le pregunta la forma en que el sistema buscara la información del usuario específico.

⇒ **INSERTAR UN NUEVO USUARIO.**

Insertar un nuevo usuario

Ingrese el nombre del usuario:

Ingrese el apellido del usuario:

Ingrese el nombre del documento de identidad del usuario:

Ingrese numero del documento del usuario:

USUARIO:

PASSWORD:

PERMISOS: USUARIO SUPERUSUARIO

a)

USUARIOS

- Visualizar la tabla de Usuarios
- Información de un usuario específico
- Insertar un nuevo usuario**
- Modificar registro de los usuarios
- Eliminar registro de usuario
- Eliminar todos los usuarios
- OPCIONES AVANZADAS

INSERTAR UN USUARIO EN LA BASE DE DATOS

Ingrese el nombre del usuario:

Ingrese el apellido del usuario:

Ingrese el nombre del documento de identidad del usuario:

Ingrese numero del documento del usuario:

USUARIO:

PASSWORD:

PERMISOS:

- USUARIO
- SUPERUSUARIO

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

b)

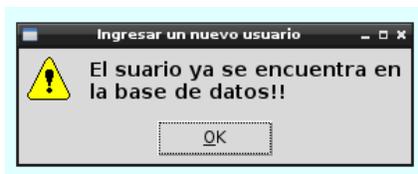
Figura A.90: Ventana para ingresar los datos necesarios para un nuevo usuario en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función despliega una ventana para que se ingrese los datos necesarios para registrar un nuevo usuario en la base de datos de usuarios, estos datos son:

- **Nombres:** Son los nombres del nuevo usuario.
- **Apellidos:** Son los apellidos del nuevo usuario.
- **Nombre del documento de identidad:** El nombre de un documento que corrobore la identidad del usuario.
- **Número del documento de identidad:** El numero de un documento que corrobore la identidad del usuario.
- **USUARIO:** Es un nombre que se le otorga al usuario para que el sistema lo identifique al ingresar al sistema.
- **PASSWORD:** Es una palabra clave que el usuario otorgara para identificarlo con el sistema al querer ingresar a los menús.
- **Permisos:** Son los permisos que el sistema le otorga al usuario para el acceso a diferentes partes del mismo, los permisos son:
 - **USUARIO:** El permiso le otorga al usuario acceso al sistema, pero le permite la interacción solamente con las funciones que abarca el menú principal.
 - **SUPERUSER:** El permiso le otorga al usuario acceso al sistema, le permite el acceso a todo el sistema, le permite la iteración sobre las funciones del menú principal y al de opciones avanzadas.

Debe tener en mente que los datos de los nombres y los apellidos del usuario según un documento de identidad, nombre y número del documento de identidad de la persona, y permisos, son datos obligatorios.

El sistema advertirá si el usuario que se quiera registrar en la base de datos, ya haya sido ingresado anteriormente, mostrando una ventana de advertencia informando de que el usuario ya ha sido registrado en la base de datos.



a)

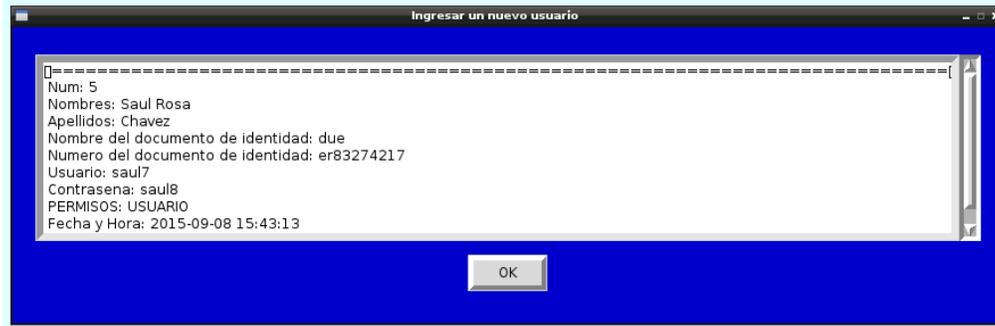
EL USUARIO YA SE ENCUENTRA EN LA BASE DE DATOS!

Nombre	Apellido	Nombre del documento de identidad del usuario	Numero del documento de identidad del usuario	USUARIO	PASSWORD	PERMISOS	Fecha y Hora
Eddys	Lacayo	due	6238921738	lacayo	lacayo	SUPERUSER	7 de Junio de 2015 a las 11:06

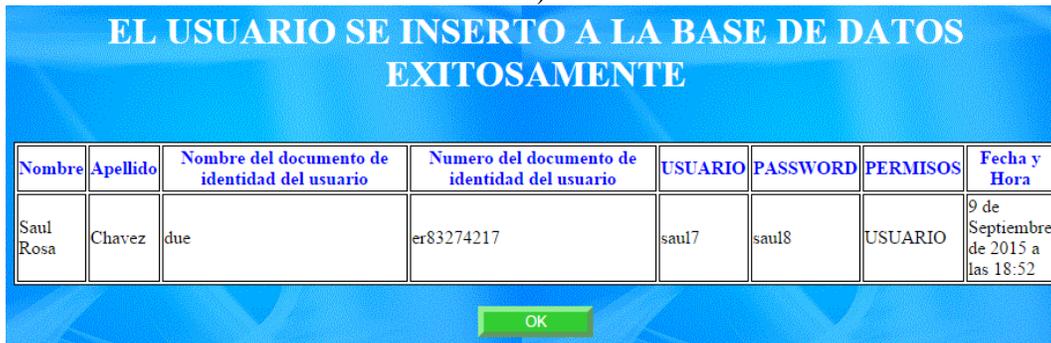
OK

b)

Figura A.91: Ventanas de advertencia de que la informando del usuario ya se encuentra registrado en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



a)



b)

Figura A.92: Resultados de la función al registrar un usuario en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Por último el sistema pregunta si desea ingresar un nuevo usuario, si acepta, el sistema reiniciara la función, pero si no acepta terminara la función y regresara al menú de usuarios.

⇒ **MODIFICAR REGISTROS DE LOS USUARIOS.**



a)



b)

Figura A.93: Menú de la función modificar los registros de los usuarios de la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema es capaz de modificar la información de los registros de la base de datos de usuarios, le otorga la oportunidad de modificar o cambiar los datos personales del usuario registrado, así como también el nombre de USUARIO y el PASSWORD del mismo.

Al elegir el registro que se modificara de la información del usuario en la base de datos, el sistema buscara la información del usuario con el nombre de USUARIO, y si existe prosigue a preguntarle al usuario sobre los nuevos datos personales del usuario, el nuevo nombre de USUARIO o el nuevo PASSWORD.

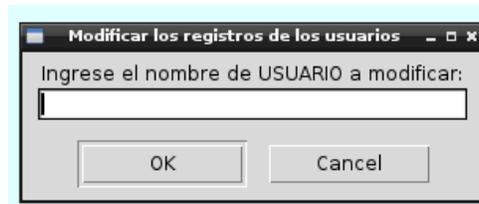


Figura A.94: Forma en que el sistema busca la información específica del usuario a modificar, Interface gráfica situada en el módulo de control.

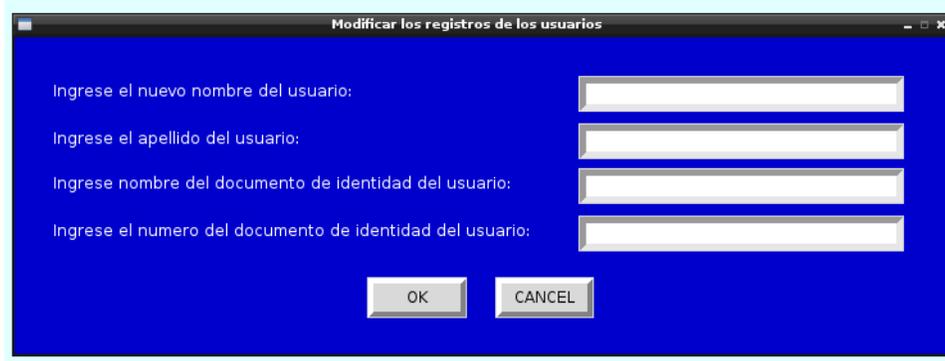


a)



b)

Figura A.95: Ventanas para ingresar ya sea el nuevo nombre de USUARIO o el nuevo PASSWORD, a) Interface gráfica situada en el módulo de control, b) Ejemplo de las ventanas para modificar la información específica de un usuario, Interface gráfica WEB.



a)

MODIFICAR LOS REGISTROS DE LOS USUARIOS

Modificar los datos personales de un usuario
 Modificar el nombre del USUARIO
 Modificar el PASSWORD
 Menu Usuarios

MODIFICAR LOS DATOS PERSONALES DEL USUARIO.

INGRESE EL NOMBRE DE USUARIO A MODIFICAR:

INGRESE EL NUEVO NOMBRE DE LA PERSONA DEL USUARIO:

INGRESE EL NUEVO APELLIDO DE LA PERSONA DEL USUARIO:

INGRESE EL NUEVO NOMBRE DEL DOCUMENTO DE IDENTIDAD DEL USUARIO:

INGRESE EL NUEVO NUMERO DEL DOCUMENTO DE IDENTIDAD DEL USUARIO:

Modificar
 CANCEL

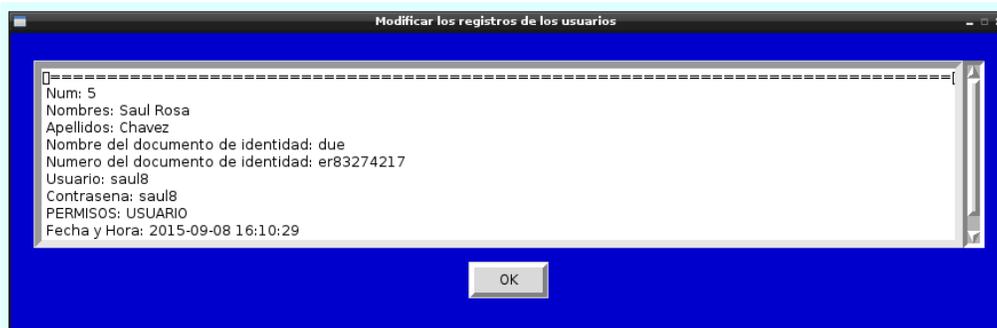
SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

b)

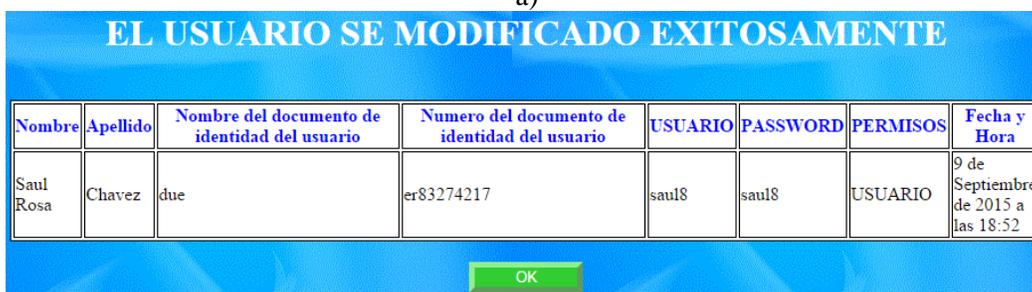
Figura A.96: Ventana para ingresar la nueva información del usuario, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El usuario debe tener en cuenta que los datos para la modificación de la información, todos son obligatorios. Lugo que se haya ingresado la nueva información la función buscara la información

del usuario y la modificara, si la modificación del registro del usuario en específico se efectuó correctamente el sistema mostrara dicha información del usuario modificado.



a)

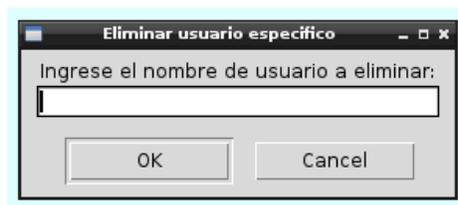


b)

Figura A.97: Resultados de la modificación de los registros de un usuario en específico (en este caso se modificó el nombre de USUARIO), a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Al presionar el botón OK el sistema reiniciara la función dándole oportunidad al usuario de modificar otro registro de un usuario específico.

⇒ **ELIMINAR REGISTROS DE USUARIOS.**



a)



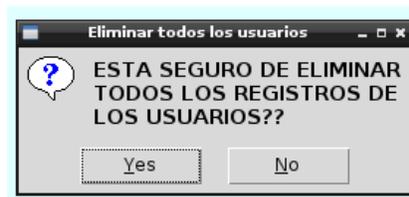
b)

Figura A.98: Opción de búsqueda para poder eliminar la información de un usuario específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función es capaz de eliminar la información de un usuario específico, así se tiene más control en el documentado de la información de los usuarios registrados en la base de datos. Cuando se haya ingresado el nombre de USUARIO a eliminar el sistema buscará en la base de datos de usuarios y advertirá si dicho elemento se encuentra registrado, si es así el sistema eliminará toda su información específica e informará si el proceso de eliminación se efectuó correctamente.

⇒ **ELIMINAR TODOS LOS USUARIOS.**

La función es capaz de eliminar en su totalidad la información de los usuarios que se encuentran en la base de datos, el sistema le pregunta al usuario si verdaderamente desea eliminar toda la información de la base de datos de usuarios, si se acepta el sistema eliminará o borrará toda la información de la base de datos, pero si no acepta el sistema regresará al menú de usuarios.



a)



b)

Figura A.99: Forma en que el sistema advierte de que se eliminara toda la información de la base de datos de usuarios, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema advierte de que eliminara toda la información de la base de datos de usuarios, ya que no se desea que lo realice en forma accidental, perdiendo así un registro adecuado de los usuarios que poseen permisos para ingresar al sistema e interactuar con las diferentes funciones del mismo.

NOTA: Se debe tener presente que si se elimina en su totalidad toda la información correspondiente a los usuarios registrados en el sistema, los usuarios tendrán "una puerta trasera" para poder ingresar, esto es un usuario con permisos especiales para poder ingresar a todas las funciones y procesos del sistema este es:

USUARIO: **sysrfid**
 PASSWORD: **sysrfid**

Se debe tener cuidado con el uso de este usuario especial o de esta puerta trasera ya que pase lo que pase no se puede eliminar de los registros de usuarios, y así evitar el ingreso al sistema de personas no deseadas.

- **USUARIOS SMS.**



a)

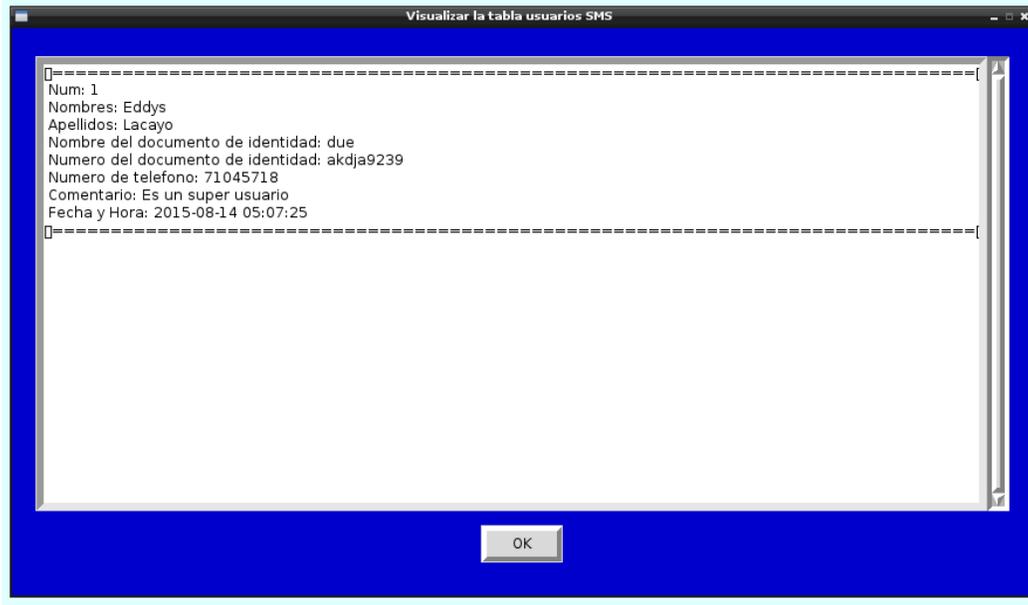


b)

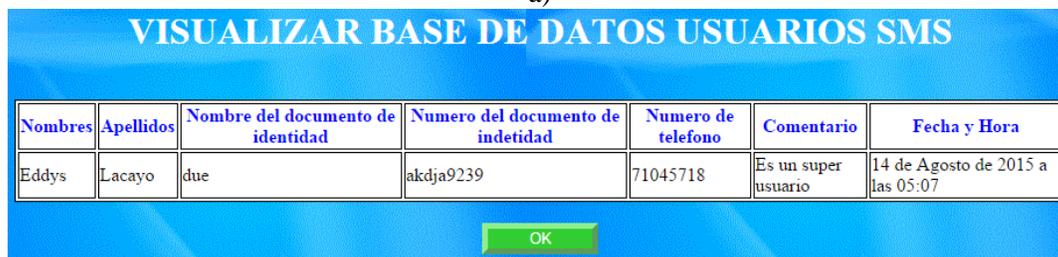
Figura A.100: Menú de las sub funciones de la función de USUARIOS SMS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Los usuarios SMS son los usuarios a los cuales se les enviara mensajes de texto hacia sus teléfonos celulares, estos mensajes de texto son mensajes de alerta, informando al usuario sobre cuales elementos no se detectaron cuando el sistema realizo un escaneo, estos mensajes contienen los ID de los elementos no se detectaron, así como también contiene el ID de los elementos detectados cuando se activa la función de ESCANEEO, y no tiene permiso de salida, el usuario debe tener presente que los usuarios registrados en el sistema y los usuarios SMS no son los mismos. A continuación se describirán las sub funciones de la función usuarios:

⇒ **VISUALIZAR LA TABLA DE USUARIOS SMS.**



a)



b)

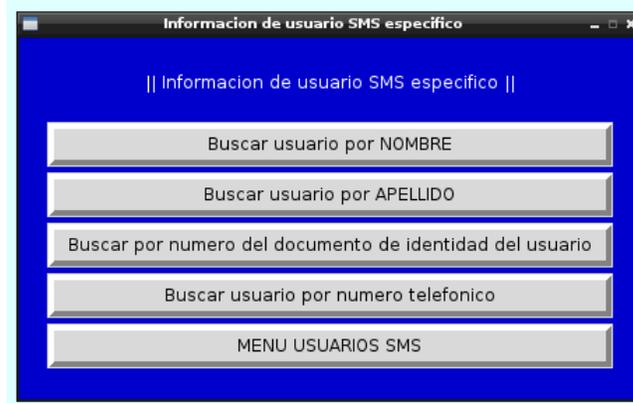
Figura A.101: Visualización de los datos de todos los USUARIOS SMS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

En esta sub función el sistema le muestra al usuario todos los datos relacionados a los usuarios SMS registrados en el sistema, como se observa en la figura A.101, dicha información de los usuarios es:

- **Nombres:** Son los nombres del usuario.
- **Apellidos:** Son los apellidos del usuario.
- **Nombre del documento de identidad:** El nombre de un documento que corrobore la identidad del usuario.
- **Número del documento de identidad:** El numero de un documento que corrobore la identidad del usuario.
- **Número de teléfono:** Es el número de teléfono al cual el sistema GSM le enviara los mensajes de texto.
- **Comentario:** Es información extra que el usuario del sistema haya puesto por su relevancia para la identificación del usuario SMS.
- **Fecha y Hora:** Le informa la hora y la fecha en que el usuario fue ingresado a la base de datos de usuarios SMS del sistema.

Por último si el usuario ase click en el botón OK, el sistema le pregunta si quiere visualizar de nuevo la base de datos de los usuarios SMS registrados en el sistema, si acepta el sistema vuelve a mostrar toda la información, pero si no acepta regresara al menú de sub funciones de la función de usuarios SMS.

⇒ **INFORMACION ESPECÍFICA DE UN USUARIO SMS.**



a)



b)

Figura A.102: Menú de la función información de un usuario SMS específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función es capaz de mostrar la información específica de un usuario SMS registrado en la base de datos, la función pide elegir la forma en que el sistema buscara la información, la primera forma es por el nombre del usuario SMS, esto es por el nombre del usuario según el documento de identificación del mismo; la segunda forma es por el apellido del usuario SMS a buscar, igualmente como la anterior manera es el apellido obtenido por algún documento de identificación del usuario; la tercera forma es con el NUMERO DEL DOCUMENTO DE IDENTIDAD del usuario SMS registrado en el sistema, este documento, puede ser cualquier documento que identifique a la persona, pero se debe buscar la información del usuario SMS con el número del documento con que

fue registrada; el sistema advierte si no ha ingresado la información necesaria; y la cuarta forma es por medio del número de teléfono al cual se le enviara los mensajes de texto, el número de teléfono debe de ser en dígitos en formato decimal; el sistema le pide al usuario ingresar el número de teléfono del usuario a buscar; si el sistema encuentra la información del usuario SMS registrado en la base de datos dicha información será mostrada.



a)





INFORMACION DE USUARIO SMS ESPECIFICO

- Buscar usuario por NOMBRE
- Buscar usuario por APELLIDO
- Buscar por numero del documento de identidad del usuario
- Buscar usuario por numero telefonico
- Menu Usuarios SMS

BUSCAR USUARIO SMS POR APELLIDO

INGRESE EL APELLIDO DEL USUARIO:

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA



INFORMACION DE USUARIO SMS ESPECIFICO

- Buscar usuario por NOMBRE
- Buscar usuario por APELLIDO
- Buscar por numero del documento de identidad del usuario
- Buscar usuario por numero telefonico
- Menu Usuarios SMS

BUSCAR USUARIO SMS POR EL NUMERO DE UN DOCUMENTO DE IDENTIDAD

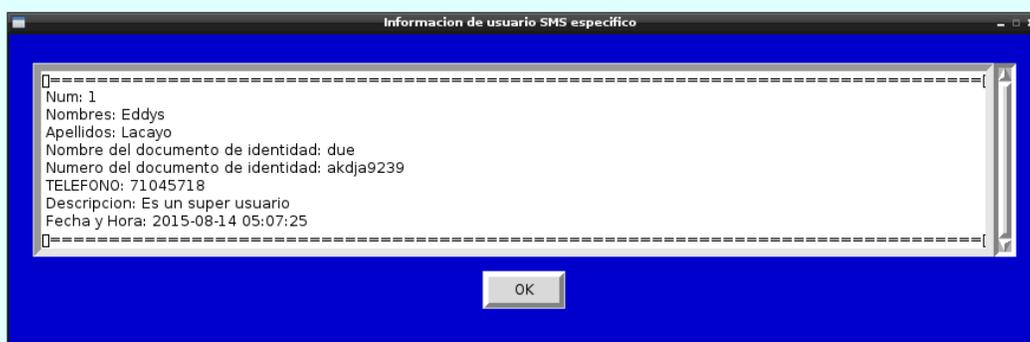
INGRESE EL NOMBRE DE LA PERSONA QUE REALIZO EL PRESTAMO:

SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA

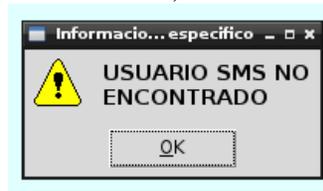


b)

Figura A.103: Formas es que el sistema busca la información específica de un usuario SMS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



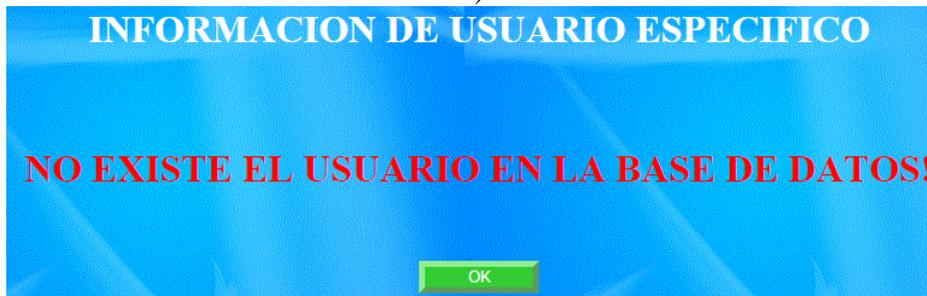
a)



b)

INFORMACION DE USUARIO SMS ESPECIFICO						
Nombres	Apellidos	Nombre del documento de identidad	Numero del documento de identidad	Numero de telefono	Comentario	Fecha y Hora
Eddys	Lacayo	due	akdja9239	71045718	Es un super usuario	14 de Agosto de 2015 a las 05:07

c)



d)

Figura A.104: Resultado de la búsqueda de la información de un usuario SMS específico, a) La información del usuario SMS fue encontrada, interface gráfica situada en el módulo de control, b) La información del usuario SMS fue no encontrada, interface gráfica situada en el módulo de control, c) La información del usuario SMS fue encontrada, Interface gráfica WEB, d) La información del usuario SMS fue no encontrada, Interface gráfica WEB.

Por ultimo al presionar el botón OK, de la ventana que se observa en la figura A.104, la sub función regresa al menú de la función USUARIOS SMS y le pregunta la forma en que el sistema buscara la información del usuario SMS específico.

⇒ **INSERTAR UN NUEVO USUARIO SMS.**

a)

The image shows a web application interface for managing SMS users. At the top left, there is a blue header with the text 'USUARIOS SMS'. Below this, a sidebar contains several green buttons with white text: 'Visualizar la tabla de Usuarios SMS', 'Información de un usuario SMS específico', 'Insertar un nuevo usuario SMS', 'Modificar registro de los usuarios SMS', 'Eliminar registro de usuario SMS', 'Eliminar todos los usuarios SMS', and 'OPCIONES AVANZADAS'. The main content area has a blue background with a circuit-like pattern and is titled 'INSERTAR UN USUARIO SMS EN LA BASE DE DATOS'. It contains the following form fields: 'Ingrese el nombre del usuario:', 'Ingrese el apellido del usuario:', 'Ingrese el nombre del documento de identidad del usuario:', 'Ingrese numero del documento del usuario:', 'Ingrese el numero telefonico:', and 'Ingrese comentario:'. At the bottom of the form are two buttons: 'Insertar' and 'CANCEL'. The footer of the page reads 'SISTEMA DE IDENTIFICACION POR RADIOFRECUENCIA'.

b)

Figura A. 105: Ventana para ingresar los datos necesarios para un nuevo usuario SMS en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

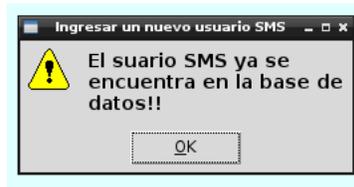
La función despliega una ventana para que se ingrese los datos necesarios para registrar un nuevo usuario SMS en la base de datos de usuarios SMS, estos datos son:

- **Nombres:** Son los nombres del nuevo usuario SMS.
- **Apellidos:** Son los apellidos del nuevo usuario SMS.
- **Nombre del documento de identidad:** El nombre de un documento que corrobore la identidad del usuario.
- **Número del documento de identidad:** El numero de un documento que corrobore la identidad del usuario.
- **Número de teléfono:** Es el número de teléfono al cual el sistema GSM le enviara los mensajes de texto.
- **Comentario:** Es información extra que el usuario del sistema haya puesto por su relevancia para la identificación del usuario SMS.

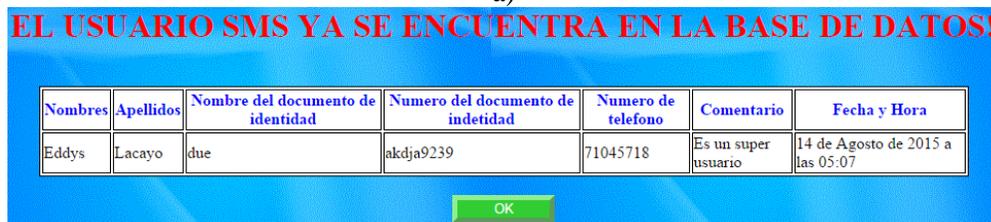
Debe tener en mente que los datos de los nombres y los apellidos del usuario según un documento de identidad, nombre y número del documento de identidad de la persona y el número de teléfono

del usuario, son datos obligatorios, esto quiere decir que el sistema no procederá a ingresar los datos si el usuario no los ingresa, el sistema lo advertirá del error.

El sistema advertirá si el usuario SMS que se quiera registrar en la base de datos, ya haya sido ingresado anteriormente, mostrando una ventana de advertencia informando de que el usuario SMS ya ha sido registrado en la base de datos.

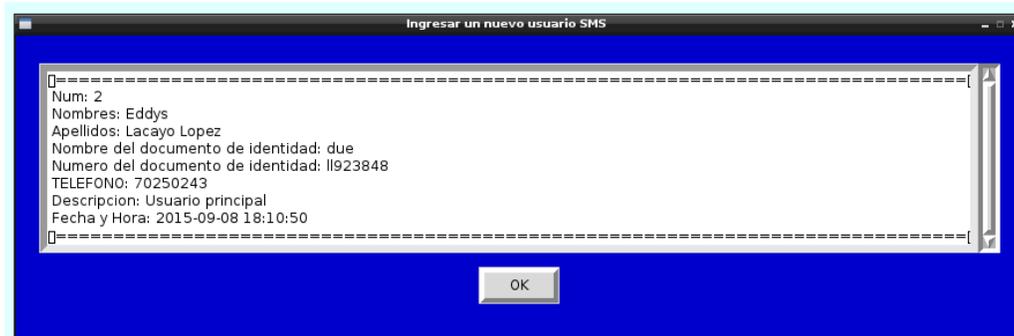


a)

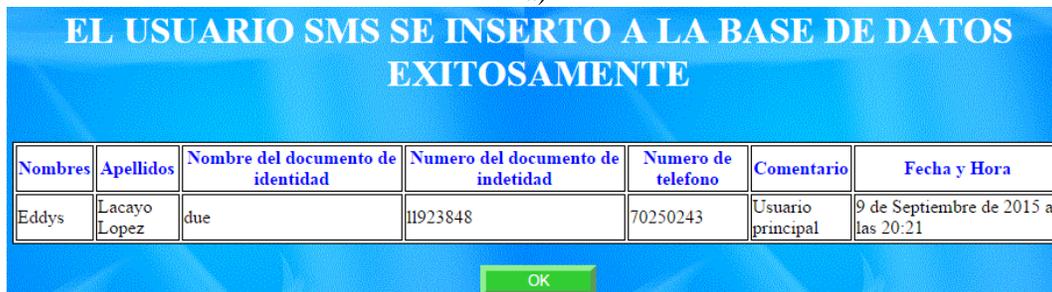


b)

Figura A.106: Ventanas de advertencia de que la informando del usuario SMS ya se encuentra registrado en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.



a)



b)

Figura A.107: Resultados de la función al registrar un usuario SMS en la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Por último el sistema pregunta si desea ingresar un nuevo usuario SMS, si se acepta, el sistema reiniciara la función, pero si el usuario no acepta el sistema terminara la función y regresara al menú de usuarios SMS.

⇒ **MODIFICAR REGISTROS DE LOS USUARIOS SMS.**



a)



b)

Figura A.108: Menú de la función modificar los registros de los usuarios SMS de la base de datos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema es capaz de modificar la información de los registros de la base de datos de usuarios SMS, le otorga la oportunidad de modificar o cambiar los datos personales del usuario registrado, así como también del número de teléfono y del comentario del mismo.

Al elegir el registro que se modificara de la información del usuario SMS registrado en la base de datos, el sistema buscara la información con el número de teléfono, el sistema interroga a la base de datos de usuarios SMS sobre la existencia de dicha información, y si existe el sistema prosigue a preguntar sobre los nuevos datos personales del usuario SMS, el nuevo número de teléfono o el nuevo comentario.

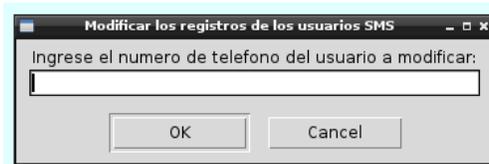
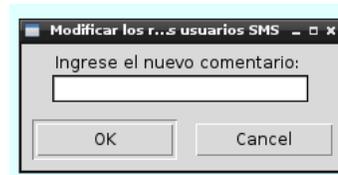
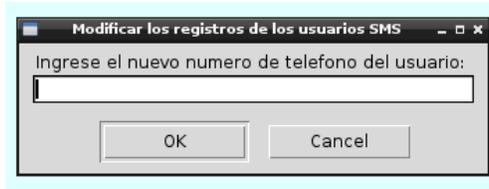


Figura A.109: Forma en que el sistema busca la información específica del usuario SMS a modificar, Interface gráfica situada en el módulo de control.



a)



b)

Figura A.110: Ventanas para ingresar ya sea el nuevo número de teléfono y el nuevo comentario, a) Interface gráfica situada en el módulo de control, b) Ejemplo de una ventana para la modificación de la información de un usuario sms específico Interface gráfica WEB.

Modificar los registros de los usuarios SMS

Ingrese el nuevo nombre del usuario:

Ingrese el apellido del usuario:

Ingrese nombre del documento de identidad del usuario:

Ingrese el numero del documento de identidad del usuario:

a)

MODIFICAR LOS REGISTROS DE LOS USUARIOS SMS

Modificar los datos personales de un usuario SMS
 Modificar el número de teléfono
 Modificar el comentario
 Menu Usuarios SMS

MODIFICAR LOS DATOS PERSONALES DEL USUARIO SMS.

INGRESE EL NUMERO DE TELEFONO DEL USUARIO A MODIFICAR:

INGRESE EL NUEVO NOMBRE DE LA PERSONA DEL USUARIO:

INGRESE EL NUEVO APELLIDO DE LA PERSONA DEL USUARIO:

INGRESE EL NUEVO NOMBRE DEL DOCUMENTO DE IDENTIDAD DEL USUARIO:

INGRESE EL NUEVO NUMERO DEL DOCUMENTO DE IDENTIDAD DEL USUARIO:

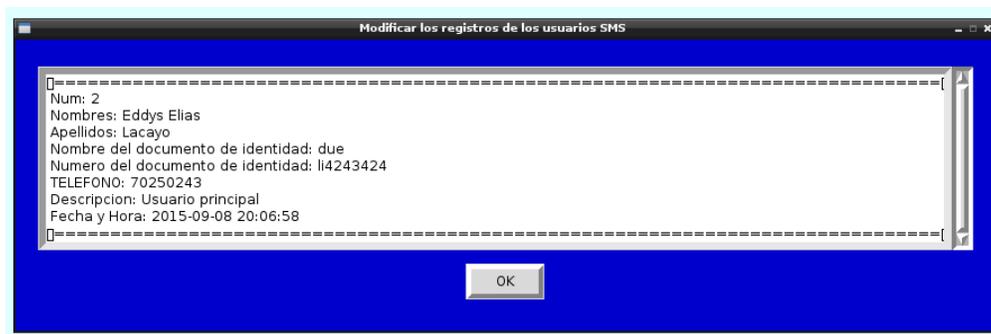
Modificar
 CANCEL

b)

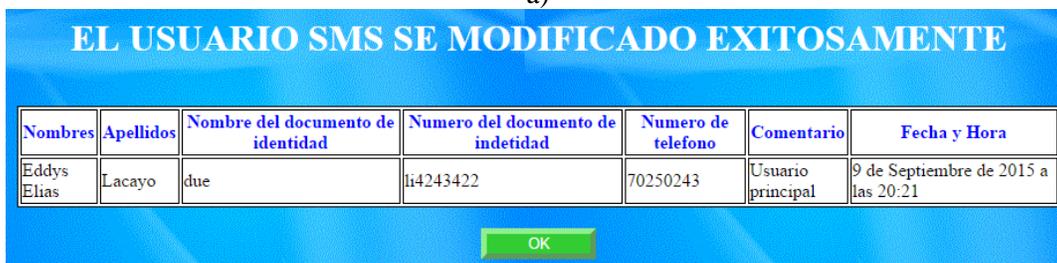
Figura A.111: Ventana para ingresar la nueva información del usuario SMS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Debe tener en cuenta que los datos para la modificación de la información del usuario SMS, como los datos personales del usuario y el número de teléfono del usuario, son obligatorios; los datos del nuevo comentario no son obligatorios esto quiere decir que si el usuario no ingresa ninguna información, borrara los datos del comentario anterior y lo reemplazara con una celda en blanco. Lugo que se haya ingresado la nueva información la función buscara los registros del usuario SMS

y la modificara, si la modificación del registro del usuario en específico se efectuó correctamente el sistema mostrara dicha información del usuario SMS modificado.



a)

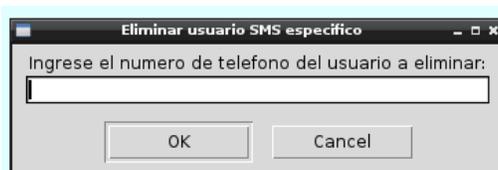


b)

Figura A.112: Resultados de la modificación de los registros de un usuario SMS en específico (en este caso se modificó el nombre de USUARIO), a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Al presionar el botón OK el sistema reiniciara la función dándole oportunidad al usuario de modificar otro registro de un usuario SMS específico.

⇒ **ELIMINAR USUARIOS SMS.**



a)



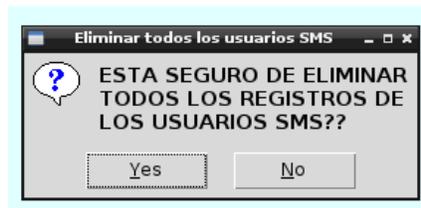
b)

Figura A.113: Opción de búsqueda para poder eliminar la información de un usuario SMS específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Cuando se haya ingresado el número de teléfono del usuario a eliminar el sistema buscará en la base de datos de usuarios SMS y advertirá si dicho elemento se encuentra registrado, si es así el sistema eliminará toda su información específica del usuario SMS e informará al usuario si el proceso de eliminación se efectuó correctamente.

⇒ **ELIMINAR TODOS LOS USUARIOS SMS.**

La función es capaz de eliminar en su totalidad la información de los usuarios SMS que se encuentran en la base de datos, el sistema le pregunta al usuario si verdaderamente desea eliminar toda la información de la base de datos, si se acepta se eliminará o borrará toda la información de la base de datos, pero si no acepta el sistema regresará al menú de usuarios.



a)



b)

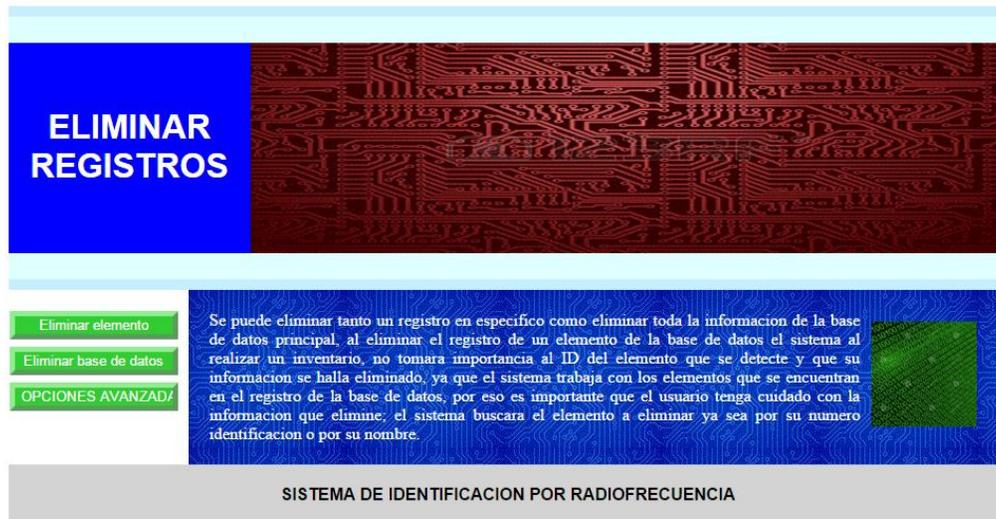
Figura A.114: Forma en que el sistema advierte de que se eliminara toda la información de la base de datos de usuarios SMS, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema advierte de que eliminara toda la información de la base de datos de usuarios SMS, ya que no se desea que lo realice en forma accidental, perdiendo así un registro adecuado de los usuarios a los cuales se les enviara los mensajes de alarma sobre los elementos no detectados en el escaneo y que no tiene permisos de préstamos, así también de los elementos detectados cuando se activa la función de escaneo, y no tiene permiso de salida.

- **ELIMINAR REGISTROS.**



a)

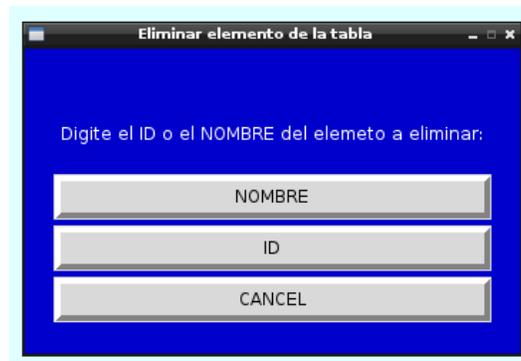


b)

Figura A.115: Menú de la función sobre eliminar la información de los registros de los elementos, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El sistema es capaz de eliminar los registros de los elementos en la base de datos principal, el usuario debe de tener cuidado al ejecutar las subfunciones de la función eliminar registros, ya que al eliminar el registro de un elemento también eliminara toda la información relacionada a dicho elemento, esto quiere decir que si se elimina el registro de un elemento específico y se encuentra también información del mismo elemento en la base de datos de préstamos, esos registros también se eliminara, y así sucederá si existe registros también en la base de datos de alarmas, por eso el usuario debe tener cuidado al ejecutar dichas funciones.

⇒ **ELIMINAR ELEMENTOS.**



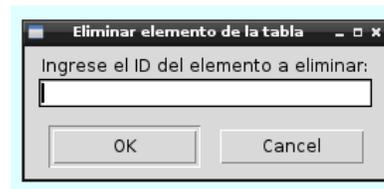
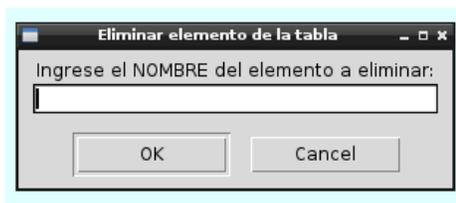
a)



b)

Figura A.116: Opciones de búsqueda para poder eliminar la información de un elemento específico, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

La función es capaz de eliminar la información de un elemento específico, si el usuario elimina la información de un elemento de la base de datos principal el sistema también eliminara todos los registros en la base de datos de préstamos y/o alarmas si se encuentra información de dicho elemento.



a)



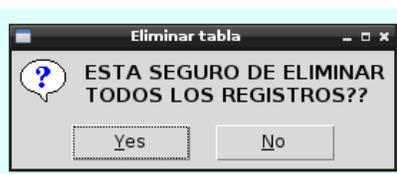
b)

Figura A.117: Formas en que el sistema busca la información específica del elemento a eliminar, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

Cuando el usuario haya ingresado ya sea el ID o el NOMBRE de elemento a eliminar el sistema buscara en la base de datos principal y advertirá al usuario si dicho elemento se encuentra registrado, si es así se eliminara toda su información específica del elemento en la base de datos principal y/o en las bases de datos de préstamos e alarmas, si se encuentran registros de dicho elemento en esas bases de datos, e informara si el proceso de eliminación se efectuó correctamente.

⇒ **ELIMINAR TABLA.**

La función es capaz de eliminar en su totalidad la información de los elementos que se encuentran en la base de datos principal, el sistema le pregunta al usuario si verdaderamente desea eliminar toda la información, si acepta el sistema eliminara o borrara toda la información de la base de datos principal, pero si no acepta el sistema regresara al menú de opciones avanzadas.



a)



b)

Figura A.118: Forma en que el sistema advierte al usuario de que se eliminara toda la información de la base de datos principal, a) Interface gráfica situada en el módulo de control, b) Interface gráfica WEB.

El usuario debe tener en mente que si se ejecuta dicha sub función no solo se eliminara todos los registros en la base de datos principal sino también en la base de datos de préstamos e alarmas. El sistema advierte al usuario de que eliminara toda la información de la base de datos principal, ya que no se desea que lo realice en forma accidental, perdiendo así un registro adecuado de los

elementos que se encuentran en el rango de distancia del escaneo del módulo RFID, y también de los elementos que tienen permiso de préstamos y han generado alguna alarma.

ANEXO B: CÓDIGOS DE CONFIGURACIÓN DE LOS DIFERENTES ELEMENTOS QUE COMPONEN EL SISTEMA.

- 🚩 Código de configuración cargado en el ARDUINO MEGA 2560, para el control del módulo RFID AS3992:

```
/*CODIGO ARDUINO MEGA - PROGRAMA RFID*/

//FUNCIONES
void coman_powerantena();
void com_inventario();
void com_invrssi();
void coman_selectag();
void coman_readtag();
void coman_passwtag();
void coman_freq();
void selec_func();
void num_power2();
void num_power3();
void trans_hex2();
void visualizar();
void completar(int var22);
void trans_hex();
void hex_impar();
void num_freq();

//VARIABLES
char command_activa[ ]={0x18, 0x03, 0xFF}; //ACTIVAR ANTENA
char command_desact[ ]={0x18, 0x03, 0x00}; //DESACTIVAR ANTENA
char command_scantag[]={0x31, 0x03, 0x01}; //INVENTARIO
unsigned char SCMD[] = {0x43,0x03,0x01}; //INVENTARIO CON RSSI
int cont1, var1, lon1, numtag, x, y, hx, hy, hz, cm=0, mx=0, pp, tim=0;
int pp1, var12, var22, p1, p2, x2, y1, lim1, lim2, col;
byte comand_rfid1[40], coman_leer1[40], z1, z12;
unsigned char RSSI, I, Q;
unsigned char Rcard[12];
long power1[5], power2[5], Freq2[7], Freqx[7], Freq[4];

void setup( ){
  Serial.begin(9600); //Velocidad de comunicacion entre la PC y el arduino MEGA
  Serial1.begin(9600); //Velocidad de comunicacion entre el modulo RFID y el arduino MEGA
}

void loop( ){

  if(x2==0){
    Serial.println("\n\r\n\r");
    Serial.println(" #####");
    Serial.println(" ##### <<<<<< SISTEMA RFID >>>>>> #####");
    Serial.println(" #####");
    x2++; //Se aumenta en uno la variable x2 para pasar al siguiente proceso
  }
}
```

```

//Menu principal
if(x2==1){
  Serial.println("\n\r"); //Imprime en pantalla de la PC dos ENTER
  Serial.println("\n\r <1> POTENCIA DE LA ANTENA");
  //delay(1000); //Espera 1 segundo
  Serial.println("\n\r <2> INVENTARIO");
  //delay(1000); //Espera 1 segundo
  Serial.println("\n\r <3> SELECCIONAR ETIQUETA ESPECIFICA");
  //delay(1000); //Espera 1 segundo
  Serial.println("\n\r <4> ESCRIBIR ETIQUETA");
  //delay(1000); //Espera 1 segundo
  Serial.println("\n\r <5> CAMBIAR CONTRASENA DE LA ETIQUETA");
  //delay(1000); //Espera 1 segundo
  Serial.println("\n\r <6> CAMBIAR FRECUENCIA DE OPERACION");
  //Serial.println("7. SALIR.");
  x2++; //La variable aumenta en uno para avanzar con el siguiente proceso
}

if(x2==2){
  while(Serial.available()){
    var12=Serial.read(); //Lee el caracter ingresado por la PC y lo guarda en la variable var12
    Serial.print("\n\r \n\r >>Opcion: ");
    Serial.write(var12); //Imprime en la pantalla de la PC el valor que contiene la variable var12
    if(var12=='1'){
      mx=1; //Se otorga el valor de 1 a la variable mx para que active a la funcion llamada y se desactiva al cambiar el
      valor de la variable mx
      var1='1'; //Se otorga el valor del caracter 1 a la variable var1 para comenzar el proceso de la funcion llamada, la
      cual pide este caracter para comenzar
      coman_powerantena(); //Se llama a la funcion para que el usuario pueda activar, desactivar y modificar la potencia
      de la antena
      x2=0; //Se iguala a cero para que imprima de nuevo en la pantalla de la PC el menu principal
    }
    if(var12=='2'){
      x2=3; //Se iguala a tres para que pueda imprimir en pantalla de la PC el menu secundario
    }
    if(var12=='3'){
      mx=1; //Se otorga el valor de 1 a la variable mx para que active a la funcion llamada y se desactiva al cambiar el
      valor de la variable mx
      var1='3'; //Se otorga el valor del caracter 3 a la variable var1 para comenzar el proceso de la funcion llamada, la
      cual pide este caracter para comenzar
      coman_selectag(); //Se llama a la funcion para que el usuario pueda seleccionar una tag especifica
      x2=0; //Se iguala a cero para que imprima de nuevo en la pantalla de la PC el menu principal
    }
    if(var12=='4'){
      mx=1; //Se otorga el valor de 1 a la variable mx para que active a la funcion llamada y se desactiva al cambiar el
      valor de la variable mx
      var1='4'; //Se otorga el valor del caracter 4 a la variable var1 para comenzar el proceso de la funcion llamada, la
      cual pide este caracter para comenzar
      coman_readtag(); //Se llama a la funcion para que el usuario pueda modificar el ID de la etiqueta espesifica
      x2=0; //Se iguala a cero para que imprima de nuevo en la pantalla de la PC el menu principal
    }
    if(var12=='5'){
      mx=1; //Se otorga el valor de 1 a la variable mx para que active a la funcion llamada y se desactiva al cambiar el
      valor de la variable mx
      var1='5'; //Se otorga el valor del caracter 5 a la variable var1 para comenzar el proceso de la funcion llamada, la
      cual pide este caracter para comenzar
      coman_passwtag(); //Se llama a la funcion para que el usuario pueda modificar el password de la etiqueta
      x2=0; //Se iguala a cero para que imprima de nuevo en la pantalla de la PC el menu principal
    }
    if(var12=='6'){

```



```

//delay(1000); //Espera 1 segundo
Serial.println("\n\r 3. Desactivar antena");
//delay(1000); //Espera 1 segundo
Serial.println("\n\r 4. Salir");
x++; //Aumenta en uno la variable x, para pasar al siguiente proceso
}

if(Serial.available()){

  if(x==3){
    while(Serial.available()){
      var22=Serial.read(); //Lee el caracter ingresado por la PC y lo guarda en la variable var22
      //Serial.print("\n\r\n\rOpcion: ");
      //Serial.write(var22); //Imprime en la pantalla de la PC el valor que contiene la variable var22
      if(var22=='1'){
        comand_rfid1[2]=0xFF; //Se otorga el valor de 0xFF en la posicion 2 del arreglo comand_rfid1, este valor da la
orden de desactivar la antena
        x=12; //Se le otorga el valor de 12 a la variable para que salte los procesos y llegar asta el proceso que se
encarga de escribir el comando para cambiar la potencia de la antena
        p2++; //Aumenta en uno la variable p2 para aumenta en uno la posicione en el arreglo comand_rfid1
        cont1=1; //Se otorga el valor de 1 a la variable para acivar el proceso de mostrar el mensaje de activar la antena
        }
      if(var22=='2'){
        x++; //Aumenta en uno la variable x para poder pasar al siguiente porceso
        cont1=2; //Se otorga el valor de 2 a la variable para acivar el proceso de mostrar el mensaje de activar la antena
        }
      if(var22=='3'){
        comand_rfid1[2]=0x00; //Se otorga el valor de 0x00 en la posicion 2 del arreglo comand_rfid1, este valor da la
orden de activar la antena
        x=12; //Se le otorga el valor de 12 a la variable para que salte los procesos y llegar asta el proceso que se
encarga de escribir el comando para cambiar la potencia de la antena
        p2++; //Aumenta en uno la variable p2 para aumenta en uno la posicione en el arreglo comand_rfid1
        cont1=3; //Se otorga el valor de 1 a la variable para acivar el proceso de mostrar el mensaje de activar la antena
        }
      if(var22=='4'){
        var1=0; //Se le otorga el valor de 0 a la variable var1 para que finalice la funcion y salir de ella
        mx=2; //Se le otorga el valor de 2 a la variable mx para salir completamente de la funcion actual
        x=100; //Se otorga el valor de 100 a la variable x4 para que no continue con los siguientes procesos y termine
todo
        }
      if(var22!='1' && var22!='2' && var22!='3' && var22!='4'){
        Serial.println("\n\r \n\r OPCION ERRONEA");
        Serial.println(" Elija una opcion en el menu.\n\r");
        x=2; //Se otorga el valor de 2 a la variable para que pueda regresar a mostrar el menu
        }
      }
    } //Fin de if x=3

    if(x==4){
      Serial.println("\n\r \n\r \n\r Ingrese la potencia de la antena: ");
      Serial.println(" (La potencia de la antena debe de estar en el rango de 1 a 254)");
      lim1=0; //Se inicializa la variable en 0, para inicializar el conteo de la cantidad de los caracteres que componen la
potencia de la antena los cuales no deben sobrepasar la cantidad de 3
      pp=0; //Se inicializa la variable en 0, para inicializar la posicion en la cual se ingresara los caracteres en el arreglo
power2 el cual contenbra la potencia de la antena
      pp1=0; //Se inicializa la variable en 0, para inicializar la posicion en el arreglo power2 y cambiar los caracteres
ingresados anteriormente en el mismo arreglo por sus equivalentes en decimales
      x++; //La variable aumenta en uno para avanzar con el siguiente proceso
    }

    if(Serial.available()){

```

```

while(x<10){
  //Lee el valor de la potencia de la antena ingresada por la PC
  while(Serial.available()){
    delay(250); //Espera 1/4 de segundo
    power2[pp] = Serial.read(); //Lee caracteres escritos en la PC y las guarda en el arreglo power2, en el cual se
guardara el caracter en el arreglo esta definida por pp
    var22=power2[pp]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo power2,
en la posicion definida por la variable pp
    Serial.write(power2[pp]); //Se imprime en la pantalla de la PC el caracter guardado por el arreglo power2 en la
posicion definida por la variable pp
    lim1++; //Aumenta en uno la variable lim1 para tener la cantidad de caracteres que componen la potencia de la
antena y tener el control de que sean 3 caracteres

    //Este proceso se activa cuando se ingresa un caracter que no es un numero decimal, y si se activa ya no se
efectuan mas procesos y se termina toda la funcion
    if(power2[pp]!='0' && power2[pp]!='1' && power2[pp]!='2' && power2[pp]!='3' && power2[pp]!='4'){
      if(power2[pp]!='5' && power2[pp]!='6' && power2[pp]!='7' && power2[pp]!='8' && power2[pp]!='9'){
        Serial.println("\n\r\n\r\n\r ERROR: La potencia de la antena debe de ser en numeros decimales.");
        while(Serial.available()){
          delay(250); //Espera 1/4 de segundo
          power2[pp] = Serial.read(); //Se guarda los caracteres ingresados por la PC en el arreglo power2 en la
posicion dada por la variable pp, para que ya no sean mostrado en pantalla de la PC
          pp++; //Aumenta en uno la variable para avanzar en la pocision del arreglo power2
          lim1++; //Aumenta en uno la variable para contar la cantidad de caracteres ingresados por la PC
        }
        x=20; //Se otorga el valor de 20 a la variable x para que ya no prosiga con los de mas procesos y termine toda
la funcion
        var1='1'; //Se otorga el valor de 7 a la variable var1 para que ya no prosiga con los de mas procesos y termine
toda la funcion
      }
    }
    pp++; //Aumenta en uno la variable pp para avanzar en la posicion del arreglo power2
  }
  x++; //Aumenta en uno la variable x para continuar con el siguiente proceso, asta que llegue al valr de 10
}

if(x==21){
  x=2; //Se otorga el valor de 2 a la variable x para que ya no prosiga con los de mas procesos y termine toda la
funcion
}

if(x==10){
  if(lim1>3){
    Serial.println("\n\r\n\r\n\r ERROR: La potencia de la antena debe de ser de tres digitos maximo.");
    var1='1'; //Se otorga el valor de 7 a la variable var1 para que ya no prosiga con los de mas procesos y termine
toda la funcion
    x=2; //Se otorga el valor de 2 a la variable x para que ya no prosiga con los de mas procesos y termine toda la
funcion
  }
}

while(x==10){
  if(power2[pp1]=='0'){
    if (pp1<pp){
      power1[pp1]=0; //Se otorga el valor de 0 a la pocision que da la variable pp en el arreglo power1 cuando el
caracter ingresado por la PC es 0
      selec_func(); //Se llama a la funcion para escojer al proceso que convertira los caracteres que componen la
potencia de la antena en una sola variable
    }
  }
}

```

```

    }
    if(power2[pp1]=='1'){
        if (pp1<pp){
            power1[pp1]=1; //Se otorga el valor de 1 a la posicion que da la variable pp en el arreglo power1 cuando el
            caracter ingresado por la PC es 1
            selec_func(); //Se llama a la funcion para escojer al proceso que convertira los caracteres que componen la
            potencia de la antena en una sola variable
        }
    }
    if(power2[pp1]=='2'){
        if (pp1<pp){
            power1[pp1]=2; //Se otorga el valor de 2 a la posicion que da la variable pp en el arreglo power1 cuando el
            caracter ingresado por la PC es 2
            selec_func(); //Se llama a la funcion para escojer al proceso que convertira los caracteres que componen la
            potencia de la antena en una sola variable
        }
    }
    if(power2[pp1]=='3'){
        if (pp1<pp){
            power1[pp1]=3; //Se otorga el valor de 3 a la posicion que da la variable pp en el arreglo power1 cuando el
            caracter ingresado por la PC es 3
            selec_func(); //Se llama a la funcion para escojer al proceso que convertira los caracteres que componen la
            potencia de la antena en una sola variable
        }
    }
    if(power2[pp1]=='4'){
        if (pp1<pp){
            power1[pp1]=4; //Se otorga el valor de 4 a la posicion que da la variable pp en el arreglo power1 cuando el
            caracter ingresado por la PC es 4
            selec_func(); //Se llama a la funcion para escojer al proceso que convertira los caracteres que componen la
            potencia de la antena en una sola variable
        }
    }
    if(power2[pp1]=='5'){
        if (pp1<pp){
            power1[pp1]=5; //Se otorga el valor de 5 a la posicion que da la variable pp en el arreglo power1 cuando el
            caracter ingresado por la PC es 5
            selec_func(); //Se llama a la funcion para escojer al proceso que convertira los caracteres que componen la
            potencia de la antena en una sola variable
        }
    }
    if(power2[pp1]=='6'){
        if (pp1<pp){
            power1[pp1]=6; //Se otorga el valor de 6 a la posicion que da la variable pp en el arreglo power1 cuando el
            caracter ingresado por la PC es 6
            selec_func(); //Se llama a la funcion para escojer al proceso que convertira los caracteres que componen la
            potencia de la antena en una sola variable
        }
    }
    if(power2[pp1]=='7'){
        if (pp1<pp){
            power1[pp1]=7; //Se otorga el valor de 7 a la posicion que da la variable pp en el arreglo power1 cuando el
            caracter ingresado por la PC es 7
            selec_func(); //Se llama a la funcion para escojer al proceso que convertira los caracteres que componen la
            potencia de la antena en una sola variable
        }
    }
    if(power2[pp1]=='8'){
        if (pp1<pp){
            power1[pp1]=8; //Se otorga el valor de 8 a la posicion que da la variable pp en el arreglo power1 cuando el
            caracter ingresado por la PC es 8

```

```

    selec_func(); //Se llama a la funcion para escoger al proceso que convertira los caracteres que componen la
potencia de la antena en una sola variable
    }
}
if(power2[pp1]=='9'){
    if (pp1<pp){
        power1[pp1]=9; //Se otorga el valor de 9 a la pocision que da la variable pp en el arreglo power1 cuando el
caracter ingresado por la PC es 9
        selec_func(); //Se llama a la funcion para escoger al proceso que convertira los caracteres que componen la
potencia de la antena en una sola variable
        }
    }
    if(pp1==pp){
        x++;
    }
}

if(x==11){
    // Serial.println(power1[3]); //Imprime el valor de la posicion 3 del arreglo power1 donde se encuentra el valor de
la potencia de la antena
    if(power1[3]<1 || power1[3]>254 || power2[0]=='0'){
        Serial.println("\n\r \n\r \n\r ERROR: La potencia de la antena debe de estar en el rango de 1 a 254.");
        var1='1'; //Se otorga el valor de 7 a la variable var1 para que ya no prosiga con los de mas procesos y termine
toda la funcion
        x=2; //Se otorga el valor de 2 a la variable x4 para que ya no prosiga con los de mas procesos y termine toda la
funcion
    }
}

if(x==11){
    for(int t=0; t<pp; t++){
        power2[3]=power1[3]/16; //Se divide por 16 el valor que esta en el arreglo power1 en la posicion 3 (el cual es la
potencia de la antena en decimal), y se guarda en el arreglo power2 posicion 3
        power2[4]=power1[3]%16; //Se divide po 16 el valor que esta en el arreglo power1 en la posicion 3 (el cual es la
potencia de la antena en decimal), y se guarda en el arreglo power2 posicion 4 el resto de la division
        power1[3]=power2[3]; //Se otorga el valor del arreglo power2 posicion 4 al arreglo power1 posicion 3
        y++; //Aumenta en uno la variable y para que continue con la siguiente division
        coman_leer1[p1]=power2[4]; //Se otorga el valor del arreglo power2 posicion 4 al arreglo coman_leer1 posicion
que da la variable p1
        trans_hex2(); //Se llama a la funcion para tranformar los caracteres ingresados por la PC en byte de dos numeros
hex
        p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer3
    }
}

if(x==11){
    x++; //Aumenta en uno la variable x par poder continuar con el siguiente proceso
}

if(x==12){
    //Serial.println("\n\r"); //Imprime dos ENTER en la pantalla de la PC
    //Serial.println(p2); //Imprime en la pantalla de la PC el valor de la variable p2
    Serial1.write(comand_rfid1, p2); //Escribe todo el arreglo comand_rfid1 en el modulo RFID, el cual tiene la orden
de cambio de potencia de la antena
    //for(int i=0; i<(p2); i++){
        //Serial.print(comand_rfid1[i], HEX); //Imprime en la pantalla de la PC todo el arreglo comand_rfid1
        //}
    }
} //Fin de leer desde la PC

//Leyendo la respuesta del modulo RFID

```

```

if(Serial1.available()){

while (Serial1.available()){
delay(250); //Espera 1/4 de segundo
var22 = Serial1.read(); //Guarda lo que lee del modulo RFID en la variable var22
//Serial.print(var1, HEX); //Imprime lo que contiene la variable var1 en pantalla de la PC en formato HEX, la
respuesta del modulo RFID
}
if(cont1==1){
Serial.println("\n\r \n\r \n\r \n\r <<< ANTENA ACTIVADA >>>\n\r \n\r");
cont1=4; //se le otorga el valor de 4 a la variable cont1 para terminar el proceso de leer la respuesta del modulo
RFID
}
if(cont1==2){
Serial.println("\n\r \n\r \n\r \n\r <<< POTENCIA MODIFICADA >>>\n\r \n\r");
cont1=4; //se le otorga el valor de 4 a la variable cont1 para terminar el proceso de leer la respuesta del modulo
RFID
}
if(cont1==3){
Serial.println("\n\r \n\r \n\r \n\r <<< ANTENA DESACTIVADA >>>\n\r \n\r");
cont1=4; //se le otorga el valor de 4 a la variable cont1 para terminar el proceso de leer la respuesta del modulo
RFID
}
var1='1'; //Se otorga el caracter 1 a la variable var1 para que la funcion se reinicie
x=2; //Se otorga el valor de dos a la variable para que la funcion se reinicie
} //Fin de leer del modulo RFID
} //Fin de var=1
} //Fin del primer while
} //Fin de coman_powerantena()

/*Esta funcion realiza un escaneo con el modulo RFID a las etiquetas RFID y muestra en pantalla los ID de las
respectivas tag y esta compuesta de dos opciones,
las cuales son INVENTARIO SENCILLO (se activa cuando var1=1) y INVENTARIO CONTINUO (se activa si
var1=3), el INVENTARIO SENCILLO es un escaneo que se efectua
una vez y se vuelve a activar cuando el usuario ingrese el caracter 1, el INVENTARIO CONTINUO es un escaneo que
se efectua varias veces y se detiene asta que
el usuario ingrese el caracter w (las dos clases de inventario se detiene o se sale de ellos cuando se ingresa el caracter
w)*/
void com_inventario(){

if(var1=='3'){
Serial.println("\n\r \n\r \n\r \n\r"); //Imprime en la pantalla de la PC 5 ENTER
Serial.println(" ++++++<\/pre>

```



```

    var1 = Serial1.read(); //Guarda el primer byte del arreglo de la respuesta de la orden de inventario del modulo
RFID
    //Serial.print(var1, HEX); //Imprime en la pantalla de la PC el primer byte de respuesta del modulo RFID, en
formato HEX
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
    }
}

//LONGITUD DEL ID DE LA TAG
if(cont1<2){
    if (Serial1.available()){
        lon1 = Serial1.read(); //Guarda el segundo byte leído del modulo RFID en la variable lon1 el cual muestra la
longitud del ID de la tag
        //Serial.println("Longitud de ID tag: ");
        //Serial.println(lon1, HEX); //Escribe o imprime en la pantalla de la PC el segundo byte de respuesta del modulo
RFID en la pantalla de la PC guardado en la variable lon1 en formato HEX, el cual muestra la longitud de EPC o ID de la
tag
        cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
    }
}
//Serial.println(); //Imprime o escribe un ENTER en la pantalla de la PC

//NUMERO DE ETIQUETAS LEIDAS
if(cont1<3){
    if (Serial1.available()){
        numtag = Serial1.read(); //Guarda el tercer byte del arreglo de la respuesta del modulo RFID en la variable numtag
el cual muestra el numero de etiquetas leidas por el modulo RFID
        Serial.println("Numero de etiquetas leidas ");

        if(numtag<1){
            Serial.println("NO HAY ETIQUETAS");
            while(Serial1.available()){
                var1=Serial1.read(); //Guarda el tercer byte leído del modulo RFID en la variable var1 el cual muestra el
numero de tag leidas, si el byte es 00 el modulo no detecto ninguna etiqueta
                Serial.print(var1, HEX); //Imprime en la pantalla de la PC el tercer byte de respuesta del modulo RFID
guardado en la variable var1 en formato HEX, el cual muestra que el modulo no pudo leer ninguna etiqueta
            }
            cont1=9000; //Se le otorga el valor de 9000 a la variable cont1 para que no siga las demas ordenes y termine todo
            lon1=100; //Se le otorga el valor de 100 a la variable lon1 para que no siga los demas procesos relacionados con
esta variable
        }

        Serial.println(numtag); //Imprime en la pantalla de la PC el tercer byte de respuesta del modulo RFID guardado en
la variable numtag en formato HEX, la cual muestra el numero de tag leidas
        cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
    }
}
//Serial.println(); //Imprime un ENTER en la pantalla de la PC

while(cont1<6){
    if (Serial1.available()){
        var1 = Serial1.read(); //Guarda los bytes leídos del modulo RFID en la variable var1 los cuales muestran la
longitud de todo el arreglo de respuesta del modulo RFID y bytes resevados
        //Serial.print(var1, HEX); //Imprime en la pantalla de la PC, byte a byte, los bytes guardados en la variable var1
        cont1++; //Aumenta en uno la variable cont1 (para seguir con el siguiente proceso la variable tiene que tener el
valor de 6)
    }
}
Serial.println(); //Imprime un ENTER en la pantalla de la PC

//ID DE LA ETIQUETA

```

```

if(cont1<lon1){
  Serial.println("ID tag \r\n");
  Serial.print(x); //Imprime en la pantalla de la PC el valor de la variable x, la cual muestra el orden de los IDs de las
tags leidas, para una mejor visualizacion
  Serial.print(": ");
}
//Serial.println(); //Imprime o escribe en la pantalla de la PC un ENTER

//Imprime los bytes que componen el ID de las tag en formato HEX
if(hx==1){
  while(cont1<lon1){
    if(Serial1.available()){
      var1=Serial1.read(); //Lee, del modulo RFID, byte a byte los bytes que componen el ID de la tag y se guadan en
la variable var1
      completar(var1); //Se llama a la funcion completar mandandole la variable var1, y asi visualizar mejor los
numeros o bytes que tienen un cero a la izquierda, imprimiendo el cero
      Serial.print(var1, HEX); //Imprime en la pantalla de la PC los bytes que componen el ID de la tag, guardados en
la variable var1, en formato HEX
      cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso hasta que cont1 tenga el mismo
valor que lon1
    }
  }
  Serial.println(); //Imprime un ENTER en la pantalla de la PC
}

//Imprime los bytes que componen el ID de las tag en formato ascii
if(hx==2){
  while(cont1<lon1){
    if(Serial1.available()){
      var1=Serial1.read(); //Lee, del modulo RFID, byte a byte los bytes que componen el ID de la tag y se guadan en
la variable var1
      Serial.write(var1); //Imprime o escribe en la pantalla de la PC los bytes que componen el ID de la tag, guardados
en la variable var1
      cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso hasta que cont1 tenga el mismo
valor que lon1
    }
  }
  Serial.println(); //Imprime un ENTER en la pantalla de la PC
}

//SI SE ENCUENTRA MAS DE UNA ETIQUETA.
while(numtag > 1){

  cont1=0; //Variable de control que se ocupa para particionar el arreglo de bytes de la respuesta del modulo RFID, y
asi poder visualizar los bytes en la forma que se necesiten
  x++; //Lleva el conteo de la cantidad de etiquetas leidas por el modulo RFID, y las imprime en pantalla de la PC
para una mejor presentacion en pantalla

  if(cont1<1){
    if (Serial1.available()){
      var1 = Serial1.read(); //Guarda el primer byte del arreglo de la respuesta de la orden de inventario del modulo
RFID
      //Serial.println(var1, HEX); //Imprime en la pantalla de la PC el primer byte de respuesta del modulo RFID, en
formato HEX
      cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
    }
  }

  //LONGITUD DEL ID DE LA TAG
  if(cont1<2){
    if (Serial1.available()){

```

```

lon1 = Serial1.read(); //Guarda el segundo byte leído del modulo RFID en la variable lon1 el cual muestra la
longitud del ID de la tag
//Serial.println("Longitud de ID tag: ");
//Serial.println(lon1, HEX); //Imprime en la pantalla de la PC el segundo byte de respuesta del modulo RFID en
la pantalla de la PC guardado en la variable lon1 en formato HEX, el cual muestra la longitud de EPC o ID de la tag
cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
}
}
//Serial.println(); //Imprime un ENTER en la pantalla de la PC

while(cont1<6){
if (Serial1.available()){
var1 = Serial1.read(); //Guarda los bytes leídos del modulo RFID en la variable var1 los cuales muestran la
longitud de todo el arreglo de respuesta del modulo RFID y bytes reservados
//Serial.print(var1, HEX); //Imprime en la pantalla de la PC, byte a byte, los bytes guardados en la variable var1
cont1++; //Aumenta en uno la variable cont1 (para seguir con el siguiente proceso la variable tiene que tener el
valor de 6)
}
}
//Serial.println(); //Imprime un ENTER en la pantalla de la PC

//ID DE LA TAG
//if(cont1<lon1){
//Serial.println("ID tag: ");
Serial.print(x); //Imprime en la pantalla de la PC el valor de la variable x, la cual muestra el orden del ID de las
tags leídas
Serial.print(": ");
//}
//Serial.println(); //Imprime un ENTER en la pantalla de la PC

//Imprime los bytes que componen el ID de las tag en formato HEX
if(hx==1){
while(cont1<lon1){
if(Serial1.available()){
var1 = Serial1.read(); //Lee, del modulo RFID, byte a byte los bytes que componen el ID de la tag y se guardan
en la variable var1
completar(var1); //Se llama a la función completar mandándole la variable var1, y así visualizar mejor los
números o bytes que tienen un cero a la izquierda, imprimiendo el cero
Serial.print(var1, HEX); //Imprime en la pantalla de la PC los bytes que componen el ID de la tag, guardados en
la variable var1, en formato HEX
cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso hasta que cont1 tenga el mismo
valor que lon1
}
}
Serial.println(); //Imprime un ENTER en la pantalla de la PC
}

//Imprime los bytes que componen el ID de las tag en formato ascii
if(hx==2){
while(cont1<lon1){
if(Serial1.available()){
var1 = Serial1.read(); //Lee, del modulo RFID, byte a byte los bytes que componen el ID de la tag y se guardan
en la variable var1
Serial.write(var1); //Imprime o escribe en la pantalla de la PC los bytes que componen el ID de la tag, guardados
en la variable var1
cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso hasta que cont1 tenga el mismo
valor que lon1
}
}
Serial.println(); //Imprime un ENTER en la pantalla de la PC
}
}

```



```

modulo
  x=1; //Lleva el conteo de la cantidad de etiquetas leidas por el modulo RFID, y las imprime en pantalla de la PC para
una mejor presentacion en pantalla
  p1=0; //Se inicializa la variable p1 con el valor de 0, esta variable tiene la posicion del arreglo Freq a la hora de
imprimir en la pantalla de la PC
  p2=2; //Se inicializa la variable p2 con el valor 2, esta variable tiene la posicion del arreglo Freq en el cual se
guardara la frecuencia, y empieza de 2 ya que el modulo RFID da la frecuencia de operacion al revés
  hx=1; //Esta variable controla la forma de visualizar en pantalla de la PC el ID de la tag
  //cuando hx=1 se visualiza en formato HEX y si hx=2 se visualiza en formato ascii
  hy=0; //Variable de control para imprimir en pantalla de la PC el menu de visualizacion de datos una sola vez asta que
la variable cambia de valor

while(var1=='2' || (var1 == 'w' || var1=='W')){

  //visualizar(); //Se llama a la funcion para poder elegir la forma en que se visualizara los datos de el modulo RFID

  if(cont1<1){
    if(Serial1.available()){
      var1=Serial1.read(); //Guarda el primer byte del arreglo de la respuesta de la orden de inventario del modulo RFID
      //Serial.println(var1, HEX); //Imprime en la pantalla de la PC el primer byte de respuesta del modulo RFID, en
formato HEX
      cont1++; //Aumenta en uno la variable cont13 y continua con el siguiente proceso
    }
  }

  //LONGITUD DEL ID DE LA TAG
  if(cont1<2){
    if (Serial1.available()){
      lon1 = Serial1.read(); //Guarda el segundo byte leido del modulo RFID en la variable lon1 el cual muestra la
longitud del ID de la tag
      //Serial.println("Longitud de ID tag: ");
      //Serial.println(lon1, HEX); //Escribe o imprime en la pantalla de la PC el segundo byte de respuesta del modulo
RFID en la pantalla de la PC guardado en la variable lon1 en formato HEX, el cual muestra la longitud de EPC o ID de la
tag
      cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
    }
  }
  //Serial.println(); //Imprime un ENTER en la pantalla de la PC

  //NUMERO DE ETIQUETAS LEIDAS
  if(cont1<3){
    if (Serial1.available()){
      numtag = Serial1.read(); //Guarda el tercer byte del arreglo de la respuesta del modulo RFID en la variable numtag
el cual muestra el numero de etiquetas leidas por el modulo RFID
      Serial.println("Numero de etiquetas leidas ");

      if(numtag<1){
        Serial.println("NO HAY ETIQUETAS.");
        while(Serial1.available()){
          var1 = Serial1.read(); //Guarda el tercer byte leido del modulo RFID en la variable var1 el cual muestra el
numero de tag leidas, si el byte es 00 el modulo no detecto ninguna etiqueta
          Serial.print(var1, HEX); //Imprime en la pantalla de la PC el tercer byte de respuesta del modulo RFID
guardado en la variable var1 en formato HEX, el cual muestra que el modulo no pudo leer ninguna etiqueta
        }
        cont1=9000; //Se le otorga el valor de 9000 a la variable cont1 para que no siga las demas ordenes y termine todo
        lon1=100; //Se le otorga el valor de 100 a la variable lon1 para que no siga los demas procesos relacionados con
esta variable
      }

      Serial.println(numtag); //Imprime en la pantalla de la PC el tercer byte de respuesta del modulo RFID guardado en
la variable numtag en formato HEX, la cual muestra el numero de tag leidas

```

```

    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
  }
}

//COMANDO RSSI
if(cont1<4){
  Serial.println(); //Imprime un ENTER en la pantalla de la PC
  if(Serial1.available()){
    var1 = Serial1.read(); //Guarda el cuarto byte del arreglo de la respuesta del modulo RFID y el cual es el valor de
RSSI
    //RSSI = var1; //Se otorga el valor de la variable var1 a la variable RSSI
    I = (var1&0x0F)*2; //Los valores de las variables I y Q representan la intensidad de la senal en que fue leida la
tag, entre mayor sea el valor mas intensa sera la senal o
    Q = (var1>>4)*2; //mas cercana estara la tag del modulo RFID (el valor mayor es 28), entre menor sea el valor
mas lejano estara la tag del modulo (el valor minimo es 0)
    //Serial.println(var1, HEX); //Imprime en la pantalla de la PC el cuarto byte del arreglo de la respuesta del modulo
RFID guardado en la variable var1 en formato HEX
    Serial.println("Valor de RSSI ");
    Serial.print("-"); //Imprime en la pantalla de la PC el caracter -
    Serial.print(I); //Imprime en la pantalla de la PC el valor de la variable RSSI
    Serial.println(" dBm.");
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
  }
}

if(cont1<7){
  Serial.println(); //Imprime un ENTER en la pantalla de la PC
  Serial.println("Frecuencia de operacion ");
}

// FRECUENCIA
while(cont1<7){
  if(Serial1.available()){
    var1 = Serial1.read(); //Guarda los bytes leidos desde el modulo RFID, que representan la frecuencia de operacion
de dicho modulo
    Freq[p2] = var1; //Se guarda los bytes que estan en la variable var1 en el arreglo Freq, posicionandose en el lugar
de guadado del arreglo con la variable p2, y la posicion va de mayor a menor ya que la respuesta del modulo da la
frecuencia al revés
    p2--; //Disminuye en uno la variable p2, ya que se trata de girar el orden de los bytes para poder vizualisarla mejor
    //Serial.print(var1, HEX);
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
  }
}

//Convierte el valor de la frecuencia de HEX a decimal
if(cont1<8){
  Freq[0]= Freq[0]*0x10000; //Agrega cuatro ceros a la derecha del primer byte del valor de la frecuencia, por
ejemplo para el valor 0x0D esto es 0x0D0000
  //Serial.println(Freq[0], HEX); //Imprime en la pantalla de la PC el valor del primer byte del valor de la frecuencia
en formato HEX
  Freq[1]= Freq[1]*0x100; //Agrega dos ceros a la derecha del segundo byte del valor de la frecuencia, por ejemplo
para el valor 0x3C esto es 0x3C00
  //Serial.println(Freq[1], HEX); //Imprime en la pantalla de la PC el valor de la variable Freq[1] en formato HEX
  //Serial.println(Freq[2], HEX); //Imprime en la pantalla de la PC el valor de la variable Freq[2] en formato HEX
  Freq[3]= Freq[0]+Freq[1]+Freq[2]; //Se suman los valores de los tres bytes que componen el valor de la frecuencia
y su resultado se guarda en la variable Freq[3], estas variables son de tipo long (4 byte)
  Serial.print(Freq[3]); //Imprime en la pantalla de la PC el resultado de la suma anterior que esta en la variable
Fraq[3], por ejemplo 0x0D0000+0x3C00+0xAC=0x0D3CAC esto equibale en decimal 867500 lo cual es el valor de la
frecuencia
}
//Serial.println(); //Imprime en la pantalla de la PC un ENTER

```

```

if(cont1<8){
  Serial.println(" kHz.");
}
Serial.println(); //Imprime en la pantalla de la PC un ENTER

while(cont1<10){
  if (Serial1.available()){
    var1 = Serial1.read(); //Guarda los bytes leidos del modulo RFID en la variable var1 los cuales muestran la
longitud de todo el arreglo de respuesta del modulo RFID y bytes resevados
    //Serial.print(var1, HEX); //Imprime en la pantalla de la PC, byte a byte, los bytes guardados en la variable var1
    cont1++; //Aumenta en uno la variable cont1 (para seguir con el siguiente proceso la variable tiene que tener el
valor de 10)
  }
}

//ID DE LA ETIQUETA
if(cont1<lon1){
  Serial.println(); ///Imprime en la pantalla de la PC un ENTER
  Serial.println("ID tag \r\n");
  Serial.print(x); //Imprime en la pantalla de la PC el valor de la variable x, la cual muestra el orden de los IDs de las
tags leidas, para una mejor visualizacion
  Serial.print(": ");
}
//Serial.println(); //Imprime en la pantalla de la PC un ENTER

//Imprime los bytes que componen el ID de las tag en formato HEX
if(hx==1){
  while(cont1<lon1){
    if(Serial1.available()){
      var1 = Serial1.read(); //Lee, del modulo RFID, byte a byte los bytes que componen el ID de la tag y se guadan en
la variable var1
      completar(var1); //Se llama a la funcion completar mandandole la variable var1, y asi visualizar mejor los
numeros o bytes que tienen un cero a la izquierda, imprimiendo el cero
      Serial.print(var1, HEX); //Imprime en la pantalla de la PC los bytes que componen el ID de la tag, guardados en
la variable var1, en formato HEX
      cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso hasta que cont1 tenga el mismo
valor que lon1
    }
  }
  Serial.println(); //Imprime un ENTER en la pantalla de la PC
}

//Imprime los bytes que componen el ID de las tag en formato ascii
if(hx==2){
  while(cont1<lon1){
    if(Serial1.available()){
      var1 = Serial1.read(); //Lee, del modulo RFID, byte a byte los bytes que componen el ID de la tag y se guadan en
la variable var1
      Serial.write(var1); //Imprime o escribe en la pantalla de la PC los bytes que componen el ID de la tag, guardados
en la variable var1
      cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso hasta que cont1 tenga el mismo
valor que lon1
    }
  }
  Serial.println(); //Imprime en la pantalla de la PC un ENTER
}

//SI SE ENCUENTRA MAS DE UNA ETIQUETA.
while(numtag > 1){

```

```

cont1=0; //Variable de control que se ocupa para partisionar el arreglo de bytes de la respuesta del modulo RFID, y
asi poder visualizar los bytes en la forma que se necesiten
x++; //Lleva el conteo de la cantidad de etiquetas leidas por el modulo RFID, y las imprime en pantalla de la PC
para una mejor presentacion en pantalla

if(cont1<1){
  if (Serial1.available( )){
    var1 = Serial1.read(); //Guarda el primer byte del arreglo de la respuesta de la orden de inventario del modulo
RFID
    //Serial.println(var1, HEX); //Imprime en la pantalla de la PC el primer byte de respuesta del modulo RFID, en
formato HEX
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
  }
}

//LONGITUD DEL ID DE LA TAG
if(cont1<2){
  if (Serial1.available( )){
    lon1 = Serial1.read(); //Guarda el segundo byte leido del modulo RFID en la variable lon1 el cual muestra la
longitud del ID de la tag
    //Serial.println("Longitud de ID tag: ");
    //Serial.println(lon1, HEX); //Imprime en la pantalla de la PC el segundo byte de respuesta del modulo RFID en
la pantalla de la PC guardado en la variable lon1 en formato HEX, el cual muestra la longitud de EPC o ID de la tag
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
  }
}
//Serial.println(); //Imprime en la pantalla de la PC un ENTER

while(cont1<10){
  if (Serial1.available( )){
    var1 = Serial1.read(); //Guarda los bytes leidos del modulo RFID en la variable var1 los cuales muestran la
longitud de todo el arreglo de respuesta del modulo RFID y bytes resevados
    //Serial.print(var1, HEX); //Imprime en la pantalla de la PC, byte a byte, los bytes guardados en la variable var1
    cont1++; //Aumenta en uno la variable cont1 (para seguir con el siguiente proceso la variable tiene que tener el
valor de 10)
  }
}
//Serial.println(); //Imprime en la pantalla de la PC un ENTER

//ID DE LA ETIQUETA
//if(cont1<lon1){
//Serial.println("ID tag: ");
Serial.print(x); //Imprime o escribe en la pantalla de la PC el valor de la variable x, la cual muestra el orden del ID
de las tags leidas
Serial.print(": ");
//}
//Serial.println(); //Imprime un ENTER en la pantalla de la PC

//Imprime los bytes que componen el ID de las tag en formato HEX
if(hx==1){
  while(cont1<lon1){
    if(Serial1.available()){
      var1 = Serial1.read(); //Lee, del modulo RFID, byte a byte los bytes que componen el ID de la tag y se guadan
en la variable var1
      completar(var1); //Se llama a la funcion completar mandandole la variable var1, y asi visualizar mejor los
numeros o bytes que tienen un cero a la izquierda, imprimiendo el cero
      Serial.print(var1, HEX); //Imprime en la pantalla de la PC los bytes que componen el ID de la tag, guardados en
la variable var1, en formato HEX
      cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso hasta que cont1 tenga el
mismo valor que lon1
    }
  }
}

```



```

Serial.println("\n\r\n\r"); //Imprime en la pantalla de la PC 3 ENTER
}

delay (1000); //Espera 1 segundo

p1=0; //La variable p1 se inicializa en 0, esta variable controla la posicion en el arreglo coman_leer1, en el cual se
guarda los caracteres leidos desde la PC
p2=3; //La variable p2 se inicializa en 3, esta variable controla la posicion en el arreglo comand_rfid1, y desde la
posicion 3 de este arreglo comienza el ID de la tag a buscar
x=2; //La variable x se inicializa en 3, esta variable particiona el arreglo de caracteres leidos desde la PC o controla el
orden en que es leido los caracteres escritos desde la PC
y=0; //La variable y se inicializa en 0, se utiliza para formar un byte de dos numeros hex formados con dos caracteres
trasformados en numeros hex
lim2=0; //La variable lim2 se inicializa en 0, se utiliza para conocer si se ingreso o digitado 24 caracteres maximo
para el ID de la tag y si no es asi el proceso termina y sale de la funcion
z1=0x00; //La variable z1 se inicializa en 0x00, se utiliza para llevar un conteo de cuantos bytes de dos numeros hex
se han formado para el arreglo de la orden de seccionar una tag
hx=1; //Esta variable controla la forma de visualizar en pantalla de la PC el ID de la tag
//cuando hx=1 se visualiza en formato HEX y si hx=2 se visualiza en formato ascii
hy=0; //Variable de control para imprimir en pantalla de la PC el menu de visualizacion de datos una sola vez asta
que la variable cambia de valor

while(var1=='3' || var1=='w' || var1=='W'){

//visualizar(); //Se llama a la funcion para poder elegir la forma en que se visualizara los datos de el modulo RFID

if(x==2){
if(hx==1){
Serial.println("Digite el ID de la etiqueta.");
Serial.println("El ID tiene que ser en hexadecimal");
x++; //Aumenta en uno la variable x y continua con el siguiente proceso
}
if(hx==2){
Serial.println("Digite el ID de la etiqueta.");
Serial.println("Existe diferencia entre mayusculas y minusculas.");
x++; //Aumenta en uno la variable x y continua con el siguiente proceso
}
}

if(Serial.available()){

//Lee los caracteres que se ingresan desde la PC y los transforma en bytes de dos numeros HEX
if(hx==1){
while(x<30){
//Lee el ID de la tag ingresado por la pc
while(Serial.available()){
delay(250); //Espera 1/4 de segundo
coman_leer1[p1] = Serial.read(); //Lee los caracteres escritos en la PC y las guarda en el arreglo coman_leer1,
la posicion en que se guarda en el arreglo es definido por la variable p1
var22=coman_leer1[p1]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo
coman_leer1, en la posicion definida por la variable p1
Serial.write(coman_leer1[p1]); //Se imprime en la pantalla de la PC el caracter guardado por el arreglo
coman_leer1 en la posicion definida por la variable p1
y++; //Aumenta en uno la variable y, para activar el primer proceso de crear un byte de dos numeros hex
lim2++; //Aumenta en uno la variable lim2, contando la cantidad de caracteres ingresados desde la PC
trans_hex(); //Se llama a la funcion para transformar los caracteres ingresados por la PC en byte de dos
numeros hex
p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer1
}
x++; //Aumenta en uno la variable x, y al llegar a la cantidad de 30 se procede a continuar con el siguiente
proceso
}
}
}

```

```

    }
}

//Obtiene los bytes hex de los caracteres ingresados por la computadora(ocupa los equivalentes en ASCII de dichos
caracteres)
if(hx==2){
    while(x<30){
        //Lee el ID de la tag ingresado por la pc
        while(Serial.available()){
            delay(250); //Espera 1/4 de segundo
            coman_leer1[p1] = Serial.read(); //Lee los caracteres escritos en la PC y las guarda en el arreglo coman_leer1,
la posicion en que se guarda en el arreglo es definido por la variable p1
            var22=coman_leer1[p1]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo
coman_leer1, en la posicion definida por la variable p1
            Serial.write(coman_leer1[p1]); //Se imprime o escribe en la pantalla de la PC el caracter guardado por el
arreglo coman_leer1 en la posicion definida por la variable p1
            lim2++; //Aumenta en uno la variable lim2, contando la cantidad de caracteres ingresados desde la PC
            comand_rfid1[p2] = coman_leer1[p1]; //Se otorga al arreglo comand_rfid1 en la posicion p2, el valor del
arreglo coman_leer1 en la posicion p1, porque el arreglo coman_leer1 ya contiene el byte en hex del equivalente en
ASCII del caracter
            p2++; //Aumenta en uno la variable p2 para avanzar en la posccion del arreglo comand_rfid1
            z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de
la orden de seleccionar una tag especifica
            p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer1
        }
        x++; //Aumenta en uno la variable x, y al llegar a la cantidad de 30 se procede a continuar con el siguiente
proceso
    }
}

if(lim2>24){
    Serial.println("\n\r \n\r ERROR: El ID de la tag tiene que ser de una longitud de 24 digitos MAXIMO");
    cont1=8000; //Se otorga el valor de 8000 a la vriable para que ya no continue con los procesos y reinicie la
funcion
    var1=5; //La variable var1 se le da el valor de 5 para que ya no continue con los procesos relacionados a esta
variable
    x=150; //La variable x se le da el valor de 150 para que ya no continue con los procesos y termine
    cont1=3; //La variable cont1 se le da el valor de 4 para que ya no continue con los procesos y termine
}
//Serial.println(); //Imprime un ENTER en la pantalla de la PC

if(hx==2){
    if(lim2>12){
        Serial.println("\n\r \n\r ERROR: El ID de la tag tiene que ser de una longitud de 12 caracteres MAXIMO");
        cont1=8000; //Se otorga el valor de 8000 a la vriable para que ya no continue con los procesos y reinicie la
funcion
        var1=5; //La variable var1 se le da el valor de 5 para que ya no continue con los procesos relacionados a esta
variable
        x=150; //La variable x se le da el valor de 150 para que ya no continue con los procesos y termine
        cont1=3; //La variable cont1 se le da el valor de 4 para que ya no continue con los procesos y termine
    }
}

if(x==30){
    if(y==0){
        z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la
orden de seleccionar una tag especifica
    }
    Serial.println(); //Imprime un ENTER en la pantalla de la PC
}
}

```

```

if(x==30){
    hex_impar(); //Se llama a la funcion cuando los numero ingresados de la PC son impares
}

if(x==30){
    while(p2<15){
        comand_rfid1[p2]=0x00; //Rellena las restantes posiciones del arreglo comand_rfid1 con el numero 0x00
        p2++; //Aumenta en uno la variable p2 para avanzar en la posicion del arreglo comand_rfid1
        z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la
orden de seleccionar una tag especifica
    }
}

if(p2==16){
    z1=z1-0x01; //Se le resta un uno a la variable z1
}

//Byte que muestran el tamaño del ID de la tag y de todo el arreglo del comando comand_rfid1
if(x==30){
    comand_rfid1[1]=z1+0x03; //Se le suma 0x03 a la variable z1 o tres posiciones, y se guarda en la posicion 1 del
arreglo comand_rfid1, en esa posicion se debe colocar el tamaño de todo el arreglo que se le enviara al modulo RFID en
bytes
    comand_rfid1[2]=z1; //Se le asigna el valor de la variable z1 a la posicion 2 del arreglo comand_rfid1, en esa
posicion se debe colocar el tamaño de todo el ID de la tag, en el arreglo que se le enviara al modulo RFID en bytes
}

if(x==30){
    //Serial.println(p2); //Se imprime en la pantalla de la PC el valor de la variable p2
    //Serial.println(comand_rfid1[1], HEX); //Imprime en la pantalla de la PC la posicion 1 del arreglo comand_rfid1
en formato hex
    //Serial.println(comand_rfid1[2], HEX); //Imprime en la pantalla de la PC la posicion 2 del arreglo comand_rfid1
en formato hex
    Serial1.write(comand_rfid1, p2); //Se escribe todo el arreglo comand_rfid1 en el modulo RFID, el cual da la
orden de buscar una tag especifica
    //for(int i=0; i<(p2); i++){
    //Serial.println(comand_rfid1[i], HEX); //Imprime todo el arreglo comand_rfid1 en la pantalla de la PC en
formato hex
    //}
}

} //Fin if leer del teclado

//Leyendo la respuesta del modulo RFID
if(Serial1.available()){

    cont1=0; //Se inicializa en 0 la variable cont1, la cual permite dividir el arreglo de la respuesta del modulo RFID
para tener una mejor presentacion
    Serial.println(); //Imprime un ENTER en la pantalla de la PC

    while(cont1<2){
        if (Serial1.available( )){
            delay(250); //Espera 1/4 de segundo
            var1 = Serial1.read(); //Guarda los primeros 2 bytes leidos del modulo RFID en la variable var1
            //Serial.print(var1, HEX); //Imprime en la pantalla de la PC en formato HEX los bytes guardados en la variable
var1
            cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso hasta que cont1 tenga el valor
de 2
        }
    }
    Serial.println(); //Imprime un ENTER en la pantalla de la PC
}

```

```

//LEE EL BYTE QUE DICE SI HAY O NO HALLO TAG
if(cont1<3){
  if (Serial1.available()){
    var1 = Serial1.read(); //Guarda el tercer byte leído del modulo RFID en la variable var1, el cual muestra si la tag
    fue encontrada la tag especifica o no
    //Serial.println(var1, HEX); //Imprime el tercer byte en la pantalla de la PC en formato HEX
    //Si el tercer byte de respuesta del modulo RFID su valor es 0 entonces la tag buscada fue encontrada y si su
    valor es 9 la tag buscada no fue encontrada
    if(var1==0){
      Serial.println("Etiqueta ENCONTRADA");
    }else if(var1==9){
      Serial.println("Etiqueta NO ENCONTRADA");
    }
    cont1++; //Aumenta en uno la variable cont1 para poder pasar al siguiente proceso
  }
}

} //Fin leer del modulo RFID

} //Fin while si var1 = 3
if(cont1!=8000 || cont1==8000){
  if(cont1==3 || cont1==8000){
    Serial.println("\n\r \r\n \r\n >>> Precione 3 si quiere seleccionar otra etiqueta <<< \r\n");
    Serial.println(" >>> Precione w para salir de la funcion <<< \r\n \r\n");
    cont1++; //Aumenta en uno la variable caont1 para imprimir una sola ves los mansajes anteriores
  }
}
} //Fin primer while
} //Fin coman_selectag

/*Esta funcion le permite al usuario modificar el ID de una etiqueta, pero para realizarlo el usuario debe ingresar el
password
de dicha etiqueta para modificarla, el password tiene que ser en formato hexadecimal y no mas de ocho digitos, el ID
puede ser de
dos formas, en formato hexadecimal o en formato ASCII (letras del abecedario y caracteres, que se transforma en su
equivalente ASCII
o numero hexadecimal), la forma o el formato en que se modifica o escribe la etiqueta lo determina el usuario
(escogiendo una opcion
en el menu visualizar), la funcion tambien muestra si se modifico correctamente o no*/
void coman_readtag(){

while (mx==1){

  x=0; //Se inicializa la variable x para inicializar los proceso de la funcion
  if(Serial.available()){
    var1=Serial.read(); //Guarda lo que lee de la PC en la variable var1
    //El caracter w es el caracter de salida de toda la funcion actual
    if(var1=='w' || var1=='W'){
      var1=0; //Se le otorga el valor de 0 a la variable var1 para que no continúe con los procesos relacionados con la
      variable
      mx=2; //Se le otorga el valor de 2 a la variable mx para salir completamente de la funcion actual
    }
  }

  if (var1== '4'){
    comand_rfid1[0]=0x35; //Se otorga el valor de 0x35 en la posicion 0 del arreglo comand_rfid1
  }

  if(var1=='4'){
    Serial.println("\n\r \n\r \n\r \n\r"); //Imprime en la pantalla de la PC 5 ENTER
    Serial.println("

```

```

+++++");
  Serial.println("      ++++++ <<<<<<<<  ESCRIBIR  ETIQUETA  >>>>>>>
+++++");
  Serial.println("
+++++");
  Serial.println("\n\r\n\r"); //Imprime en la pantalla de la PC 3 ENTER
}

delay (1000); //Espera 1 segundo

p1=0; //La variable p1 se inicializa en 0, esta variable controla la posicion en el arreglo coman_leer1, en el cual se
guarda los caracteres leidos desde la PC
p2=4; //La variable p2 se inicializa en 4, esta variable controla la posicion en el arreglo comand_rfid1, y desde la
posicion 4 de este arreglo comienza el password de la tag a escribir
x=2; //La variable x se inicializa en 3, esta variable particiona el arreglo de caracteres leidos desde la PC o controla el
orden en que es leido los caracteres escritos desde la PC
y=0; //La variable y se inicializa en 0, se utiliza para formar un byte de dos numeros hex formados con dos caracteres
trasformados en numeros hex
lim1=0; //La variable lim1 se inicializa en 0, se utiliza para conocer si se ingreso o digitado 8 caracteres para el
password de la tag y si no es asi el proceso termina y sale del programa
lim2=0; //La variable lim2 se inicializa en 0, se utiliza para conocer si se ingreso o digitado 24 caracteres maximo
para el ID de la tag y si no es asi el proceso termina y sale de la funcion
z1=0x00; //La variable z1 se inicializa en 0x00, se utiliza para llevar un conteo de cuantos bytes de dos numeros hex
se han formado
hx=1; //Esta variable controla la forma de visualizar en pantalla de la PC el ID de la tag
//cuando hx=1 se visualiza en formato HEX y si hx=2 se visualiza en formato ascii
hy=0; //Variable de control para imprimir en pantalla de la PC el menu de visualizacion de datos una sola vez asta que
la variable cambia de valor

while(var1=='4'){

  //visualizar(); //Se llama a la funcion para poder elegir la forma en cual se puede visualizar los datos de respuesta del
modulo RFID o ID de la etiqueta

  if(x==2){
    if(hx==1 || hx==2){
      Serial.println("\n\rIngrese password:");
      Serial.println(" (En lenguaje hexadecimal)");
      Serial.println(" (Password por defecto: 00 00 00 00)");
      x++; //Aumenta en uno la variable x y continua con el siguiente proceso
    }
  }

  if(Serial.available()){

    comand_rfid1[2]=0x01; //Se otorga el valor 0x01 en la posicion 2 del arreglo comand_rfid1
    comand_rfid1[3]=0x02; //Se otorga el valor 0x02 en la posicion 3 del arreglo comand_rfid1

    while(x<13){
      //Lee password de la tag ingresado por la pc
      while(Serial.available()){
        delay(250); //Espera 1/4 de segundo
        coman_leer1[p1] = Serial.read(); //Lee los caracteres escritos en la PC y las guarda en el arreglo coman_leer1, la
posicion en que se guarda en el arreglo es definido por la variable p1
        var22=coman_leer1[p1]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo
coman_leer1, en la posicion definida por la variable p1
        Serial.write(coman_leer1[p1]); //Se imprime o escribe en la pantalla de la PC el caracter guardado por el arreglo
coman_leer1 en la posicion definida por la variable p1
        y++; //Aumenta en uno la variable y
        lim1++; //Aumenta en uno la variable lim1, contando la cantidad de caracteres ingresados desde la PC
        trans_hex(); //Se llama a la funcion para tranformar los caracteres ingresados por la PC en byte de dos numeros

```

```

hex
  p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer1
  }
  x++; //Aumenta en uno la variable x, y al llegar a la cantidad de 13 se procede a continuar con el siguiente proceso
  }

  if(lim1>8){
    Serial.println("\n\r\n\r ERROR: Password tiene que ser MAXIMO de 8 digitos");
    cont1=8000; //Se otorga el valor de 8000 a la vriable para que ya no continue con los procesos y reinicie la
funcion
  var1=5; //La variable var1 se le da el valor de 5 para que ya no continue con los procesos relacionados a esta
variable
  x=150; //La variable x se le da el valor de 150 para que ya no continue con los procesos y termine
  cont1=4; //La variable cont1 se le da el valor de 4 para que ya no continue con los procesos y termine
  }
  //Serial.println(); //Imprime un ENTER en la pantalla de la PC

  if(x==13){
    hex_impar(); //Se llama a la funcion cuando los numero ingresados de la PC son impares
  }

  if(x==13){
    while(p2<8){
      comand_rfid1[p2]=0x00; //Rellena las restantes posiciones del arreglo comand_rfid1 con el numero 0x00
      p2++; //Aumenta en uno la variable p2 para avanzar en la posicion del arreglo comand_rfid1
      //z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la
orden de escribir una tag
    }
  }

  if(x==13){
    if(hx==1){
      Serial.println("\n\r\n\rIngrese el ID que se quiere escribir en la tag: ");
      Serial.println(" (El ID tiene que ser en hexadecimal)");
    }
    if(hx==2){
      Serial.println("\n\r\n\rIngrese el ID que se quiere escribir en la tag: ");
    }
    x++; //Aumenta en uno la variable x para poder seguir con el siguiente proceso
    y=0; //La variable y se inicializa en 0, se utiliza para formar un byte de dos numeros hex formados con dos
caracteres trasformados en numeros hex
    z1=0x00; //La variable z1 se inicializa en 0x00, se utiliza para llevar un conteo de cuantos bytes de dos numeros
hex se han formado
  }

  delay(1000); //Espera 1 segundo

  if(Serial.available()){

    if(x==14){
      //Serial.println("ID de la tag: ");
      if(lim1==8){
        z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la
orden de seleccionar una tag especifica
        //p2++; //Aumenta en uno la variable p2 para que se salte una posicion en el arreglo comand_rfid1
      }
      if(lim1<8){
        z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la
orden de seleccionar una tag especifica
        p2=p2+1; //Aumenta en uno la variable p2 para que se salte una posicion en el arreglo comand_rfid1
      }
    }
  }

```

```

}

//Lee los caracteres que se ingresan desde la PC y los transforma en bytes de dos numeros HEX
if(hx==1){
  while(x<46){
    //Lee ID de la tag ingresado por la PC
    while(Serial.available()){
      delay(250); //Espera 1/4 de segundo
      coman_leer1[p1] = Serial.read(); //Lee caracteres escritas en la PC y las guarda en el arreglo coman_leer1, el
      cual la posicion en el cual se guardara el caracter en el arreglo esta definida por p1
      var22=coman_leer1[p1]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo
      coman_leer1, en la posicion definida por la variable p1
      Serial.write(coman_leer1[p1]); //Se imprime en la pantalla de la PC el caracter guardado por el arreglo
      coman_leer1 en la posicion definida por la variable p1
      y++; //Aumenta en uno la variable y, para activar el primer proceso de crear un byte de dos numeros hex
      lim2++; //Aumenta en uno la variable lim2, contando la cantidad de caracteres ingresados desde la PC
      trans_hex(); //Se llama a la funcion para tranformar los caracteres ingresados por la PC en byte de dos
      numeros hex
      p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer1
    }
    x++; //Aumenta en uno la variable x, y al llegar a la cantidad de 46 se procede a continuar con el siguiente
    proceso
  }
}

//Obtiene los bytes hex de los caracteres ingresados por la computadora(ocupa los equivalentes en ASCII de dichos
caracteres)
if(hx==2){
  while(x<46){
    //Lee ID de la tag ingresado por la PC
    while(Serial.available()){
      delay(250); //Espera 1/4 de segundo
      coman_leer1[p1] = Serial.read(); //Lee caracteres escritas en la PC y las guarda en el arreglo coman_leer1, el
      cual la posicion en el cual se guardara el caracter en el arreglo esta definida por p1
      var22=coman_leer1[p1]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo
      coman_leer1, en la posicion definida por la variable p1
      Serial.write(coman_leer1[p1]); //Se imprime o escribe en la pantalla de la PC el caracter guardado por el
      arreglo coman_leer1 en la posicion definida por la variable p1
      lim2++; //Aumenta en uno la variable lim2, contando la cantidad de caracteres ingresados desde la PC
      comand_rfid1[p2]=coman_leer1[p1];
      p2++; //Aumenta en uno la variable p2 para avanzar en la posccion del arreglo comand_rfid1
      z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la
      orden de seleccionar una tag especifica
      p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer1
    }
    x++; //Aumenta en uno la variable x, y al llegar a la cantidad de 46 se procede a continuar con el siguiente
    proceso
  }
}

if(lim2>24){
  Serial.println("\n\r\n\r ERROR: El ID de la tag tiene que ser de una longitud de 24 digitos MAXIMO");
  cont1=8000; //Se otorga el valor de 8000 a la vriable para que ya no continue con los procesos y reinicie la
  funcion
  var1=5; //La variable var1 se le da el valor de 5 para que ya no continue con los procesos relacionados a esta
  variable
  x=150; //La variable x se le da el valor de 150 para que ya no continue con los procesos y termine
  cont1=4; //La variable cont1 se le da el valor de 4 para que ya no continue con los procesos y termine
}
//Serial.println(); //Imprime un ENTER en la pantalla de la PC

```

```

if(hx==2){
  if(lim2>12){
    Serial.println("\n\r\n\r ERROR: El ID de la tag tiene que ser de una longitud de 12 caracteres MAXIMO");
    cont1=8000; //Se otorga el valor de 8000 a la variable para que ya no continúe con los procesos y reinicie la
funcion
    var1=5; //La variable var1 se le da el valor de 5 para que ya no continúe con los procesos relacionados a esta
variable
    x=150; //La variable x se le da el valor de 150 para que ya no continúe con los procesos y termine
    cont1=4; //La variable cont1 se le da el valor de 4 para que ya no continúe con los procesos y termine
  }
}

if(x==46){
  hex_impar(); //Se llama a la funcion cuando los numero ingresados de la PC son impares
}
Serial.println(); //Imprime un ENTER en la pantalla de la PC

} //Fin Segundo if de leer del teclado

if(x==46){
  //p2=p2-1; //Se disminuye en uno la variable p2 para tener un mejor control del arreglo comand_rfid1 y
posicionarse en lugar adecuado en dicho arreglo para ingresar los datos para el comando de escribir tag
  while(p2<21){
    comand_rfid1[p2]=0x00; //Rellena las restantes posiciones del arreglo comand_rfid1 con el numero 0x00
    p2++; //Aumenta en uno la variable p4 para avanzar en la posicion del arreglo comand_rfid1
    z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la
orden de escribir una tag
  }
}

//Byte que muestran el tamaño del ID de la tag y de todo el arreglo del comando comand_rfid1
if(x==46){
  comand_rfid1[1]=z1+0x09; //Se le suma 0x09 a la variable z1 o nueve posiciones, y se guarda en la posicion 1 del
arreglo comand_rfid1, en esa posicion se debe colocar el tamaño de todo el arreglo que se le enviara al modulo RFID en
bytes
  comand_rfid1[8]=z1/0x02; //Se le asigna el valor de la variable z1 dividida en 2 a la posicion 8 del arreglo
comand_rfid1, en esa posicion se debe colocar el tamaño de todo el ID de la tag, en palabras (palabra = dos byte), en el
arreglo que se le enviara al modulo RFID en bytes
}

//if(y==1){
//p2++; //Aumenta en uno la variable p2 para avanzar de posiciones en el arreglo comand_rfid1
//}

if(x==46){
  //Serial.println(p2); //Imprime el valor que posee la variable p2
  Serial1.write(comand_rfid1, p2); //Se imprime todo el arreglo comand_rfid1 en el modulo RFID, el cual da la
orden de escribir una tag especifica
  //for(int i=0; i<(p2); i++){
  //Serial.println(comand_rfid1[i], HEX); //Imprime todo el arreglo comand_rfid1 en la pantalla de la PC en
formato hex
  //}
}

} //if leer del teclado

//Leyendo la respuesta del modulo RFID
if(Serial1.available()){

  cont1=0; //Se inicializa en 0 la variable cont1, la cual permite dividir el arreglo de la respuesta del modulo RFID
  Serial.println(); //Imprime un ENTER en la pantalla de la PC
}

```

```

while(cont1<2){
  if (Serial1.available( )){
    delay(250); //Espera 1/4 de segundo
    var1 = Serial1.read(); //Guarda los bytes leidos del modulo RFID en la variable var1
    //Serial.print(var1, HEX); //Imprime en la pantalla de la PC en formato HEX los bytes guardados en la variable
var1
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso hasta que cont4 tenga el valor
de 2
  }
}
Serial.println(); //Imprime un ENTER en la pantalla de la PC

//MUESTRA SI SE ESCRIBIO CORRECTAMENTE LA ETIQUETA O NO
if(cont1<3){
  if (Serial1.available( )){
    var1 = Serial1.read(); //Guarda el tercer byte leido del modulo RFID en la variable var1, el cual muestra si la tag
fue escrita la tag especifica o no
    //Serial.println(var1, HEX); //Imprime el tercer byte en la pantalla de la PC en formato HEX
    //Si el tercer byte de respuesta del modulo RFID su valor es 0 entonces la tag se escribio correctamente, si su
valor es 9 la tag no fue encontrada para escribir, si el valor es 2 existe un error en el password ingresado y si es diferente
de 0 la tag no se escribio correctamente
    if(var1==0){
      Serial.println("Se escribio la etiqueta CORRECTAMENTE");
    }else if(var1==9){
      Serial.println(" Error al escribir etiqueta: Etiqueta no encontrada");
    }else if(var1==2){
      Serial.println(" Error al escribir etiqueta: Error password");
    }else if(var1!=0){
      Serial.println(" Error al escribir la etiqueta.");
    }
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
  }
}

while(cont1<4){
  if (Serial1.available( )){
    delay(250); //Espera 1/4 de segundo
    var1 = Serial1.read(); //Guarda el cuarto byte de la respuesta del modulo RFID el cual dice cuantas palabras
(parabra=dos byte) fueron escritas correctamente
    //Serial.print(var1, HEX); //Imprime en la pantalla de la PC en formato HEX el byte guardado en la variable var1
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
  }
}

} //Fin de leer del modulo RFID

} //Fin de while si var1 = n
if(cont1!=8000 || cont1==8000){
  if(cont1==4 || cont1==8000){
    Serial.println("\n\r \r\n \r\n >>> Precione 4 si desea escribir otra etiqueta <<< \r\n");
    Serial.println(" >>> Precione w para salir de la funcion <<< \r\n \r\n");
    cont1++; //Aumenta en uno la variable caont1 para imprimir una sola vez los mensajes anteriores
  }
}
} //Fin del primer while
} //Fin de coman_readtag()

/*Esta funcion cambia el password de la etiqueta RFID, dicho password tiene que ser en formato hexadecimal y como
maximo de ocho caracteres,
se debe ingresar primero el password actual de dicha etiqueta y luego el nuevo password al cual se cambiara, luego le

```

```

mandara el comando al
modulo RFID para ser ejecutado, la funciomm tambien le dira al usuario si fue correctamente cambiado el password de la
etiqueta o no*/
void coman_passwtag(){

while (mx==1){

x=0; //Se inicializa la variable x para inicializar los proceso de la funcion
if(Serial.available()){
var1=Serial.read(); //Guarda lo que lee de la PC en la variable var1
//El caracter w es el caracter de salida de toda la funcion actual
if(var1=='w' || var1=='W'){
var1=0; //Se le otorga el valor de 0 a la variable var1 para que no continue con los procesos relacionados con la
variable
mx=2; //Se le otorga el valor de 2 a la variable mx para salir completamente de la funcion actual
}
}

if (var1== '5'){
comand_rfid1[0]=0x35; //Se otorga el valor de 0x35 en la posicion 0 del arreglo comand_rfid1
}

if(var1=='5'){
Serial.println("\n\r\n\r\n\r\n\r"); //Imprime en la pantalla de la PC 5 ENTER
Serial.println("
+++++");
Serial.println(" +++++ <<<<<<< CAMBIAR CONTRASENA DE LA ETIQUETA >>>>>>>
+++++");
Serial.println("
+++++");
Serial.println("\n\r\n\r"); //Imprime en la pantalla de la PC 3 ENTER
}

delay (1000); //Espera 1 segundo

p1=0; //La variable p1 se inicializa en 0, esta variable controla la posicion en el arreglo coman_leer1, en el cual se
guarda los caracteres leidos desde la PC
p2=4; //La variable p2 se inicializa en 4, esta variable controla la posicion en el arreglo comand_rfid1, y desde la
posicion 4 de este arreglo comienza el password de la tag
x=2; //La variable x se inicializa en 3, esta variable particiona el arreglo de caracteres leidos desde la PC o controla el
orden en que es leido los caracteres escritos desde la PC
y=0; //La variable y se inicializa en 0, se utiliza para formar un byte de dos numeros hex formados con dos caracteres
trasformados en numeros hex
co1=4; //La variable co1 se inicializa en 4, porque cuenta desde el inicio del los dyte del password actual de la tag y
esto comienza en la posicion 5 del arreglo comand_rfid1, y el numero que tendra esta variable co1 es la posicion en
donde se colocara el byte que representa el numero de palabras del nuevo password que se escribira
lim1=0; //La variable lim1 se inicializa en 0, se utiliza para conocer si se ingreso o digitado 8 caracteres para el
password actual de la tag y si no es asi el proceso termina y sale del programa
lim2=8; //La variable lim2 se inicializa en 8, se utiliza para conocer si se ingreso o digitado 8 caracteres para el
password nuevo de la tag y si no es asi el proceso termina y sale del programa
z1=0x00; //La variable z1 se inicializa en 0x00, se utiliza para llevar un conteo de cuantos bytes del nuevo password,
bytes de dos numeros hex se han formado
z12=0x04; //La variable z12 se inicializa en 0x04, se utiliza para llevar un conteo de cuantos bytes del password
actual o viejo, bytes de dos numeros hex se han formado

while(var1=='5' || (var1=='w' || var1=='W')){

if(x==2){
Serial.println("\n\r\n\rINGRESE PASSWORD ACTUAL:");
Serial.println(" (En lenguaje hexadecimal)");
Serial.println(" (Password por defecto: 00 00 00 00)");
}
}
}

```

```

    p2=4; //La variable p2 se inicializa en 4, esta variable controla la posicion en el arreglo comand_rfid1, y desde la
    posicion 4 de este arreglo comienza el password de la tag
    x++; //Aumenta en uno la variable x y continua con el siguiente proceso
}

if(Serial.available()){

    comand_rfid1[2]=0x00; //Se otorga el valor 0x00 en la posicion 2 del arreglo comand_rfid1
    comand_rfid1[3]=0x02; //Se otorga el valor 0x02 en la posicion 3 del arreglo comand_rfid1

    while(x<13){
        //Lee password actual de la tag ingresado por la pc
        while(Serial.available()){
            delay(250); //Espera 1/4 de segundo
            coman_leer1[p1] = Serial.read(); //Lee caracteres escritas en la PC y las guarda en el arreglo coman_leer1, el
            cual la posicion en el cual se guardara el caracter en el arreglo esta definida por p1
            var22=coman_leer1[p1]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo
            coman_leer1, en la posicion definida por la variable p1
            Serial.write(coman_leer1[p1]); //Imprime en la pantalla de la PC el caracter guardado por el arreglo
            coman_leer1 en la posicion definida por la variable p1
            y++; //Aumenta en uno la variable y
            lim1++; //Aumenta en uno la variable lim1
            trans_hex(); //Se llama a la funcion para tranformar los caracteres ingresados por la PC en byte de dos numeros
            hex
            p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer1
        }
        x++; //Aumenta en uno la variable x, y al llegar a la cantidad de 13 se procede a continuar con el siguiente
        proceso
    }

    if(lim1>8){
        Serial.println("\n\r\n\r ERROR: Password tiene que ser MAXIMO de 8 digitos");
        cont1=8000; //Se otorga el valor de 8000 a la vriable para que ya no continue con los procesos y reinicie la
        funcion
        var1=5; //La variable var1 se le da el valor de 5 para que ya no comtinue con los procesos relacionados a esta
        variable
        lim2=8; //La variable lim2 se le da el valor de 8 para que ya no continue con los procesos relacionados a esta
        variable
        x=50; //La variable x se le da el valor de 50 para que ya no comtinue con los procesos y termine
    }
    //Serial.println(); //Imprime un ENTER en la pantalla de la PC

    if(x==13){
        hex_impar(); //Se llama a la funcion cuando los numero ingresados de la PC son impares
    }

    if(x==13){
        if(y==1){
            p2=p2+1; //Se incrementa en uno la variable p2 para asi agregar uno al conteo de los byte del arreglo
            comand_rfid1 cuando los caracteres ingresados por la PC son pares (los del nuevo password)
        }
    }

    if(x==13){
        //Serial.println(p2); //Imprime el valor que posee la variable p2
        p2=p2-1; //Se disminuye en uno la variable p2 para poder rellenar con 00 las casillas del arreglo que contiene el
        password,la caul es comand_rfid1
        while(p2<8){
            comand_rfid1[p2]=0x00; //Rellena las restantes posiciones del arreglo comand_rfid1 con el numero 0x00
            p2++; //Aumenta en uno la variable p2 para avanzar en la posicion del arreglo comand_rfid1
        }
    }
}

```

```

}

if(x==13){
  Serial.println("\n\r\n\r\n\rINGRESE PASSWORD NUEVO: ");
  Serial.println(" (En lenguaje hexadecimal)");
  Serial.println(" (El password tiene que ser maximo de 8 digitos)");
  y=0; //Se inicializa la variable y para comenzar a formar los bytes de dos numeros hex
  z1=0x00; //Se inicializa la variable z1 para que comience a contar los bytes del arreglo comand_rf1d1
  x++; //Aumenta en uno la variable x, y al llegar a la cantidad de 13 se procede a continuar con el siguiente proceso
}

delay(1000); //Espera 1 segundo

if(Serial.available()){

  if(x==14){
    //Serial.println(); //Imprime un ENTER en la pantalla de la PC
    if(p2==8){
      p2=p2+1; //Aumenta en uno la variable p2, se salta una posicion en el arreglo comand_rf1d1
    }
    if(p2<8){
      p2=p2+2; //Aumenta en dos la variable p2, se salta dos posicion en el arreglo comand_rf1d1
    }
    lim2=0; //Se inicializa la variable lim2 en 0 para que comience a contar los 8 caracteres que tiene que contar el
nuevo password
  }

  while(x<26){
    //Lee el nuevo password de la tag ingresado por la pc
    while(Serial.available()){
      delay(250); //Espera 1/4 de segundo
      coman_leer1[p1] = Serial.read(); //Lee caracteres escritas en la PC y las guarda en el arreglo coman_leer1, el
cual la posicion en el cual se guardara el caracter en el arreglo esta definida por p1
      var22=coman_leer1[p1]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo
coman_leer1, en la posicion definida por la variable p1
      Serial.write(coman_leer1[p1]); //Se imprime o escribe en la pantalla de la PC el caracter guardado por el arreglo
coman_leer1 en la posicion definida por la variable p1
      y++; //Aumenta en uno la variable y
      lim2++; //Aumenta en uno la variable lim2
      trans_hex(); //Se llama a la funcion para tranformar los caracteres ingresados por la PC en byte de dos numeros
hex
      p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer1
    }
    x++; //Aumenta en uno la variable x para poder seguir con el siguiente proceso hasta que llegue al valor de 26
  }

  if(x==26){
    hex_impar();
  }
  Serial.println(); //Imprime un ENTER en la pantalla de la PC
} //Fin del segundo if de leer del teclado

if(lim2>8){
  Serial.println("\n\r\n\r ERROR: Password tiene que ser MAXIMO de 8 digitos");
  cont1=8000; //Se otorga el valor de 8000 a la vriable para que ya no continue con los procesos y reinicie la
funcion
  var1=5; //La variable var1 se le da el valor de 5 para que ya no continue con los procesos relacionados a esta
variable
  x=50; //La variable x se le da el valor de 50 para que ya no continue con los procesos y termine
}

```

```

    if(x==26){
        if(y==0){
            z1=z1+0x01; //Se incrementa en uno la variable z1 para asi agregar uno al conteo de los byte del arreglo
comand_rfid1 cuando los caracteres ingresados por la PC son pares (los del nuevo password)
        }
        if(y==1){
            z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la
orden de escribir una tag
            p2=p2+1; //Se incrementa en uno la variable p2 para asi agregar uno al conteo de los byte del arreglo
comand_rfid1 cuando los caracteres ingresados por la PC son pares (los del nuevo password)
        }
    }

    if(x==26){
        p2=p2-1; //Se disminuye en uno la variable p2 para poder rellenar con 00 las casillas del arreglo que contiene el
password,la caul es comand_rfid1
        while(p2<13){
            comand_rfid1[p2]=0x00; //Rellena las restantes posiciones del arreglo comand_rfid1 con el numero 0x00
            p2++; //Aumenta en uno la variable p2 para avanzar en la posicion del arreglo comand_rfid1
            z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la
orden de escribir una tag
        }
    }

    if(x==26){
        comand_rfid1[1]=z1 + 0x08; //Se le suma el valor de la variable 0x08 a la variable z1, y se guarda en la posicion 1
del arreglo comand_rfid1, en esa posicion se debe colocar el tamano de todo el arreglo de byte que se le enviara al
modulo RFID en bytes
        comand_rfid1[8]=z1/0x02; //Se le asigna el valor de la variable z1 dividida en 2 a la posicion del valor de la
variable co1 del arreglo comand_rfid1, en esa posicion se debe colocar el tamano de todo el nuevo password de la tag, en
palabras (palabra = dos byte), en el arreglo que se le enviara al modulo RFID en bytes
    }

    if(y==1){
        p2++; //Aumenta en uno la variable p2 para saltar una posicion en el arreglo comand_rfid1
    }

    if(p2==14){
        p2=p2-1; //Disminuye en uno la variable p2 para poder escribir un arreglo de 13 caracteres en el modulo rfid
    }
    if(p2==15){
        p2=p2-2; //Disminuye en dos la variable p2 para poder escribir un arreglo de 13 caracteres en el modulo rfid
    }

    if(x==26){
        //Serial.println(p2); //Imprime el valor que posee la variable p2
        Serial1.write(comand_rfid1, p2); //Se imprime todo el arreglo comand_rfid1 en el modulo RFID, el cual da la
orden de cambiar el password de una tag especifica
        //for(int i=0; i<(p2); i++){
        //Serial.println(comand_rfid1[i], HEX); //Imprime todo el arreglo comand_rfid1 en la pantalla de la PC en
formato hex
        //}
    }

} //Fin del if leer del teclado

//Leyendo la respuesta del modulo RFID
if(Serial1.available()){

    cont1=0; //Se inicializa en 0 la variable cont1, la cual permite dividir el arreglo de la respuesta del modulo RFID
    Serial.println(); //Imprime un ENTER en la pantalla de la PC

```

```

while(cont1<2){
  if (Serial1.available( )){
    delay(250); //Espera 1/4 de segundo
    var1 = Serial1.read(); //Guarda los bytes leidos del modulo RFID en la variable var1
    //Serial.print(var1, HEX); //Imprime en la pantalla de la PC en formato HEX los bytes guardados en la variable
var1
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso hasta llegar al valor de 2
  }
}
Serial.println(); //Imprime un ENTER en la pantalla de la PC

//MUESTRA SI SE ESCRIBIO CORRECTAMENTE EL NUEVO PASSWORD EN LA ETIQUETA O NO
if(cont1<3){
  if (Serial1.available( )){
    var1 = Serial1.read(); //Guarda el tercer byte leido del modulo RFID en la variable var1, el cual muestra si el
password de la tag fue cambiado correctamente o no
    //Serial.println(var1, HEX); //Imprime el tercer byte en la pantalla de la PC en formato HEX
    //Si el tercer byte de respuesta del modulo RFID su valor es 0 entonces la tag se le cambio el password
correctamente, si su valor es 9 la tag no fue encontrada para escribir, si el valor es 2 existe un error en el password actual
ingresado y si es diferente de 0 la tag no se cambio el password correctamente
    if(var1==0){
      Serial.println("Se cambio el password de la etiqueta CORRECTAMENTE.");
    }else if(var1==9){
      Serial.println("\n\r Error al cambiar el password de la etiqueta: Etiqueta no encontrada");
    }else if(var1==2){
      Serial.println("\n\r Error al cambiar el password de la etiqueta: Error en password ");
    }else if(var1!=0){
      Serial.println("\n\r Error al cambiar el password de la etiqueta.");
    }
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
  }
}

while(cont1<4){
  if (Serial1.available( )){
    delay(250); //Espera 1/4 de segundo
    var1 = Serial1.read(); //Guarda el cuarto byte de la respuesta del modulo RFID el cual dice cuantas palabras
(palabra=dos byte) fueron escritas correctamente
    //Serial.print(var1, HEX); //Imprime en la pantalla de la PC en formato HEX el byte guardado en la variable var1
    cont1++; //Aumenta en uno la variable cont1 y continua con el siguiente proceso
  }
}

} //Fin de leer del modulo RFID

} //Fin de while si var1 = 5
if(cont1!=8000 || cont1==8000){
  if(cont1==4 || cont1==8000){
    Serial.println("\n\r \r\n \r\n >>> Precione 5 si quiere cambiar la contrasena a una etiqueta de nuevo <<< \r\n");
    Serial.println(" >>> Precione w para salir de la funcion <<< \r\n \r\n");
    cont1++; //Aumenta en uno la variable caont1 para imprimir una sola ves los mansajes anteriores
  }
}
} //Fin de primer while
} //Fin de coman_passwtag

/*Esta funcion le permite al usuario modificar la frecuencia de operacion del modulo RFID (un sistema RFID opera en el
rango de 810,000kHz a 990,000kHz, por ende la funcion opera en el mismo rango),
al usuario se le otrogan dos opciones, la primera es Desactivar el modo salto del modulo esto quiere decir hacer trabajar al
modulo en una frecuencia fija, la otra opcion es Activar el modo salto del

```

```

modulo, esto es permitirle al modulo operar en diferentes frecuencias de operacion o cada vez que el modulo realice una
operacion este lo ara saltando en diferentes frecuencias (la frecuencia de inicio
es la ingresada en la opcion Desactivar modo salto y cada salto de frecuencia que realice el modulo es cada frecuencia
ingresada en la opcion Activar modo salto)*/
void coman_freq(){

while (mx==1){

x=0; //Se inicializa la variable x para inicializar los proceso de la funcion

if (var1=='6'){
comand_rfid1[0]=0x41; //Se otorga el valor de 0x41 en la posicion 0 del arreglo comand_rfid1, es el inicio del
comando de cabio de frecuencia
comand_rfid1[1]=0x08; //Se otorga el valor de 0x08 en la posicion 1 del arreglo comand_rfid1, muestra la longitud
de todo el arreglo de la orden de cambio de frecuencia
}

delay (1000); //Espera 1 segundo

p1=0; //La variable p1 se inicializa en 0, esta variable controla la posicion en el arreglo coman_leer1, en el cual se
guarda los valores de la combercion a byte de dos numeros en formato hex
p2=3; //La variable p2 se inicializa en 3, esta variable controla la posicion en el arreglo comand_rfid1, y desde la
posicion 3 de este arreglo comienza el valor de la frecuencia a la cual se cambiara
x=2; //La variable x se inicializa en 2, esta variable particiona el arreglo de caracteres leidos desde la PC o controla el
orden en que es leido los caracteres escritos desde la PC
y=0; //La variable y se inicializa en 0, se utiliza para formar un byte de dos numeros hex formados con dos caracteres
trasformados en numeros hex
lim1=0; //La variable lim1 se inicializa en 0, se utiliza para conocer si se ingreso o digitado 6 caracteres para la
frecuencia del modulo y si no es asi el proceso termina y sale del programa
pp=0; //La variable pp se inicializa en 0, esta variable controla la posicion de los arreglos Freqx y Freq2, en el cual se
guarda los caracteres ingresados por la PC para formar la frecuencia

while(var1=='6'){

if(x==2){
Serial.println("\n\r \n\r \n\r \n\r"); //Imprime en la pantalla de la PC 5 ENTER
Serial.println("
+++++++");
Serial.println(" ++++++ <<<<<<<< CAMBIAR FRECUENCIA DE OPERACION >>>>>>>>
+++++++");
Serial.println("
+++++++");
Serial.println("\n\r \n\r"); //Imprime en la pantalla de la PC 3 ENTER
Serial.println("\n\r 1. Activar modo salto (Agregar la frecuencia a la lista de frecuencias.");
Serial.println("      NOTA: Al utilizar este comando el modulo ara un escaneo saltando de frecuencia en
frecuencia");
Serial.println("      establecidas cada una, cada vez que se ingrese una frecuencia en al opcion 1");
Serial.println("      (la frecuencia inicial es la ingresada en la opcion 2).");
//delay(1000); //Espera 1 segundo
Serial.println("\n\r 2. Desactivar modo salto (Limpiar lista de frecuencia).");
Serial.println("      NOTA: Al utilizar este comando el modulo escaneara a una frecuencia fija.");
//delay(1000); //Espera 1 segundo
Serial.println("\n\r 3. Salir.");
x++; //La variable aumenta en uno para avanzar con el siguiente proceso
}

if(Serial.available()){

if(x==3){
while(Serial.available()){
var22=Serial.read(); //Lee el caracter ingresado por la PC y lo guarda en la variable var22

```

```

Serial.print("\n\r\n\r Opcion: ");
Serial.write(var22); //Imprime en la pantalla de la PC el valor que contiene la variable var22
if(var22=='1'){
    comand_rfid1[2]=0x04; //Se le otorga el valor de 0x04 en la pocision 2 del arreglo comand_rfid1 el cual activa
el modo salto y/o agregar una frecuencia a la lista de salto
    x++; //La variable aumenta en uno para avanzar con el siguiente proceso
}
if(var22=='2'){
    comand_rfid1[2]=0x08; //Se otorga el valor de 0x04 en la posicion 2 del arreglo comand_rfid1 el cual desactiva
el modo salto y ase que el modulo funcione a una frecuencia fija
    x++; //La variable aumenta en uno para avanzar con el siguiente proceso
}
if(var22=='3'){
    var1=7; //Se otorga el valor de 7 a la variable var1 para que no continue con los siguientes procesos y termine
todo
    x=100; //Se otorga el valor de 100 a la variable x para que no continue con los siguientes procesos y termine
todo
    mx=2; //Se le otorga el valor de 2 a la variable mx para salir completamente de la funcion actual
}
if(var22!='1' && var22!='2' && var22!='3'){
    Serial.println("\n\r\n\r OPCION ERRONEA");
    Serial.println(" Elija una opcion en el menu.\n\r");
    x=2;
    //var1=7;
}
}
} //if x7=3

if(x==4){
    Serial.println("\n\r\n\rIngrese la frecuencia en kHz: ");
    Serial.println(" (La frecuencia debe de estar en el rango de 810,000kHz a 990,000kHz)");
    lim1=0;
    pp=0;
    x++; //La variable aumenta en uno para avanzar con el siguiente proceso
    //p2=p2+2;
}

if(Serial.available()){
    while(x<13){
        //Lee la frecuencia ingresada por la PC
        while(Serial.available()){
            delay(250); //Espera 1/4 de segundo
            Freqx[pp] = Serial.read(); //Lee caracteres escritos en la PC y las guarda en el arreglo Freqx, en el cual se
guardara el caracter en el arreglo esta definida por pp
            var22=Freqx[pp]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo Freqx, en
la posicion definida por la variable pp
            Serial.write(Freqx[pp]); //Se imprime en la pantalla de la PC el caracter guardado por el arreglo Freqx en la
posicion definida por la variable pp
            lim1++; //Aumenta en uno la variable lim1 para tener la cantidad de caracteres que componen la frecuecnia y
tener el control de que sean 6 caracteres

            //Este proceso se activa cuando se ingresa un caracter que no es un numero decimal, y si se activa ya no se
efectuan mas procesos y se termina toda la funcion
            if(Freqx[pp]!='0' && Freqx[pp]!='1' && Freqx[pp]!='2' && Freqx[pp]!='3' && Freqx[pp]!='4'){
                if(Freqx[pp]!='5' && Freqx[pp]!='6' && Freqx[pp]!='7' && Freqx[pp]!='8' && Freqx[pp]!='9'){
                    Serial.println("\n\r\n\r ERROR: La frecuencia debe de ser en numeros decimales.");
                    while(Serial.available()){
                        delay(250); //Espera 1/4 de segundo
                        Freq2[pp] = Serial.read(); //Se guarda los caracteres ingresados por la PC en el arreglo Freq2 en la
posicion dada por la variable pp, para que ya no sean mostrado en pantalla de la PC
                        pp++; //Aumenta en uno la variable para avanzar en la pocision del arreglo Freq2
                    }
                }
            }
        }
    }
}

```

```

        lim1++; //Aumenta en uno la variable para contar la cantidad de caracteres ingresados por la PC
    }
    x=20; //Se otorga el valor de 2 a la variable x para que ya no prosiga con los de mas procesos y termine toda
la funcion
    var1='6'; //Se otorga el valor de 6 a la variable var1 para que ya no prosiga con los de mas procesos y
termine toda la funcion
    }
    }
    if(pp==0){
        //Este proceso se activa cuando el primer caracter que se ingresa por la PC no es un 8 y/o un 9 (ya que el
modulo RFID trabaja en la vanda de 820MHz a 990MHz), y si se activa ya no continuara los demas procedimientos y
termina toda la funcion
        if(Freqx[0] != '8' && Freqx[0] != '9'){
            Serial.println("\n\r\n\r ERROR: La frecuencia debe de estar en el rango de 810,000kHz a 990,000kHz.");
            while(Serial.available()){
                delay(250); //Espera 1/4 de segundo
                Freq2[pp] = Serial.read(); //Se guarda los caracteres ingresados por la PC en el arreglo Freq2 en la
pocision dada por la variable pp, para que ya no sean mostrado en pantalla de la PC
                pp++; //Aumenta en uno la variable para avanzar en la pocision del arreglo Freq2
                lim1++; //Aumenta en uno la variable para contar la cantidad de caracteres ingresados por la PC
            }
            x=20; //Se otorga el valor de 2 a la variable x para que ya no prosiga con los de mas procesos y termine toda
la funcion
            var1='6'; //Se otorga el valor de 6 a la variable var1 para que ya no prosiga con los de mas procesos y
termine toda la funcion
        }
    }
    if(Freqx[pp]=='0'){
        Freqx[pp]=0; //Se otorga el valor de 0 a la pocision que da la variable pp en el arreglo Freqx cuando el
caracter ingresado por la PC es 0
        num_freq(); //Se llama a ala funcion para poder formar la variable que contenbra la frecuencia
    }
    if(Freqx[pp]=='1'){
        Freqx[pp]=1; //Se otorga el valor de 1 a la pocision que da la variable pp en el arreglo Freqx cuando el
caracter ingresado por la PC es 1
        num_freq(); //Se llama a ala funcion para poder formar la variable que contenbra la frecuencia
    }
    if(Freqx[pp]=='2'){
        Freqx[pp]=2; //Se otorga el valor de 2 a la pocision que da la variable pp en el arreglo Freqx cuando el
caracter ingresado por la PC es 2
        num_freq(); //Se llama a ala funcion para poder formar la variable que contenbra la frecuencia
    }
    if(Freqx[pp]=='3'){
        Freqx[pp]=3; //Se otorga el valor de 3 a la pocision que da la variable pp en el arreglo Freqx cuando el
caracter ingresado por la PC es 3
        num_freq(); //Se llama a ala funcion para poder formar la variable que contenbra la frecuencia
    }
    if(Freqx[pp]=='4'){
        Freqx[pp]=4; //Se otorga el valor de 4 a la pocision que da la variable pp en el arreglo Freqx cuando el
caracter ingresado por la PC es 4
        num_freq(); //Se llama a ala funcion para poder formar la variable que contenbra la frecuencia
    }
    if(Freqx[pp]=='5'){
        Freqx[pp]=5; //Se otorga el valor de 5 a la pocision que da la variable pp en el arreglo Freqx cuando el
caracter ingresado por la PC es 5
        num_freq(); //Se llama a ala funcion para poder formar la variable que contenbra la frecuencia
    }
    if(Freqx[pp]=='6'){
        Freqx[pp]=6; //Se otorga el valor de 6 a la pocision que da la variable pp en el arreglo Freqx cuando el
caracter ingresado por la PC es 6
        num_freq(); //Se llama a ala funcion para poder formar la variable que contenbra la frecuencia
    }

```

```

    }
    if(Freqx[pp]=='7'){
        Freqx[pp]=7; //Se otorga el valor de 7 a la posicion que da la variable pp en el arreglo Freqx cuando el
caracter ingresado por la PC es 7
        num_freq(); //Se llama a ala funcion para poder formar la variable que contenbra la frecuencia
    }
    if(Freqx[pp]=='8'){
        Freqx[pp]=8; //Se otorga el valor de 8 a la posicion que da la variable pp en el arreglo Freqx cuando el
caracter ingresado por la PC es 8
        num_freq(); //Se llama a ala funcion para poder formar la variable que contenbra la frecuencia
    }
    if(Freqx[pp]=='9'){
        Freqx[pp]=9; //Se otorga el valor de 9 a la posicion que da la variable pp en el arreglo Freqx cuando el
caracter ingresado por la PC es 9
        num_freq(); //Se llama a ala funcion para poder formar la variable que contenbra la frecuencia
    }
    pp++; //Aumenta en uno la variable pp para avanzar en la posicion del arreglo Freqx

}
x++; //Aumenta en uno la variable x para continuar con el siguiente proceso, asta que llegue al valr de 13
}
}

if(x==21){
    x=2;
}

//Este proceso se activa cuando no se ingresa 6 caracteres para formar la frecuencia, asiendo que toda la funcion
termine
if(x==13){
    if(lim1!=6){
        Serial.println("\n\r\n\r ERROR: La frecuencia debe de ser en kHz.");
        var1=6; //Se otorga el valor de 6 a la variable var1 para que ya no prosiga con los de mas procesos y termine
toda la funcion
        x=2; //Se otorga el valor de 2 a la variable x para que ya no prosiga con los de mas procesos y termine toda la
funcion
    }
}

if(x==13){
    for(int t=0; t<6; t++){
        Freq2[0]=Freqx[6]/16; //Se divide por 16 el valor que esta en el arreglo Freqx en la posicion 6 (el cual es la
frecuencia en decimal), y se guarda en el arreglo Freq2 posicion 0
        Freq2[1]=Freqx[6]%16; //Se divide po 16 el valor que esta en el arreglo Freqx en la posicion 6 (el cual es la
frecuencia en decimal), y se guarda en el arreglo Freq2 posicion 1 el resto de la division
        Freqx[6]=Freq2[0]; //Se otorga el valor del arreglo Freq2 posicion 0 al arreglo Freqx posicion 6
        y++; //Aumenta en uno la variable y para que continue con la siguiente division
        comand_leer1[p1]=Freq2[1]; //Se otorga el valor del arreglo Freq2 posicion 1 al arreglo comand_leer1 posicion que
da la variable p1
        trans_hex2(); //Se llama a la funcion para tranformar los caracteres ingresados por la PC en byte de dos numeros
hex
        p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo comand_leer3
    }
}

if(x==13){
    //Serial.println(p2); //Imprime en la pantalla de la PC el valor de la variable p2
    p2=p2+2; //Se le suma 2 a la variable p2
    comand_rfid1[6]=0xD8; //Se le otorga el valor 0xD8 a el arreglo comand_rfid1 en la posicion 6, el cual representa
el umbral del valor de RSSI, en este caso es 0xD8=40=-40dBm
    comand_rfid1[7]=0x01; //Se le otorga el valor 0x01 a el arreglo comand_rfid1 en la posicion 7, el cual representa

```

```

el numero del perfil al cual se le cambiara la frecuencia
}

if(x==13){
  Serial.println("\n\r"); //Imprime dos ENTER en la pantalla de la PC
  //Serial.println(p2); //Imprime en la pantalla de la PC el valor de la variable p2
  Serial1.write(comand_rfid1, p2); //Escribe todo el arreglo comand_rfid1 en el modulo RFID, el cual tiene la orden
de cambio de frecuencia
  //for(int i=0; i<(p2); i++){
  //Serial.print(comand_rfid1[i], HEX); //Imprime en la pantalla de la PC todo el arreglo comand_rfid1
  //}
}

} //Fin if leer del teclado

//Leyendo la respuesta del modulo RFID
if(Serial1.available()){

  cont1=0; //Inicializa la variable cont1 en 0 para poder particionar el arreglo de respuesta del modulo RFID y tener
un mejor presentacion
  Serial.println(); //Imprime un ENTER en la pantalla de la PC

  while(cont1<2){
    if (Serial1.available()){
      delay(250); //Espera 1/4 de segundo
      var22 = Serial1.read(); //Lee dos bytes del arreglo de respuesta del modulo RFID y los guarda en la variable
var22
      //Serial.print(var22, HEX); //Imprime en la pantalla de la PC los bytes que estan guardados en la variable var22
en formato hex
      cont1++; //Aumenta en uno la variable cont1 y asta que llegue al valor de 2 se pasara al siguiente proceso
    }
  }
  Serial.println(); //Imprime en la pantalla de la PC un ENTER

  //MUESTRA SI EL MODO SALTO A SIDO ACTIVADO O DESACTIVADO
  if(cont1<3){
    if (Serial1.available()){
      var22 = Serial1.read(); //Lee el tercer byte de respuesta del modulo RFID, el cual muestra si se ha activado el
modo salto
      //Serial.println(var22, HEX); //Imprime en la pantalla de la PC el valor de la variable var22 (el tercer byte) en
formato hex
      //Si el tercer byte es 0xFE el modo de salto esta desactivado y el modulo RFID operara en una frecuencia fija y si
el valor es 0xFC el modo salto a sido activado y/o se ha agregado una frecuencia a la lista de saltos
      //Si el valor es 2 existe un error a al cambiar la frecuencia y si el valor es distinto de 0 tambien existe un error al
cambiar la frecuencia
      if(var22==0xFE){
        Serial.println("\n\rModo salto desactivado.");
        Serial.println(" Frecuencia establecida correctamente");
      }else if(var22==0xFC){
        Serial.println("\n\rModo salto activado.");
        Serial.println(" Frecuencia establecida correctamente");
      }else if(var22==2){
        Serial.println("\n\r Error al cambiar la frecuencia de operacion: Error en password ");
      }else if(var22!=0){
        Serial.println("\n\r Error al cambiar la frecuencia de operacion.");
      }else if(var22==0){
        Serial.println("\n\r Error al cambiar la frecuencia de operacion.");
      }
    }
    cont1++; //Aumenta en uno la variable cont1 para que pueda pasar al siguiente proceso
  }
}
}

```

```

if(cont1<4){
  if (Serial1.available( )){
    delay(250); //Espera 1/4 de segundo
    var22 = Serial1.read(); //Lee el cuarto byte de la respuesta del modulo RFID y lo guarda en la variable var22
    //Serial.println(var22, HEX); //Imprime en la pantalla de la PC el valor de la variable var22 en formato hex
    cont1++; //Aumenta en uno la variable cont1 para que pueda pasar al siguiente proceso
  }
}

while(cont1<67){
  if (Serial1.available( )){
    //delay(250); //Espera 1/4 de segundo
    var22 = Serial1.read(); //Lee los 60 bytes de la respuesta del modulo RFID y los guarda en la variable var22 (son
numeros 0x00)
    //Serial.print(var22, HEX); //Imprime en la pantalla de la PC el valor de la variable var22 en formato hex
  }
  cont1++; //Aumenta en uno la variable cont1 para que pueda pasar al siguiente proceso
}

var1='6';
x=2;
} //Fin de leer del modulo RFID

} //Fin de while si var1 = 6
} //Fin de primer while
} //Fin de coman_freq()

/*Esta funcion divide en grupo de dos los caracteres ingresados por
la PC y los convierte en bytes de dos numeros en formato hexadecimal*/
void trans_hex20{
  //Convercion de caracteres a numeros hexadecimales
  if(y==1){
    if(coman_leer1[p1]==0){
      coman_leer1[p1]=0x00; //Combierte el numero 0 guardado en el arreglo coman_leer1 en la posicion p1 en el numero
0x00
      if(pp==1){
        comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
        p2++; //Aumenta en uno la variable p2 para poder avanzar en la posicion en el arreglo comand_rfid1
      }
    }
  }
  if(coman_leer1[p1]==1){
    coman_leer1[p1]=0x01; //Combierte el numero 1 guardado en el arreglo coman_leer1 en la posicion p1 en el numero
0x01
    if(pp==1){
      comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
      p2++; //Aumenta en uno la variable p2 para poder avanzar en la posicion en el arreglo comand_rfid1
    }
  }
  if(coman_leer1[p1]==2){
    coman_leer1[p1]=0x02; //Combierte el numero 2 guardado en el arreglo coman_leer1 en la posicion p1 en el numero
0x02
    if(pp==1){
      comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
      p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
    }
  }
  if(coman_leer1[p1]==3){

```

```

coman_leer1[p1]=0x03; //Combierte el numero 3 guardado en el arreglo coman_leer1 en la posicion p1 en el numero
0x03
if(pp==1){
comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
}
}
if(coman_leer1[p1]==4){
coman_leer1[p1]=0x04; //Combierte el numero 4 guardado en el arreglo coman_leer1 en la posicion p1 en el numero
0x04
if(pp==1){
comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
}
}
if(coman_leer1[p1]==5){
coman_leer1[p1]=0x05; //Combierte el numero 5 guardado en el arreglo coman_leer1 en la posicion p1 en el numero
0x05
if(pp==1){
comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
}
}
if(coman_leer1[p1]==6){
coman_leer1[p1]=0x06; //Combierte el numero 6 guardado en el arreglo coman_leer1 en la posicion p1 en el numero
0x06
if(pp==1){
comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
}
}
if(coman_leer1[p1]==7){
coman_leer1[p1]=0x07; //Combierte el numero 7 guardado en el arreglo coman_leer1 en la posicion p1 en el numero
0x07
if(pp==1){
comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
}
}
if(coman_leer1[p1]==8){
coman_leer1[p1]=0x08; //Combierte el numero 8 guardado en el arreglo coman_leer1 en la posicion p1 en el numero
0x08
if(pp==1){
comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
}
}
if(coman_leer1[p1]==9){
coman_leer1[p1]=0x09; //Combierte el numero 9 guardado en el arreglo coman_leer1 en la posicion p1 en el numero
0x09
if(pp==1){
comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
}
}
}

```

```

}
if(coman_leer1[p1]==10){
    coman_leer1[p1]=0x0A; //Combierte el numero 10 guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0x0A
    if(pp==1){
        comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
        p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
    }
}
if(coman_leer1[p1]==11){
    coman_leer1[p1]=0x0B; //Combierte el numero 11 guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0x0B
    if(pp==1){
        comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
        p2++; //Aumenta en uno la variable p2 para poder avanzar en la posicion en el arreglo comand_rfid1
    }
}
if(coman_leer1[p1]==12){
    coman_leer1[p1]=0x0C; //Combierte el numero 12 guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0x0C
    if(pp==1){
        comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
        p2++; //Aumenta en uno la variable p2 para poder avanzar en la posicion en el arreglo comand_rfid1
    }
}
if(coman_leer1[p1]==13){
    coman_leer1[p1]=0x0D; //Combierte el numero 13 guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0x0D
    if(pp==1){
        comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
        p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
    }
}
if(coman_leer1[p1]==14){
    coman_leer1[p1]=0x0E; //Combierte el numero 14 guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0x0E
    if(pp==1){
        comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
        p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
    }
}
if(coman_leer1[p1]==15){
    coman_leer1[p1]=0x0F; //Combierte el numero 15 guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0x0F
    if(pp==1){
        comand_rfid1[p2]=coman_leer1[p1]; //Se otorga el valor del arreglo coman_leer1 en la posicion que da la variable
p1 al arreglo comand_rfid1 posicion que da la variable p2
        p2++; //Aumenta en uno la variable p2 para poder avanzar en la pocicion en el arreglo comand_rfid1
    }
}
}

if(y==2){
    if(coman_leer1[p1]==0){
        comand_rfid1[p2] = coman_leer1[p1-1] + 0x00; //Sele suma 0x00 al numero guardado en el arreglo coman_leer1 en
la posicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos

```

```

numeros en hex
}
if(coman_leer1[p1]==1){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x10; //Sele suma 0x10 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==2){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x20; //Sele suma 0x20 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==3){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x30; //Sele suma 0x30 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==4){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x40; //Sele suma 0x40 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==5){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x50; //Sele suma 0x50 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==6){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x60; //Sele suma 0x60 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==7){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x70; //Sele suma 0x70 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==8){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x80; //Sele suma 0x80 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==9){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x90; //Sele suma 0x90 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==10){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0xA0; //Sele suma 0xA0 al numero guardado en el arreglo coman_leer1
en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==11){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0xB0; //Sele suma 0xB0 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]==12){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0xC0; //Sele suma 0xC0 al numero guardado en el arreglo coman_leer1 en
la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}

```

```

    }
    if(coman_leer1[p1]==13){
        comand_rfid1[p2] = coman_leer1[p1-1] + 0xD0; //Sele suma 0xD0 al numero guardado en el arreglo coman_leer1
        en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
        numeros en hex
    }
    if(coman_leer1[p1]==14){
        comand_rfid1[p2] = coman_leer1[p1-1] + 0xE0; //Sele suma 0xE0 al numero guardado en el arreglo coman_leer1 en
        la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
        numeros en hex
    }
    if(coman_leer1[p1]==15){
        comand_rfid1[p2] = coman_leer1[p1-1] + 0xF0; //Sele suma 0xF0 al numero guardado en el arreglo coman_leer1 en
        la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
        numeros en hex
    }
    p2++; //Aumenta en uno la variable p2 para avanzar en la posscion del arreglo comand_rfid1
}

if(y==2){
    y=0; //Se inicializa de nuevo la variable y, para asi formar de nuevo un byte de dos numeros Hex
}
} //Fin de trans_hex2()

/*Esta funcion escoje que funcion transformara los caracteres que componen la potencia de la antena en una sola
variable, esta funcion
escoje a las demas funciones o procesos viendo cuantos caracteres se ha ingresado desde la PC para componer la
potencia de la antena*/
void selec_func(){
    if(pp==1){
        power1[3]=power1[pp1]; //Se otorga el valor que esta en el arreglo power1 en la posicion que da la variable pp1 y se
        otorga en la posicion 3 del mismo arreglo
        pp1++; //Aumenta en uno la variable pp1 para poder avanzar en las posiciones del arreglo coman_leer1
    }
    if(pp==2){
        num_power2(); //Se llama a la funcion cuando se ha ingresado 2 caracteres para componer el valor de la potencia de
        la antena, y para unirlos en una sola variable
        pp1++; //Aumenta en uno la variable pp1 para poder avanzar en las posiciones del arreglo coman_leer1
    }
    if(pp==3){
        num_power3(); //Se llama a la funcion cuando se ha ingresado 3 caracteres para componer el valor de la potencia de
        la antena, y para unirlos en una sola variable
        pp1++; //Aumenta en uno la variable pp1 para poder avanzar en las posiciones del arreglo coman_leer1
    }
} //Fin de selec_func()

/*Esta funcion transforma todos los caracteres que componen la potencia de la antena en una sola variable que se puede
manipular mas facilmente,
esta funcion se activa cuando se ingresar 2 variables para componer el valor de potencia*/
void num_power2(){
    if(pp1==0){
        power1[0]=power1[0]*10; //Agrega 1 ceros a la derecha del numero que esta en el arreglo power1 en la posicion 0,
        multiplicando por 10 dicho numero
        //Serial.println(power1[0]); //Imprime en la pantalla de la PC el resultado de multiplicar 10 al numero en el arreglo
        power1 en la posicion 0
    }
    if(pp1==1){
        //Suma todos los numeros del arreglo power1 de las posiciones 0 y 1, y lo guarda en la posicion 3 de dicho arreglo,
        //conteniendo asi todo el valor de toda la potencia de la antena en la posicion 3 del arreglo
        power1[3]=power1[0]+power1[1];
        //Serial.println("\n"); //Imprime 2 ENTER en la pantalla de la PC
    }
}

```

```

//Serial.println(power1[3]); //Imprime en la pantalla de la PC el valor que contiene el arreglo power1 en la posicion 3,
el cual contiene todo el valor de la potencia de la antena
}
} //Fin de num_power2

/*Esta funcion transforma todos los caracteres que componen la potencia de la antena en una sola variable que se puede
manipular mas facilmente,
esta funcion se activa cuando se ingresan 3 variables para componer el valor de potencia*/
void num_power3(){
  if(pp1==0){
    power1[0]=power1[0]*100; //Agrega 2 ceros a la derecha del numero que esta en el arreglo power1 en la posicion 0,
multiplicando por 100 dicho numero
    //Serial.println(power1[0]); //Imprime en la pantalla de la PC el resultado de multiplicar 100 al numero en el arreglo
power1 en la posicion 0
  }
  if(pp1==1){
    power1[1]=power1[1]*10; //Agrega 1 ceros a la derecha del numero que esta en el arreglo power1 en la posicion 1,
multiplicando por 10 dicho numero
    //Serial.println(power1[1]); //Imprime en la pantalla de la PC el resultado de multiplicar 10000 al numero en el arreglo
power1 en la posicion 1
  }
  if(pp1==2){
    //Suma todos los numeros del arreglo power1 de las posiciones 0, 1 y 2, y lo guarda en la posicion 3 de dicho arreglo,
//conteniendo asi todo el valor de toda la potencia de la antena en la posicion 3 del arreglo
    power1[3]=power1[0]+power1[1]+power1[2];
    //Serial.println("\n\r"); //Imprime 2 ENTER en la pantalla de la PC
    //Serial.println(power1[3]); //Imprime en la pantalla de la PC el valor que contiene el arreglo power1 en la posicion 3,
el cual contiene todo el valor de la frecuencia
  }
} //Fin de num_power3

/*Esta funcion activa la forma en que se visualizara los datos o bytes de respuesta del modulo RFID o de los datos del ID
de la etiqueta*/
void visualizar(){
  //MENU DE VISUALIZACION DE DATOS
  if(hy==0){
    hz=0; //Se inicializa en 0 la variable para que espere asta que se obtenga un valor de la PC
    Serial.println("Escoja la forma en que se quiere visualizar los datos:");
    Serial.println(" h: Formato hexadecimal.");
    Serial.println(" q: Formato no hexadecimal. \n\r");
    hy++; //Aumenta en uno la variable hy para poder continuar con el siguiente proceso
  }

  while(hy==1){
    if(Serial.available()){
      hz=Serial.read(); //Guarda en la variable var1 los leido de la PC
      if((hz!='q' && hz!='Q') && (hz!='h' && hz!='H')){
        Serial.println(" ERROR: Opcion incorrecta! \n\r");
        //var1=0; //La variable se iguala a 0 para que ya no continúe con los procesos y comience todo de nuevo
        hx=5; //La variable se iguala a 5 para que ya no continúe con los procesos y comience todo de nuevo
        hy=0; //La variable se iguala a 5 para que ya no continúe con los procesos y comience todo de nuevo
        cont1=8000; //La variable se iguala a 8000 para que ya no continúe con los procesos y comience todo de nuevo
      }
      Serial.print(" Opcion: ");
      Serial.write(hz); //Imprime en la pantalla de la PC los guardado en la variable var1
      Serial.println("\n\r \n\r"); //Imprime en la pantalla de la PC 3 ENTER
    }
    if(hz=='h' || hz=='H'){
      hx=1; //Con hx=1 se visualizara el ID de la tag en formato HEX
      hy++; //Aumenta en uno la variable hy saliendo del menu de visualizacion de datos
      cont1=0; //Se inicializa la variable en 0 para poder particionar el arreglo de la respuesta del modulo RFID, para una

```

```

mejor presentacion
}
if(hz=='q' || hz=='Q'){
  hx=2; //Con hx=1 se visualizara el ID de la tag en formato ascii
  hy++; //Aumenta en uno la variable hy saliendo del menu de visualizacion de datos
  cont1=0; //Se inicializa la variable en 0 para poder particionar el arreglo de la respuesta del modulo RFID, para una
mejor presentacion
}
}
}

/* Esta funcion complementa los numeros del ID de la tag, imprimiendo o escribiendo un
cero a la izquierda del numero del ID, si lo requiere, para visualizar el numero completo */
void completar(int var22){
  if(var22==0x00){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero 0 del ID
  }
  if(var22==0x01){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero 1 del ID
  }
  if(var22==0x02){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero 2 del ID
  }
  if(var22==0x03){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero 3 del ID
  }
  if(var22==0x04){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero 4 del ID
  }
  if(var22==0x05){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero 5 del ID
  }
  if(var22==0x06){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero 6 del ID
  }
  if(var22==0x07){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero 7 del ID
  }
  if(var22==0x08){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero 8 del ID
  }
  if(var22==0x09){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero 9 del ID
  }
  if(var22==0x0A){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero A del ID
  }
  if(var22==0x0B){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero B del ID
  }
  if(var22==0x0C){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero C del ID
  }
  if(var22==0x0D){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero D del ID
  }
  if(var22==0x0E){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero E del ID
  }
  if(var22==0x0F){
    Serial.print(0x00, HEX); //Imprime en pantalla un cero a la izquierda del numero F del ID
  }
}

```

```

}
} //Fin de completar(int var22)

/*Esta funcion divide en grupo de dos los caracteres ingresados por
la PC y los convierte en bytes de dos numeros en formato hexadecimal*/
void trans_hex(){

  if(y==1){
    //Al ingresar un caracter que no es un numero HEX el proceso lo detecta y no deja que continúe con los demas
    procesos y todo comienza de nuevo
    if(coman_leer1[p1]!='0' && coman_leer1[p1]!='1' && coman_leer1[p1]!='2' && coman_leer1[p1]!='3' &&
    coman_leer1[p1]!='4' && coman_leer1[p1]!='5' && coman_leer1[p1]!='6' && coman_leer1[p1]!='7' &&
    coman_leer1[p1]!='8' && coman_leer1[p1]!='9' && coman_leer1[p1]!='a' && coman_leer1[p1]!='A'){
      if(coman_leer1[p1]!='b' && coman_leer1[p1]!='B' && coman_leer1[p1]!='c' && coman_leer1[p1]!='C' &&
      coman_leer1[p1]!='d' && coman_leer1[p1]!='D' && coman_leer1[p1]!='e' && coman_leer1[p1]!='E' &&
      coman_leer1[p1]!='f' && coman_leer1[p1]!='F'){
        Serial.println("\n\r\n\rERROR: Tiene que ser en formato HEXADECIMAL!\n\r");
        while(Serial.available()){
          delay(250); //Espera 1/4 de segundo
          coman_leer1[p1] = Serial.read(); //Lee los caracteres escritos en la PC y las guarda en el arreglo coman_leer1, la
          posicion en que se guarda en el arreglo es definido por la variable p1
          var22=coman_leer1[p1]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo
          coman_leer1, en la posicion definida por la variable p1
          //Serial.write(coman_leer1[p1]); //Se imprime en la pantalla de la PC el caracter guardado por el arreglo
          coman_leer1 en la posicion definida por la variable p1
          y++; //Aumenta en uno la variable y, para activar el primer proceso de crear un byte de dos numeros hex
          p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer1
        }
        var1=0; //La variable se iguala a 0 para que ya no continúe con los procesos y comience todo de nuevo
        x=100; //La variable se iguala a 100 para que ya no continúe con los procesos y comience todo de nuevo
        cont1=8000; //La variable se iguala a 8000 para que ya no continúe con los procesos y comience todo de nuevo
      }
    }
    if(coman_leer1[p1]=='0'){
      coman_leer1[p1]=0x00; //Combierte el caracter 0 guardado en el arreglo coman_leer1 en la posicion p1 en el
      numero 0x00
    }
    if(coman_leer1[p1]=='1'){
      coman_leer1[p1]=0x10; //Combierte el caracter 1 guardado en el arreglo coman_leer1 en la posicion p1 en el
      numero 0x10
    }
    if(coman_leer1[p1]=='2'){
      coman_leer1[p1]=0x20; //Combierte el caracter 2 guardado en el arreglo coman_leer1 en la posicion p1 en el
      numero 0x20
    }
    if(coman_leer1[p1]=='3'){
      coman_leer1[p1]=0x30; //Combierte el caracter 3 guardado en el arreglo coman_leer1 en la posicion p1 en el
      numero 0x30
    }
    if(coman_leer1[p1]=='4'){
      coman_leer1[p1]=0x40; //Combierte el caracter 4 guardado en el arreglo coman_leer1 en la posicion p1 en el
      numero 0x40
    }
    if(coman_leer1[p1]=='5'){
      coman_leer1[p1]=0x50; //Combierte el caracter 5 guardado en el arreglo coman_leer1 en la posicion p1 en el
      numero 0x50
    }
    if(coman_leer1[p1]=='6'){
      coman_leer1[p1]=0x60; //Combierte el caracter 6 guardado en el arreglo coman_leer1 en la posicion p1 en el
      numero 0x60
    }
  }
}

```

```

    if(coman_leer1[p1]=='7'){
        coman_leer1[p1]=0x70; //Combierte el caracter 7 guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0x70
    }
    if(coman_leer1[p1]=='8'){
        coman_leer1[p1]=0x80; //Combierte el caracter 8 guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0x80
    }
    if(coman_leer1[p1]=='9'){
        coman_leer1[p1]=0x90; //Combierte el caracter 9 guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0x90
    }
    if(coman_leer1[p1]=='a' || coman_leer1[p1]=='A'){
        coman_leer1[p1]=0xA0; //Combierte el caracter a guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0xA0
    }
    if(coman_leer1[p1]=='b' || coman_leer1[p1]=='B'){
        coman_leer1[p1]=0xB0; //Combierte el caracter b guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0xB0
    }
    if(coman_leer1[p1]=='c' || coman_leer1[p1]=='C'){
        coman_leer1[p1]=0xC0; //Combierte el caracter c guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0xC0
    }
    if(coman_leer1[p1]=='d' || coman_leer1[p1]=='D'){
        coman_leer1[p1]=0xD0; //Combierte el caracter d guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0xD0
    }
    if(coman_leer1[p1]=='e' || coman_leer1[p1]=='E'){
        coman_leer1[p1]=0xE0; //Combierte el caracter e guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0xE0
    }
    if(coman_leer1[p1]=='f' || coman_leer1[p1]=='F'){
        coman_leer1[p1]=0xF0; //Combierte el caracter f guardado en el arreglo coman_leer1 en la posicion p1 en el
numero 0xF0
    }
}

if(y==2){
    //Al ingresar un caracter que no es un numero HEX el proceso lo detecta y no deja que continue con los demas
procesos y todo comienza de nuevo
    if(coman_leer1[p1]!='0' && coman_leer1[p1]!='1' && coman_leer1[p1]!='2' && coman_leer1[p1]!='3' &&
coman_leer1[p1]!='4' && coman_leer1[p1]!='5' && coman_leer1[p1]!='6' && coman_leer1[p1]!='7' &&
coman_leer1[p1]!='8' && coman_leer1[p1]!='9' && coman_leer1[p1]!='a' && coman_leer1[p1]!='A'){
        if(coman_leer1[p1]!='b' && coman_leer1[p1]!='B' && coman_leer1[p1]!='c' && coman_leer1[p1]!='C' &&
coman_leer1[p1]!='d' && coman_leer1[p1]!='D' && coman_leer1[p1]!='e' && coman_leer1[p1]!='E' &&
coman_leer1[p1]!='f' && coman_leer1[p1]!='F'){
            Serial.println("\n\r\n\rERROR: Tiene que ser en formato HEXADECIMAL!\n\r");
            while(Serial.available()){
                delay(250); //Espera 1/4 de segundo
                coman_leer1[p1] = Serial.read(); //Lee los caracteres escritos en la PC y las guarda en el arreglo coman_leer1, la
posicion en que se guarda en el arreglo es definido por la variable p1
                var22=coman_leer1[p1]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo
coman_leer1, en la posicion definida por la variable p1
                //Serial.write(coman_leer1[p1]); //Se imprime en la pantalla de la PC el caracter guardado por el arreglo
coman_leer1 en la posicion definida por la variable p1
                y++; //Aumenta en uno la variable y, para activar el primer proceso de crear un byte de dos numeros hex
                p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer1
            }
            var1=0; //La variable se iguala a 0 para que ya no continue con los procesos y comience todo de nuevo
            x=100; //La variable se iguala a 100 para que ya no continue con los procesos y comience todo de nuevo

```

```

    cont1=8000; //La variable se iguala a 8000 para que ya no continúe con los procesos y comience todo de nuevo
  }
}
if(coman_leer1[p1]=='0'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x00; //Sele suma 0x00 al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='1'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x01; //Sele suma 0x01 al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='2'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x02; //Sele suma 0x02 al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='3'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x03; //Sele suma 0x03 al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='4'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x04; //Sele suma 0x04 al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='5'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x05; //Sele suma 0x05 al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='6'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x06; //Sele suma 0x06 al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='7'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x07; //Sele suma 0x07 al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='8'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x08; //Sele suma 0x08 al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='9'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x09; //Sele suma 0x09 al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='a' || coman_leer1[p1]=='A'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x0A; //Sele suma 0x0A al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
  numeros en hex
}
if(coman_leer1[p1]=='b' || coman_leer1[p1]=='B'){
  comand_rfid1[p2] = coman_leer1[p1-1] + 0x0B; //Sele suma 0x0B al numero guardado en el arreglo coman_leer1
  en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos

```

```

numeros en hex
}
if(coman_leer1[p1]=='c' || coman_leer1[p1]=='C'){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x0C; //Sele suma 0x0C al numero guardado en el arreglo coman_leer1
en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]=='d' || coman_leer1[p1]=='D'){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x0D; //Sele suma 0x0D al numero guardado en el arreglo coman_leer1
en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]=='e' || coman_leer1[p1]=='E'){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x0E; //Sele suma 0x0E al numero guardado en el arreglo coman_leer1
en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
if(coman_leer1[p1]=='f' || coman_leer1[p1]=='F'){
    comand_rfid1[p2] = coman_leer1[p1-1] + 0x0F; //Sele suma 0x0F al numero guardado en el arreglo coman_leer1
en la pocicion p1-1, el resultado es guardandolo en el arreglo comand_rfid1 posicion p2, creando asi un byte de dos
numeros en hex
}
p2++; //Aumenta en uno la variable p2 para avanzar en la posscion del arreglo comand_rfid1
z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la orden
de seleccionar una tag especifica
}

if(y==2){
    y=0; //Se inicializa de nuevo la variable y, para asi formar de nuevo un byte de dos numeros Hex
}

} //Fin de trans_hex()

/*Al ingresar una cantidad impar de caracteres desde la PC, la funcion toma el ultimo caracter y lo convierte en un byte
de dos numeros hexadecimales, agregando un cero a la izquierda de dicho caracter*/
void hex_impar(){

    if(y==1){
        //Al ingresar un caracter que no es un numero HEX el proceso lo detecta y no deja que continue con los demas
procesos y todo comienza de nuevo
        if(var22!='0' && var22!='1' && var22!='2' && var22!='3' && var22!='4' && var22!='5' && var22!='6' && var22!='7'
&& var22!='8' && var22!='9' && var22!='a' && var22!='A'){
            if(var22!='b' && var22!='B' && var22!='c' && var22!='C' && var22!='d' && var22!='D' && var22!='e' &&
var22!='E' && var22!='f' && var22!='F'){
                Serial.println("\n\r \n\rERROR: Tiene que ser en formato HEXADECIMAL!\n\r");
                while(Serial.available()){
                    delay(250); //Espera 1/4 de segundo
                    coman_leer1[p1] = Serial.read(); //Lee los caracteres escritos en la PC y las guarda en el arreglo coman_leer1, la
posicion en que se guarda en el arreglo es definido por la variable p1
                    var22=coman_leer1[p1]; //Se otorga a la variable var22 el valor del caracter que esta guardado en el arreglo
coman_leer1, en la posicion definida por la variable p1
                    //Serial.write(coman_leer1[p1]); //Se imprime en la pantalla de la PC el caracter guardado por el arreglo
coman_leer1 en la posicion definida por la variable p1
                    y++; //Aumenta en uno la variable y, para activar el primer proceso de crear un byte de dos numeros hex
                    p1++; //Aumenta en uno la variable p1 para poder avanzar en la posicion del arreglo coman_leer1
                }
                var1=0; //La variable se iguala a 0 para que ya no continue con los procesos y comience todo de nuevo
                x=100; //La variable se iguala a 100 para que ya no continue con los procesos y comience todo de nuevo
                cont1=8000; //La variable se iguala a 8000 para que ya no continue con los procesos y comience todo de nuevo
            }
        }
    }
}

```

```

if(var22=='0'){
    comand_rfid1[p2]=0x00; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0x00 lo
cambia por 0x00, formando asi un Byte de dos numeros hex
}
if(var22=='1'){
    comand_rfid1[p2]=0x01; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0x10 lo
cambia por 0x01, formando asi un Byte de dos numeros hex
}
if(var22=='2'){
    comand_rfid1[p2]=0x02; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0x20 lo
cambia por 0x02, formando asi un Byte de dos numeros hex
}
if(var22=='3'){
    comand_rfid1[p2]=0x03; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0x30 lo
cambia por 0x03, formando asi un Byte de dos numeros hex
}
if(var22=='4'){
    comand_rfid1[p2]=0x04; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0x40 lo
cambia por 0x04, formando asi un Byte de dos numeros hex
}
if(var22=='5'){
    comand_rfid1[p2]=0x05; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0x50 lo
cambia por 0x05, formando asi un Byte de dos numeros hex
}
if(var22=='6'){
    comand_rfid1[p2]=0x06; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0x60 lo
cambia por 0x06, formando asi un Byte de dos numeros hex
}
if(var22=='7'){
    comand_rfid1[p2]=0x07; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0x70 lo
cambia por 0x07, formando asi un Byte de dos numeros hex
}
if(var22=='8'){
    comand_rfid1[p2]=0x08; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0x80 lo
cambia por 0x08, formando asi un Byte de dos numeros hex
}
if(var22=='9'){
    comand_rfid1[p2]=0x09; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0x90 lo
cambia por 0x09, formando asi un Byte de dos numeros hex
}
if(var22=='a' || var22=='A'){
    comand_rfid1[p2]=0x0A; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0xA0 lo
cambia por 0x0A, formando asi un Byte de dos numeros hex
}
if(var22=='b' || var22=='B'){
    comand_rfid1[p2]=0x0B; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0xB0 lo
cambia por 0x0B, formando asi un Byte de dos numeros hex
}
if(var22=='c' || var22=='C'){
    comand_rfid1[p2]=0x0C; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0xC0 lo
cambia por 0x0C, formando asi un Byte de dos numeros hex
}
if(var22=='d' || var22=='D'){
    comand_rfid1[p2]=0x0D; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0xD0 lo
cambia por 0x0D, formando asi un Byte de dos numeros hex
}
if(var22=='e' || var22=='E'){
    comand_rfid1[p2]=0x0E; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0xE0 lo
cambia por 0x0E, formando asi un Byte de dos numeros hex
}
if(var22=='f' || var22=='F'){

```

```

    comand_rfid1[p2]=0x0F; //Se cambia el valor guardado en el arreglo comand_rfid1 posicion p2, el cual es 0xF0 lo
    cambia por 0x0F, formando asi un Byte de dos numeros hex
    }
    z1=z1+0x01; //Aumenta en uno la variable z1, contando cuantos bytes en formato hex estan en el arreglo de la orden
    de seleccionar una tag especifica
    }
    p2++;
} //Fin de hex_impar()

/*Esta funcion transforma todos los caracteres que componen la frecuencia en una sola variable que se puede manipular
mas facilmente*/
void num_freq(){
    if(pp==0){
        Freqx[0]=Freqx[0]*100000; //Agrega 5 ceros a la derecha del numero que esta en el arreglo Freqx en la posicion 0,
        multiplicando por 100000 dicho numero
        //Serial.println(Freqx[0]); //Imprime en la pantalla de la PC el resultado de multiplicar 100000 al numero en el arreglo
        Freqx en la posicion 0
    }
    if(pp==1){
        Freqx[1]=Freqx[1]*10000; //Agrega 4 ceros a la derecha del numero que esta en el arreglo Freqx en la posicion 1,
        multiplicando por 10000 dicho numero
        //Serial.println(Freqx[1]); //Imprime en la pantalla de la PC el resultado de multiplicar 10000 al numero en el arreglo
        Freqx en la posicion 1
    }
    if(pp==2){
        Freqx[2]=Freqx[2]*1000; //Agrega 3 ceros a la derecha del numero que esta en el arreglo Freqx en la posicion 2,
        multiplicando por 1000 dicho numero
        //Serial.println(Freqx[2]); //Imprime en la pantalla de la PC el resultado de multiplicar 1000 al numero en el arreglo
        Freqx en la posicion 2
    }
    if(pp==3){
        Freqx[3]=Freqx[3]*100; //Agrega 2 ceros a la derecha del numero que esta en el arreglo Freqx en la posicion 3,
        multiplicando por 100 dicho numero
        //Serial.println(Freqx[3]); //Imprime en la pantalla de la PC el resultado de multiplicar 100 al numero en el arreglo
        Freqx en la posicion 3
    }
    if(pp==4){
        Freqx[4]=Freqx[4]*10; //Agrega 1 ceros a la derecha del numero que esta en el arreglo Freqx en la posicion 4,
        multiplicando por 10 dicho numero
        //Serial.println(Freqx[4]); //Imprime en la pantalla de la PC el resultado de multiplicar 10 al numero en el arreglo
        Freqx en la posicion 4
    }
    if(pp==5){
        //Suma todos los numeros del arreglo Freqx de las posiciones 1, 2, 3, 4 y 5 y lo guarda en la posicion 6 de dicho
        arreglo,
        //conteniendo asi todo el valor de toda la frecuencia en la posicion 6 del arreglo
        Freqx[6]=Freqx[0]+Freqx[1]+Freqx[2]+Freqx[3]+Freqx[4]+Freqx[5];
        //Serial.println("\n\r"); //Imprime 2 ENTER en la pantalla de la PC
        //Serial.println(Freqx[6]); //Imprime en la pantalla de la PC el valor que contiene el arreglo Freqx en la posicion 6, el
        cual contiene todo el valor de la frecuencia
    }
} //Fin de void num_freq()

```

 Código de configuración cargado en el ARDUINO UNO, para el control del módulo GSM SIM900:

```

/*CODIGO ARDUINO UNO - PROGRAMA GSM*/

#include <SIM900.h>
#include <SoftwareSerial.h>
//If not used, is better to exclude the HTTP library,
//for RAM saving.
//If your sketch reboots itself probably you have finished,
//your memory available.
#include <inetGSM.h>

//If you want to use the Arduino functions to manage SMS, uncomment the lines below.
#include <sms.h>
SMSGSM sms;

//To change pins for Software Serial, use the two lines in GSM.cpp.

//GSM Shield for Arduino
//www.open-electronics.org
//this code is based on the example of Arduino Labs.

//Codigo corto para enviar y recibir SMS.

void enviar_sms(); //Funcion utilizada para que el usuario pueda enviar un mensaje de texto a un numero de telefono

int numdata;
int xx1=1;
int xx2=0;
int pp=0; //Variable que se utiliza para que el sistema recorra las partes del arreglo numcely
int pp1=0; //Variable que se utiliza para que el sistema recorra las partes del arreglo txt2
int t=0;
boolean started=false; //Variable que se utiliza para saber el estado de la comunicacion entre el sistema y el shield
GSM
boolean enviarsms=false;
char smsbuffer[160]; //Arreglo que se utiliza para guardar el mensaje de texto que el usuario a enviado hacia el sistema
char n[20];
char vary; //Variable para verificar si el usuario requiere mandar un nuevo mensaje, al ingresar el caracter w el sistema
llamara a la funcion para mandara un nuevo mensaje, si no es asi la funcion nunca se llamara
char* numcel1;
String numcelx;
char numcely[20]; //="70250243";Arreglo que utiliza el sistema para guardar el numero de telefono al cual se enviara el
mensaje
char* txt1="HOLA COMO ESTAS";
char txt2[200]; //Arreglo para guardar el mensaje de texto ingresado por el usuario

//debug begin
char sms_position;
char phone_number[20]; //string para el numero de telefono, que el usuario manda hacia el sistema
char sms_text[100];
int i;
//debug end

void setup()
{
  //Serial connection.
  Serial.begin(9600);
  Serial.println("GSM Shield testing.");
  if (gsm.begin(4800)){
    Serial.println("\nstatus=READY");
    started=true; //Se otorga el estado de true a la variable para identificar que el shield GSM esta activo y se
comunica correctamente con el sistema

```



```

while(xx1==1){
  delay(1000); //Espera 1 segundo
  while(Serial.available()){
    numcely[pp]=Serial.read(); //Guarda caracter por caracter ingresado por el usuario, en el arreglo numcely, en la
posicion determinada por la variable pp
    if(numcely[pp]!='0' && numcely[pp]!='1' && numcely[pp]!='2' && numcely[pp]!='3' && numcely[pp]!='4' &&
numcely[pp]!='5' && numcely[pp]!='6' && numcely[pp]!='7' && numcely[pp]!='8' && numcely[pp]!='9' &&
numcely[pp]!='+'){
      Serial.println("ERROR: El numero de telefono debe de ser en decimal!!");
      xx2=1;
      //break; //Sale de la funcion inmediatamente, ya que se ingreso un caracter que no es un numero decimal
    }
    pp++; //Aumenta en 1 la variable para que se pueda abansar en la siguiente posicion del arreglo numcely
    xx1=2; //Se le otorga e valor de 2 a la variable para que pueda continuar con el siguiente proceso y terminar el
actual
  }
}
Serial.println(numcely); //Imprime en la pantalla de la PC el numero de telefono ingresado por el usuario

if(xx2==0){
  Serial.println("Ingrese el mensaje de texto:");
  while(xx1==2){
    delay(1000); //Espera 1 segundo
    while(Serial.available()){
      txt2[pp1]=Serial.read(); //Guarda caracter por caracter ingresado por el usuario, en el arreglo txt2, en la
posicion determinada por la variable pp1
      pp1++; //Aumenta en 1 la variable para que se pueda abansar en la siguiente posicion del arreglo txt2
      xx1=3; //Se le otorga e valor de 3 a la variable para terminar el proceso actual
    }
  }
  //Serial.println(txt2); //Imprime en la pantalla de la PC el mensaje de texto ingresado por el usuario
  enviarsms=true;
}
}
else
{
  Serial.println("\nstatus=IDLE"); //Imprime en la pantalla de la PC el mensaje, cuando el shield GSM no esta
activado o no puede comunicarse con el sistema
};

if(started){
  //Habilitar las siguientes dos lineas si se desea enviar un SMS
  if(xx2==0){
    if (sms.SendSMS(numcely, txt2))
      Serial.println("\nSMS sent OK"); //Imprime en la pantalla de a PC el mensaje, cuando se envio correctamente el
mensaje de texto
  }

  //Si NO SPACE ,se necesita borrar SMS desde la posicion 1 a la posicion 20
  //please enable this four lines;
  //for(i=1;i<=20;i++){
  //  //sms.DeleteSMS(i);
  //}
}
}; //Fin de la funcion enviar_sms

```