



INSTITUTO POLITÉCNICO NACIONAL

**CENTRO DE INVESTIGACIÓN EN
COMPUTACIÓN**

**MAESTRÍA EN CIENCIAS DE LA
COMPUTACIÓN**

**IDENTIFICACIÓN DE VEHÍCULOS
EMPLEANDO RADIO FRECUENCIA
(RFID – EPC)**

TESIS

**QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA
COMPUTACIÓN**

P R E S E N T A

ING. ROBERTO HERNÁNDEZ ATILANO

DIRECTOR DE TESIS

M. EN C. SERGIO SANDOVAL REYES



**CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
MÉXICO, D.F.
MAYO DEL 2007**

Dedicatoria

A mi esposa e hijo por ser parte de mi vida.

A mi mamá por su cariño y mis hermanas por su apoyo.

Agradecimientos

Al Instituto Politécnico Nacional, al Centro de Investigación en Computación, al Consejo Nacional de Ciencia y Tecnología, a los profesores del Centro de Investigación en Computación.

Gracias por su apoyo a lo largo de mi formación como maestro en ciencias al profesor Sergio Sandoval Reyes.

Un agradecimiento muy especial a la profesora Sandra Dinora Orantes Jiménez, por su valiosa ayuda durante la elaboración de mi tesis, por su amistad y apoyo que me ha brindado.

Índice de Contenido

<i>Índice de Contenido</i>	v
<i>Lista de Figuras</i>	vii
<i>Lista de Tablas</i>	ix
1. Capítulo 1 - Introducción a la tecnología RFID	1
1.1. Identificación de objetos por RFID	2
1.2. Elementos de un sistema RFID	2
1.2.1. Etiqueta o transpondedor	2
1.2.2. Lector o interrogador	5
1.2.3. Principios básicos de funcionamiento	7
1.2.4. Otros aspectos de RFID	9
1.3. Definición del problema	12
1.4. Justificación de la tesis	12
1.5. Objetivos	13
1.5.1. General	13
1.5.2. Específicos	13
1.6. Organización de la tesis	14
2. Capítulo 2 - Trabajos relacionados de RFID en vehículos	16
2.1. Trabajos relacionados con la identificación de vehículos	17
2.2. Limitaciones de los trabajos relacionados	19
2.3. Solución propuesta	20
2.3.1. Consideraciones del proyecto	20
2.3.2. Análisis técnico	21
2.3.3. Interfaz del entorno	21
2.3.4. Evaluación de la fabricación	21
3. Capítulo 3 – Análisis y diseño de la aplicación	23
3.1. Análisis de la solución	24
3.2. Arquitectura de la aplicación	25
3.2.1. Casos de Uso General y Principales	26
3.2.2. Diagramas de Secuencias	28
3.2.3. Diagramas de Colaboración	32
3.2.4. Diagrama de Clases	35
3.2.5. Diagrama de Actividades	37
3.3. Puntos clave del diseño	38
4. Capítulo 4 – Implementación de la aplicación	42
4.1. Creación de la base de datos	43
4.2. Implementación del software	46
4.3. Características del lector y las etiquetas	48
4.4. Comunicación lector – computadora	51

4.5. Interacción con la base de datos	54
5. Capítulo 5 - Pruebas y resultados	58
5.1. Diseño de pruebas	59
5.1.1. Consultar la versión del software	61
5.1.2. Consultar los datos del creador y nombre de la aplicación	62
5.1.3. Consultar la versión del lector (Firmware)	63
5.1.4. Consultar la temperatura del lector	65
5.1.5. Lectura de una etiqueta	66
5.1.6. Realizar un Soft Reset	68
5.1.7. Crear un registro	69
5.1.8. Editar un registro	71
5.1.9. Borrar un registro	72
5.2. Resultados obtenidos	73
6. Capítulo 6 – Conclusiones	75
6.1. Objetivos logrados	76
6.2. Mejoras a futuro	77
6.2.1. Incorporación de nuevas etiquetas de otras clases	77
6.2.2. Acceso vía Web a la base de datos del sistema	78
Glosario	79
Bibliografía	89
Anexos	93
A. Código Fuente	94
B. Manual de Operaciones	94
C. Especificaciones Técnicas del Lector AWID	94
D. Diagrama de Clases	95

Lista de Figuras

Figura 1.1. Esquema de una etiqueta de RFID	3
Figura 1.2. Diseño interno de un lector que puede trabajar con dos frecuencias ...	6
Figura 1.3. Tipos de Lectores	7
Figura 1.4. Métodos de propagación de la información	7
Figura 1.5. Etiqueta RFID empleada para la recaudación con peaje electrónico ...	9
Figura 1.6. Sistema Desarrollado	13
Figura 2.1. Detección de camiones de SicFlotas	17
Figura 2.2. Especificaciones Técnicas del lector SicFlotas	18
Figura 2.3. Lector GP90-A de la compañía GIGA-TMS	18
Figura 2.4. Detección de Vehículos por Radio Frecuencia	20
Figura 3.1. Diagrama de Caso de Uso General	26
Figura 3.2. Diagrama de Caso de Uso del Proceso de Lectura de una Etiqueta	28
Figura 3.3. Diagrama de Secuencias del Proceso de Lectura de una Etiqueta	29
Figura 3.4. Diagrama de Secuencias del Proceso de Consultar Temperatura	30
Figura 3.5. Diagrama de Secuencias del Proceso de Consultar Versión	30
Figura 3.6. Diagrama de Secuencias del Proceso de Soft Reset del lector	31
Figura 3.7. Diagrama de secuencias del proceso de validar clave de administrador	32
Figura 3.8. Diagrama de Colaboración del Proceso de Lectura de una Etiqueta ...	33
Figura 3.9. Diagrama de Colaboración del Proceso de Consultar Temperatura ...	33
Figura 3.10. Diagrama de Colaboración del Proceso de Consultar Versión	34
Figura 3.11. Diagrama de Colaboración del Proceso de Soft Reset del lector	34
Figura 3.12. Diagrama de Colaboración del Proceso de validar clave de administrador	35
Figura 3.13. Diagrama de Clases del sistema	36
Figura 3.14. Diagrama de Actividades	37
Figura 3.15. Grafo convencional del software	39
Figura 3.16. Grafo convencional de las operaciones del administrador	40
Figura 4.1. Diagrama Entidad Relación	43
Figura 4.2. Instalación de NetBeans en Windows	46
Figura 4.3. Instalación de MySQL en Windows	48
Figura 4.4. Comunicación entre el lector y las Etiquetas	48
Figura 4.5. Comandos del Lector	49
Figura 4.6. Fórmula ocupada para obtener la temperatura del lector	50
Figura 4.7. Comandos de las Etiquetas EPC clase 1	51
Figura 4.8. Control de Flujo	53
Figura 4.9. Tabla de Registros	54
Figura 4.10. Tabla de Propietarios	54
Figura 4.11. Tabla de Etiquetas	55

Figura 4.12. Tabla de Marcas (de vehículos)	55
Figura 4.13. Tabla de Vehículos	56
Figura 5.1. Funciones en la ceja de Información	59
Figura 5.2. Funciones en la ceja de Acciones	60
Figura 5.3. Funciones en la ceja de Base de Datos	60
Figura 5.4. Validación de acceso a Base de Datos	60
Figura 5.5. Administración de la Base de Datos	61
Figura 5.6. Conexión del Portmon con el puerto que utiliza el Lector	61
Figura 5.7. Resultado del comando Versión del Sistema	62
Figura 5.8. Resultado del comando Acerca de	63
Figura 5.9. Resultado del comando Versión del Lector (Firmware)	64
Figura 5.10. Captura de Portmon del proceso de Versión del Lector	64
Figura 5.11. Resultado del comando Temperatura	65
Figura 5.12. Captura de Portmon del proceso de Temperatura	65
Figura 5.13. Captura de Portmon del proceso de Leer Etiqueta	67
Figura 5.14. Resultado del comando de Leer Etiqueta	67
Figura 5.15. Resultado del comando Soft Reset	68
Figura 5.16. Captura de Portmon del proceso de Soft Reset	69
Figura 5.17. Registros almacenados en la tabla Marca	70
Figura 5.18. Creación de un registro en la tabla Marca	70
Figura 5.19. Ventana de opciones en la edición de un registro de la tabla vehículos	71
Figura 5.20. Registros almacenados en la tabla Vehículos	71
Figura 5.21. Edición de un registro en la tabla Vehículos	72
Figura 5.22. Registros almacenados en la tabla Propietarios	73
Figura 5.23. Borrado de un registro en la tabla Propietarios	73

Lista de Tablas

Tabla 1.1. Clases de Etiquetas EPC	5
Tabla 2.1. Especificaciones Técnicas del lector GP90-Av	19
Tabla 4.1. Nivel 0 de Normalización	44
Tabla 4.2. Nivel 1 de Normalización	44
Tabla 4.3. Nivel 2 de Normalización	45
Tabla 4.4. Nivel 3 de Normalización	45
Tabla 4.5. Estructura del paquete	49
Tabla 4.6. CRC utilizado para la transmisión	52
Tabla 4.7. CRC utilizado para la recepción	52

IDENTIFICACIÓN DE VEHÍCULOS EMPLEANDO RADIO FRECUENCIA (RFID – EPC)

Resumen

La detección de objetos se hace con una tecnología llamada RFID (Radio Frequency Identification, Identificación por Radio Frecuencia), en la cual un dispositivo lector puede estar vinculado a un equipo de cómputo, que se comunica a través de una antena con una etiqueta (también conocida como transpondedor) mediante ondas de radio.

Esta tecnología ha tenido tanta aceptación en el mercado, que se decidió crear un estándar para brindar mejores servicios. A este se le conoce como: EPC (Electronic Product Code, Código Electrónico del Producto) que se apoya en RFID para identificar de manera única a los productos en sus distintas unidades de empaque para la cadena de suministro.

Se propone utilizar esta tecnología RFID y el estándar EPC para la identificación de vehículos, con la finalidad de desarrollar un sistema que sea capaz de controlar la entrada y salida de los automóviles particulares, esta aplicación puede ser utilizada en escuelas, fábricas, pensiones, etc. Uno de los requisitos de empleo es que se cuente con una persona que se encargue de controlar el sistema.

Esta aplicación permite la comunicación entre lector (fijo) y etiquetas EPC clase 1 (pasivas), del mismo modo administra una base de datos, la cual permite almacenar la información necesaria para controlar el acceso de los vehículos.

La aplicación propuesta provee de seguridad ya que permite la identificación del vehículo al momento de colocarse dentro del radio de identificación del lector RFID-EPC, mostrando la información asociada con el automóvil y del propietario.

Palabras clave: RFID, lector, etiquetas, identificación de vehículos.

IDENTIFICATION OF VEHICLES USING RADIO FREQUENCY (RFID - EPC)

Abstract

The detection of objects using a technology called RFID (Radio Frequency Identification), in which a RFID reader connected to a computer, which communicates across an antenna with a tag (also known as transponder) by means of waves of radio.

This technology has had great success on the market, which it decided to create a standard to offer better services. This one is known as: EPC (Electronic Product Code) that rests on RFID to identify in a unique way to the products in his different units of packing for the chain of supply.

The proposes to use this RFID technology and the standard EPC for the identification of vehicles, with the purpose of developing a system that is capable of controlling the ingress and egress the vehicles. This application can be used in schools, parking lots, etc. One of the requisites of employment is that one is provided with a person who should be in charge of controlling the system.

This application allows the communication between a (fixed) reader and tags EPC class 1 (passive), to manage a database, which stores the necessary information to control the access of vehicles.

The proposed application provides security since it allows the identification of the vehicle as soon as inside the radio of identification of the reader RFID-EPC, showing the information associated with the automobile and the owner.

Keywords: RFID, reader, tags, identification of vehicles.

Capítulo 1

Introducción a la tecnología RFID

Este capítulo muestra una pequeña introducción a la tecnología RFID, del mismo modo se presenta la problemática a resolver y la justificación del porque este trabajo, así como una breve descripción de todo el documento.

1.1. Identificación de objetos por RFID

Un sistema de RFID (*Radio Frequency IDentification*, Identificación por radiofrecuencia), puede estar formado por varios componentes: etiquetas, lectores de etiquetas, estaciones de programación de etiquetas, lectores de circulación y equipamiento de ordenación. El propósito de un sistema RFID es permitir que se puedan transmitir datos mediante un dispositivo portátil, llamado etiqueta, que es leída por un lector y procesada según las necesidades de una aplicación determinada. Los datos transmitidos por la etiqueta pueden proporcionar información sobre la identificación del producto marcado con esta, como por ejemplo: el precio, color, fecha de compra, etc. El uso de RFID para aplicaciones de acceso y de seguimiento aparecieron por primera vez durante los años 80's. Pronto esta tecnología destacó debido a su capacidad de identificar objetos móviles.

El EPC (*Electronic Product Code*, Código Electrónico del Producto) es un estándar que utiliza la tecnología RFID para identificar de manera única a los productos en sus distintas unidades de empaque. El EPC es la evolución del código de barras.

La tecnología del Código Electrónico de Producto, apunta a convertirse en uno de los más importantes estándares de intercambio de comunicación para eficientar los procesos de manufactura, desde el control de inventarios, la certificación de materiales, el rastreo de piezas, las aplicaciones logísticas, hasta la lucha contra la piratería y la falsificación. Es ideal para giros productivos como el farmacéutico y el de alimentos.

1.2. Elementos de un sistema RFID

Un sistema RFID consta básicamente de dos elementos: un lector (reader) y una etiqueta (transpondedor) [Klaus 2003].

1.2.1. Etiqueta o transpondedor

A las etiquetas también se les conoce con el nombre de transpondedor, que deriva de *TRANSMitter/resPONDER* (Transmisión/Recepción), lo cual explica su funcionamiento. Los componentes básicos de una etiqueta o transpondedor se pueden distinguir en la Figura 1.1. y son:

- Una memoria no volátil donde se almacenan datos.
- Una memoria ROM (memoria de solo lectura) donde se almacenan instrucciones básicas para el funcionamiento, tales como temporizadores, controladores de flujo de datos, etc.

- También puede incorporar memoria RAM (memoria de lectura y escritura) para almacenar datos durante la comunicación con el lector.
- La antena por la cual detecta el campo creado por el interrogador, y del que extrae energía para su comunicación con él.
- Diversos componentes electrónicos que procesan la señal de la antena y para el proceso de datos, como buffers, filtros, etc.

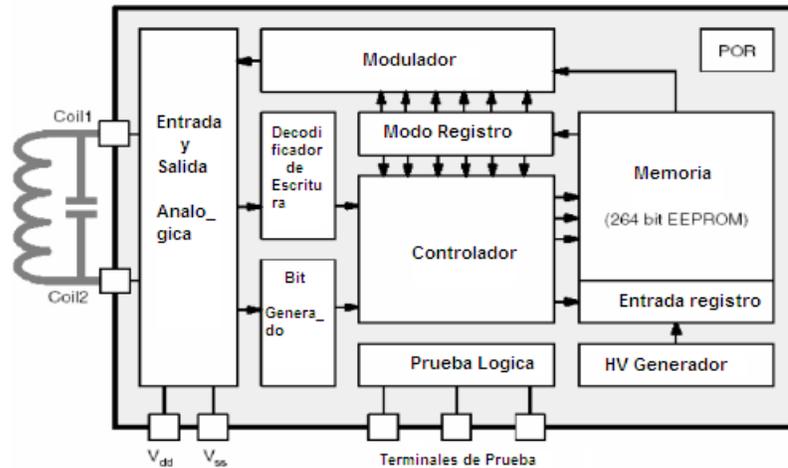


Figura 1.1. Esquema de una etiqueta de RFID

Las etiquetas necesitan poca energía para su funcionamiento, del orden de los mw (mili watts). Podemos diferenciar dos tipos de etiquetas dependiendo de la energía que utilizan para la comunicación:

- *Etiquetas activas:* son transpondedores que necesitan el apoyo de baterías adicionales, ya que no tienen suficiente energía con la que proporciona el lector.

Este tipo de etiqueta tiene la ventaja de poseer un alcance mayor de comunicación e incluso no necesitan que el lector sea quién inicie la comunicación. Además permiten habitualmente procesos de lectura y reescritura enviando previamente instrucciones al lector y la utilización de memorias más grandes (existen etiquetas con 1MegaByte de memoria). Por el contrario ofrecen una vida útil limitada (menos de diez años), dependiendo del tipo de batería y de las temperaturas a las que opera. También hay que destacar que su costo es bastante elevado, su precio suele ser cinco veces más alto que las etiquetas pasivas. De esta forma aparecen nuevas aplicaciones para sistema RFID gracias a este tipo de etiquetas alimentadas por baterías.

- *Etiquetas pasivas:* son transpondedores que no necesitan baterías adicionales, ya que únicamente se alimentan de la energía del campo generado por el

lector. Para las etiquetas pasivas, la energía que necesitan para transmitir la información que contienen proviene en su totalidad de la señal generada por el lector. Estas etiquetas aprovechan la energía suministrada por un lector para generar su propia señal que recibe nuevamente el lector.

Frecuencia y velocidad de transmisión

Las etiquetas también se pueden clasificar según el rango de frecuencias en el que operan, es decir, en qué frecuencias se comunican con el lector:

- LF (Low Frequency, baja frecuencia) en el rango de 120 Khz-134 Khz.
- HF (High Frequency, alta frecuencia) en el rango de 13.56 Mhz.
- UHF (Ultra High Frequency, ultra alta frecuencia) en el rango de 868-956 Mhz.
- Microondas (Microwave) en el rango de 2.45 Ghz, conocida como banda ISM (Industrial Scientific and Medical).

Una mayor frecuencia suele significar un menor alcance pero una mayor velocidad en la transmisión de datos, aunque también encarece el precio del sistema.

Las etiquetas RFID de baja frecuencia se utilizan comúnmente para la identificación de animales, seguimiento de barricas de cerveza, y como llave de automóviles con sistema antirrobo. En ocasiones se insertan en pequeños chips en mascotas, para que puedan ser devueltas a su dueño en caso de pérdida. En los Estados Unidos se utilizan dos frecuencias para RFID: 125 Khz. (el estándar original) y 134.5 Khz. (el estándar internacional). Las etiquetas RFID de alta frecuencia se utilizan en bibliotecas y seguimiento de libros, control de acceso en edificios, seguimiento de equipaje en aerolíneas y seguimiento de artículos de ropa. Un uso extendido de las etiquetas de alta frecuencia es como identificación de insignias, substituyendo a las anteriores tarjetas de banda magnética. Sólo es necesario acercar estas insignias a un lector para autenticar al portador.

Las etiquetas RFID de UHF se utilizan comúnmente de forma comercial en seguimiento de “pallets” (generalmente de madera, sirven para estibar grandes cantidades de mercancías) y envases, y seguimiento de camiones y remolques en envíos.

Las etiquetas RFID de microondas se utilizan en el control de acceso en vehículos de gama alta (Generalmente en peaje electrónico).

Algunas autopistas, como por ejemplo la FasTrak de California, el sistema I-Pass de Illinois y la *Philippines South Luzon Expressway E-Pass* utilizan etiquetas RFID para recaudación con peaje electrónico. Las tarjetas son leídas mientras los vehículos pasan; la información se utiliza para descontar el peaje de una cuenta prepago. El sistema ayuda a disminuir el tráfico causado por las cabinas de peaje.

Una forma más de distinguir las etiquetas es de acuerdo a un protocolo desarrollado por el Auto-ID Center¹, el cual divide las etiquetas por clases [EPC México 2007].

La Tabla 1.1. Muestra las clases en las que se dividen las etiquetas y da una descripción de lo que concierne a cada una de esas clases.

Tabla 1.1. Clases de Etiquetas EPC

EPC Clases de Etiquetas	
Clases	Características
Clase 0	Solo de lectura, etiquetas de identificación pasiva.
Clase I	De escritura una vez, etiquetas de identificación pasiva.
Clase II	Etiquetas pasivas con funciones adicionales como memoria ó encriptación.
Clase III	Etiquetas RFID semipasivas.
Clase IV	Etiquetas activas – comunicación con lectores y otras etiquetas sobre bandas de frecuencia similares.
Clase V	Esencialmente “lectores” con la capacidad de las clases I, II, III, así como con la clase IV ó cualquier otra.

1.2.2. Lector o interrogador

El otro elemento principal de un sistema RFID es el lector o interrogador. Los lectores (readers) son los encargados de enviar una señal de RF (Radio Frecuencia) para detectar las posibles etiquetas en un determinado rango de acción. En su fabricación se suelen separar en dos tipos:

- Sistemas con bobina simple, la misma bobina sirve para transmitir la energía y los datos. Son más simples y más baratos, pero tienen menos alcance.
- Sistemas interrogadores con dos bobinas, una para transmitir y otra para recibir datos. Son más caros, pero consiguen mayores prestaciones.

En la Figura 1.2, se muestra el circuito básico de un lector que puede trabajar en dos frecuencias diferentes, al centro de la imagen se muestra el módulo encargado de trabajar con la frecuencia de 915 Mhz y del lado derecho se encuentra la sección que se encarga de operar con la frecuencia de 13.56 Mhz.

¹ Laboratorios creados en Agosto del 2003, donde se desarrollan y se experimentan con aspectos técnicos de las etiquetas, existen laboratorios en Estados Unidos, Inglaterra, Australia, Japón, Suiza y China.

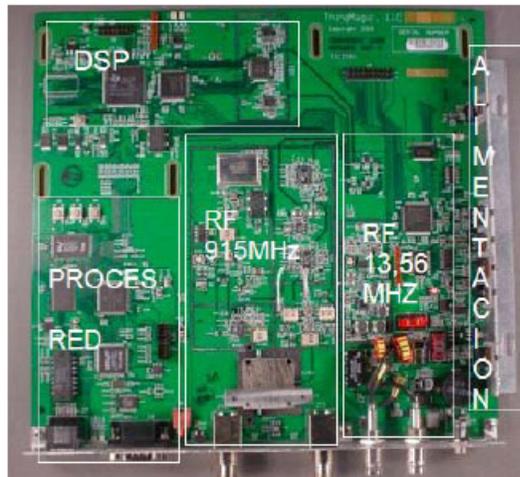


Figura 1.2. Diseño interno de un lector que puede trabajar con dos frecuencias.

Los lectores son más complejos dependiendo del transpondedor, si son sofisticados, los componentes del interrogador tienen que ser capaces de acondicionar la señal, detectar y corregir errores. Además pueden trabajar a más de una frecuencia.

Una vez que se ha recibido toda la información por parte del lector, se pueden emplear algoritmos para no confundir la transmisión actual con una nueva, indicándole a la etiqueta que deje de transmitir. Se suele usar para validar diversas etiquetas en un espacio corto de tiempo. Otro algoritmo usado por el lector, es ir llamando a los transpondedores por su número de identificación, indicándoles de esta forma el tiempo en el que deben transmitir.

Estos dos algoritmos son mecanismos para impedir la colisión de información.

En la Figura 1.3, podemos observar varios tipos de lectores de RFID:

- El lector de la izquierda, es un lector fijo de marca AWID modelo 2010AR [AWID 2007].
- El lector del centro, es un lector marca Tradewind, para ser usado por dispositivos con ranura de expansión SD, tales son el caso de los dispositivos PDA (Palm OS, Pocket PC, Symbian, etc.).
- El lector de la derecha, es un lector tipo móvil de marca Two Technologies, inc, modelo JETT*ce [EgoMexico 2007].



Figura 1.3. Tipos de Lectores

1.2.3. Principios básicos de funcionamiento

Un sistema de comunicación RFID se basa en la comunicación bidireccional entre un lector (interrogador) y una etiqueta (transpondedor), por medio de ondas de radiofrecuencia.

El sistema de transmisión de información varía según la frecuencia en la que trabaja. Así se puede clasificar un sistema de RFID en sistemas basados en el acoplamiento electromagnético o inductivo, y basados en la propagación de ondas electromagnéticas. Podemos apreciar esta diferenciación en la Figura 1.4.

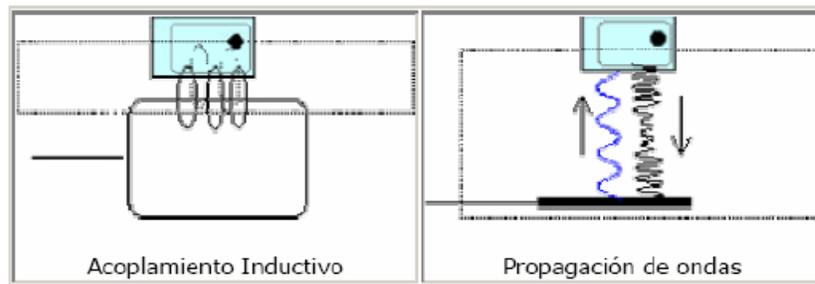


Figura 1.4. Métodos de propagación de la información

Hay que tener en cuenta que la comunicación se puede realizar en zonas industriales con metales, lo que unido a las características de ruido, interferencia y distorsión de estas comunicaciones vía radio complica la correcta recepción de bits.

La comunicación entre el lector y la etiqueta es del tipo asíncrona, lo que repercute en una mayor atención en parámetros como la forma en que se comunican los datos y la organización de flujo de bits. Todo esto conlleva el estudio de la denominada codificación de canal, con el fin de mejorar la recepción de información.

Como en toda comunicación vía radio se necesita entre los dos componentes de la comunicación un campo sinusoidal variable u onda portadora. La comunicación se consigue aplicando una variación a ese campo, ya sea en amplitud, fase o frecuencia, en función de los datos a transmitir. Este proceso se conoce como modulación. En RFID suelen ser aplicadas las modulaciones ASK (Amplitude shift keying), FSK (Frequency shift keying) y PSK (Phase shift keying).

Los diferentes métodos de propagación de la información son usados en diferentes frecuencias. De este modo el acoplamiento inductivo funciona a frecuencias más bajas y el sistema de propagación de ondas a frecuencias más elevadas. Existe también otro tipo de propagación usado en distancias menores a 1 centímetro, que puede trabajar teóricamente en frecuencias bajas hasta 30 Mhz, son los sistemas “close coupling”.

Estos sistemas usan a la vez campos eléctricos y magnéticos para la comunicación. La comunicación entre el lector y el transpondedor no ocasiona un gasto excesivo de energía, por lo que en estos sistemas se pueden usar microchips que tengan un consumo de energía bajo. Son sistemas usados generalmente en aplicaciones con un rango de alcance mínimo pero con estrictas medidas de seguridad. Se usa en aplicaciones como cerraduras de puertas electrónicas o sistemas de contactless smart card (tarjetas de identificación) [Klaus 2003]. Estos sistemas tienen cada vez menos importancia en el mercado de la tecnología RFID.

Por otro lado existen los sistemas de “remote coupling” basados en el acoplamiento inductivo (magnético) entre el lector y el transpondedor. Por eso, estos sistemas también son conocidos como “inductive radio systems”. Los sistemas basados con acoplamiento capacitivo (eléctrico) no son casi usados por la industria; en cambio los inductivos se puede decir que abarcan el 80% de los sistemas de RFID. Este sistema de comunicación entre el lector y el transpondedor trabaja en el rango de frecuencia comprendido entre los 135 Khz y los 13.56 Mhz. Aunque en algunas aplicaciones pueda trabajar a una frecuencia ligeramente más elevada. Su rango de alcance suele comprender alrededor de 1 metro. Estos sistemas siempre usan transpondedores pasivos.

Algunas aplicaciones

A continuación se muestran algunos ejemplos donde las empresas han utilizado la tecnología RFID para su beneficio propio.

La Figura 1.5. muestra una aplicación en la cual se utiliza un sistema de detección de vehículos implementada en algunas casetas de Estados Unidos.

Cabe mencionar que aquí, los vehículos son identificados en movimiento, es decir nunca se detienen para darles el acceso, por ello se requiere de etiquetas de mayor alcance (Microondas).



Figura 1.5. Etiqueta RFID empleada para la recaudación con peaje electrónico.

En México, importantes organizaciones han iniciado la adopción del EPC, tal es el caso de la tienda detallista Liverpool, donde desde el año 2004 se aplican pruebas piloto en sus áreas de línea blanca y muebles para el hogar. “A pesar de que el etiquetado EPC aún es relativamente caro, comparado con el código de barras, el uso de la tecnología EPC nos ha permitido disminuir errores en la marcación de los productos y en los envíos de mercancía, lo cual representa una gran ventaja competitiva para nuestro negocio”, resaltó Javier Méndez, gerente de Respuesta Dinámica de Liverpool México, quien reveló que esta empresa, en el año 2005, destinó alrededor de \$50,000 dólares en implementar las pruebas piloto, e invirtió otros \$300,000 dólares en el año 2006 para concluir las, con el objetivo de masificar esta solución hacia su cadena de proveeduría en el año 2007.

1.2.4 Otros aspectos de RFID

Siempre que se quiere implementar una tecnología nueva como es el caso de RFID es importante considerar costos de los equipos, condiciones ambientales y aspectos sociales.

Costos de etiquetas pasivas

La empresa Top Shop Netstore [Top 2007], provee de venta de etiquetas con las siguientes características:

- Etiqueta tipo: EPC clase 1 (pasivas).
- Área de trabajo: UHF Global (860 - 960 Mhz).
- Tamaño de la antena: 93 x 11 mm / 3.661 x 0.433".
- Tamaño real: 97 x 15 mm / 3.819 x 0.590".
- Etiqueta: en papel con adhesivo.
- Al costo de: 100 piezas por €100 euros.

Otra empresa que ofrece etiquetas pasivas es Rfidshopper, [Rfidshopper 2007], la cual vende por mayoreo o menudeo, las características de las etiquetas que ofrece son las siguientes:

Etiqueta modelo ALL-9338-02, la cual tiene un diseño económico el cual puede ser en papel o plástico, con el tamaño de 98.2 x 12.3 milímetros. (EPC clase 1 pasivas), con un costo de \$1 dólar por pieza.

Etiquetas modelo ALL-9354-02, la cual trabaja a 915 Mhz y se vende solo por mayoreo con un mínimo de 7500 piezas y tienen un costo de \$22 centavos de dólar por pieza.

Costos de etiquetas activas

La compañía Rfidshopper, provee también de etiquetas activas como son:

Etiqueta ALL-9334-02, disponibles en papel y plástico, ideal para espacios pequeños, tamaño: 50.8 x 50.8 milímetros, costo \$279.00 dólares.

Etiqueta ALL-9350-02, etiquetas de largo alcance, disponibles en papel o plástico, tamaño: 152.5 x 15.9 milímetros, costo \$119.00 dólares.

Costos de los lectores

EgoMexico [EgoMexico 2007], ofrece a la venta lectores RFID-EPC con las siguientes características:

- MPR-2010 Reader Development kit, incluye:
- Lector de EPC, modelo MPR-2010AR, marca AWID, 915 Mhz.
- Fuente de alimentación.
- 10 etiquetas EPC pasivas clase 1.
- Al precio de: \$19178.55 pesos más IVA.

Rfidshopper también vende lectores de RFID, esta empresa tiene un modelo ALR-9780, este trabaja con etiquetas EPC clase 1 and clase 1 generación 2.

Incluye: el lector, fuente de alimentación, guía de referencia, cable serial, pero la antena debe ser comprada por separada.

La antena es de radio frecuencia con polarización circular de 6 decibeles. Modelo: ALR-9610 BC.

En total tiene un costo de \$1999 (lector) + \$229 (antena) = \$2228 dólares.

Consideraciones de temperatura

Es importante considerar los aspectos de la temperatura, sobre todo en regiones donde las condiciones climáticas pudieran alterar los dispositivos electrónicos.

Por ello se deben tener presente siempre las condiciones de operación de cada uno de los productos que se utilizan, en el caso de la tecnología RFID, este factor es importante para garantizar un eficiente funcionamiento del equipo y de la identificación, con el fin de no tener alteraciones en la comunicación entre lector y etiquetas.

En el caso de los lectores de RFID es importante siempre revisar la temperatura en la que trabajan, con el fin de prevenir fallas en el dispositivo y alargar la vida de uso del producto.

El rango de temperatura² más utilizado por los fabricantes de lectores de RFID es de: Entre -30 grados centígrados y + 65 grados centígrados.

Impacto social

En un estudio encargado por el Centro de Auto Identificación de la Universidad de Cambridge³, y realizado en Alemania, Estados Unidos, Francia, Japón y el Reino Unido, se facilitan indicaciones iniciales sobre la reacción de la opinión pública a la introducción de la tecnología de RFID, sus promesas y amenazas potenciales. Según dicho estudio, la mayoría de los consumidores estimaban que la tecnología beneficiaba fundamentalmente a las empresas, pero apenas a ellos mismos. Lo peor de todo era la impresión de que no se les había dado alternativa.

Otros motivos de gran preocupación se referían a la privacidad, a la seguridad personal relacionada con el temor a que los delincuentes pudieran saber de las compras efectuadas por una persona o si ésta llevaba un reloj muy costoso, o con el temor a que la tecnología pudiera permitir ver a través de las paredes; con el miedo en relación con la salud (especialmente en Europa, tras la publicación de polémicos artículos de prensa acerca de los posibles efectos nocivos de los teléfonos móviles, como la aparición de tumores cerebrales, y por el hecho de que la tecnología de la RFID, al igual que la telefonía móvil, utilizara ondas de radio). El aumento del desempleo y otros efectos negativos en la mano de obra también se consideraron cuestiones importantes, especialmente en Alemania y Japón, que entonces se hallaban en plena recesión. En el estudio se recomendaba, entre otras cosas, que no se escatimaran esfuerzos en determinar los efectos que la introducción de la tecnología tendría en el empleo, la privacidad, y la seguridad y la salud.

² De acuerdo a las especificaciones de los lectores de la marca AWID, véase el Anexo C.

³ H. Duce: Executive briefing. Public policy: Understanding public opinion (Cambridge, Centro de Autoidentificación, Instituto para la fabricación, Universidad de Cambridge, 2003).

Otras consecuencias sociales

La Unión Europea ha expresado un interés pronunciado por que se aproveche la tecnología de la RFID para aumentar la competitividad económica de conformidad con los objetivos de la Agenda de Lisboa, de suerte que Europa se sitúe como economía de ámbito mundial, competitiva, dinámica e impulsada por el saber. Se considera que la RFID puede contribuir a la protección de la salud y la seguridad públicas, por lo que es fundamental crear un marco apropiado que permita comprender y aplicar con éxito la RFID en toda Europa. En su discurso inaugural de la Feria comercial CeBIT⁴ celebrada el 9 de marzo de 2006, en Hanover (Alemania), la Sra. Viviane Reding, Comisaria de la Unión Europea responsable de la Sociedad de la Información y los Medios de Comunicación, consideró que la tecnología de la RFID evolucionaría y alcanzaría un grado de funcionalidad sin precedentes, tanto en términos de capacidad de memoria como de procesamiento. Predijo que la RFID contribuiría a suprimir la frontera que hoy separa el ciberespacio del espacio real, al fusionar el mundo de los datos con el de las cosas a fin de que el mundo virtual de la web se haga material.

1.3. Definición del problema

Una vez descritas las partes que forman un sistema RFID, se decidió atacar el problema de la identificación de vehículos desarrollando un software que interactúe con un lector de uso comercial, donde se pueden mostrar los datos leídos de una etiqueta adherida al vehículo, estos datos pueden ser almacenados en una base de datos, la cual tiene en su contenido la información de vehículos, propietarios y etiquetas, la información obtenida del lector es comparada con la que se encuentra en la base de datos y mostrada al usuario.

Para resolver esto, debemos considerar que el software sea capaz de brindar seguridad al usuario en general, mostrando la información alojada en la base de datos del software y haciendo una comparación con los datos leídos a través del lector de etiquetas RFID-EPC.

1.4. Justificación de la tesis

Se desea diseñar un software confiable y que permita al usuario utilizar los avances de la tecnología RFID-EPC para su beneficio personal.

⁴ La Cumbre CeBIT (Welt Centrum für Büro Informations und Kommunikations technick, en alemán, o Centro Mundial para la Tecnología de la Información y la Oficina), celebrada anualmente en Hanover (Alemania), se considera la mayor feria comercial de la tecnología de la información, así como un barómetro de ese sector.

Además, se quiere probar que esta tecnología es confiable, para ello se utiliza un estándar (EPC Global⁵), en el cual esta basada la tecnología EPC, este estándar se puede utilizar para un fin específico, como es este caso, la identificación de vehículos a través de etiquetas de RFID-EPC [EPC Global 2007].

La información que se obtiene del lector se utiliza para interactuar con las base de datos. La Figura 1.6. muestra en un panorama muy general del sistema desarrollado.

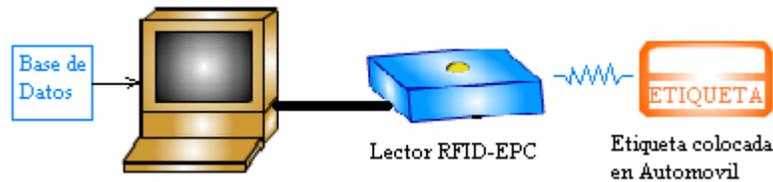


Figura 1.6. Sistema Desarrollado

1.5. Objetivos

Los objetivos General y específico, están ligados, ya que con ellos se logra implementar una aplicación que permita llevar un control de las etiquetas leídas de los vehículos.

1.5.1. General

Desarrollar un software que permita la detección de etiquetas EPC colocadas en vehículos, utilizando la tecnología de identificación por radio frecuencia (RFID-EPC) y asociándolo a una base de datos para registrar entradas y salidas.

1.5.2. Específicos

1. Implementación de un software para leer los datos de una etiqueta EPC localizada en un vehículo.
2. Incorporar una base de datos para guardar información sobre las etiquetas leídas.
3. Desarrollo de una relación entre la base de datos y la información obtenida del lector.

⁵ EPC Global: Es una organización no lucrativa a quien la industria ha confiado el establecimiento y soporte de la Red EPC como la norma global para la identificación inmediata, automática y exacta de cualquier artículo nuevo en la cadena de suministro de cualquier compañía, de cualquier industria, en cualquier parte del mundo.

4. Realizar una comparación con los datos leídos del lector con los de la base de datos y mostrarlos.
5. Trabajar con un estándar de calidad (EPC Global).

1.6. Organización de la tesis

En base a lo anterior, el propósito de este trabajo es diseñar y construir un software, que permita interactuar con una base de datos y las lecturas tomadas de un lector RFID-EPC, enfocado al paso de vehículos. Con este fin se ha organizado el presente documento en seis capítulos, los cuales se describen a continuación:

En el capítulo 1: Se presenta una descripción de la identificación de objetos por RFID, se mencionan los elementos que conforman un sistema de RFID, se describen los aspectos más relevantes de las etiquetas, de los lectores y los principios básicos de funcionamiento, de ese modo se presenta un problema actual, que es resuelto por una serie de objetivos: general y específicos.

En el capítulo 2: Se presenta información de trabajos similares, describiendo dos casos en los cuales las empresas utilizan etiquetas de RFID para el control de acceso de sus vehículos, también se describen las limitaciones de estas alternativas, de ese modo se propone una solución que las mejorará y se hacen las consideraciones correspondientes, además se menciona el equipo necesario para esta solución propuesta, también se muestra la interfaz que debe tener este equipo y una evaluación de la fabricación.

En el capítulo 3: Se define la propuesta planteada en este trabajo, mostrando cómo se lleva a cabo la solución, mencionando todos los elementos necesarios para su desarrollo. Se muestra el diseño del sistema, se describe cual es la metodología que utiliza y dentro de ella, describir la solución y finalmente se concluye con el diseño del software.

En el capítulo 4: Se realiza la implementación de este sistema, las características de las herramientas que se utilizan para implementar el sistema, la comunicación que se debe tener entre una computadora y el lector, así como una implementación con una base de datos.

En el capítulo 5: Se describen las pruebas para el software, se mencionan con la ayuda de capturas de pantalla cuál es el funcionamiento que deben tener cada uno de las funciones del sistema y se presentan los resultados obtenidos.

En el capítulo 6: Se presentan las conclusiones finales de todo el trabajo de investigación, mencionando algunas posibles mejoras que se pueden desarrollar a partir de este proyecto.

Se presenta un glosario, el cual tiene la finalidad de describir a detalle algunos de los términos utilizados en el presente trabajo.

Al final se presenta la bibliografía recopilada durante el desarrollo de este trabajo.

Por último se presentan cuatro anexos, los cuales tienen la finalidad de mostrar el código de programación utilizado en este trabajo, el manual de operaciones del software creado, las especificaciones del lector empleado y el Diagrama de Clases del software, mostrado a detalle.

En el próximo capítulo:

Se presentan algunas soluciones al problema por parte de empresas encargadas de proporcionar sistemas RFID, también se describe el motivo por el cual se considera que estas opciones no son la mejor alternativa para nuestro problema y por qué se tomó la decisión de construir un nuevo software en vez de utilizar alguno que se encuentre en el mercado.

Capítulo 2

Trabajos relacionados de RFID en vehículos

Este capítulo presenta varios trabajos relacionados con el tema de tesis, en los cuales se muestran algunas alternativas al problema y el por qué se considera que el trabajo propuesto es la mejor solución.

2.1. Trabajos relacionados con la identificación de vehículos

Compañía Sic TransCore¹ - SicFlotas

El seguimiento de vehículos y cargas es decisivo a la hora de brindar un buen servicio al cliente y operar eficientemente con el transporte.

SicFlotas es un sistema de identificación vehicular basado en el uso de tecnología inalámbrica (lectores de radio frecuencia (RFID) y etiquetas), que permiten a las compañías con flotas de vehículos realizar una identificación de la unidad o el conductor [Sic 2007].

Luego, se ubican las antenas lectoras en puntos estratégicos. Ejemplos: Accesos, estaciones, puertas, sitios de carga de combustible o de mantenimiento. Cuando la etiqueta pasa por la zona de lectura identifica al móvil y el lector retransmite la información programada al centro de control. La etiqueta está especialmente diseñada para aplicaciones en las que se opera con rangos largos y admite la exposición a ambientes severos.

La Figura 2.1. Muestra del uso de sus lectores en los camiones de carga.



Figura 2.1. Detección de camiones de SicFlotas

La Figura 2.2. Muestra las características técnicas de los lectores SicFlotas.

De las características más relevantes que se pueden apreciar en esta Figura destacan:

- El uso de dos frecuencias 902 a 904 Mhz ó 909.75 a 921.75 Mhz.
- Alcance regulable para etiquetas con batería.
- Uso de etiquetas sin batería (opcional).
- Superficie de montaje no metálica.

¹ SicTransCore: Es una empresa con base en la Argentina, dedicada a fabricar Sistemas de Control para Estacionamientos / Parking, Control de Accesos vehiculares, Cobro y Administración de Estaciones de Peaje.

Especificaciones Técnicas	
ANTENA:	TAG AT51XX
- Frecuencia de trabajo : 902 a 904 MHz o 909.75 a 921.75 MHz.	- 845 - 950 Mhz y 2400 - 2500 Mhz.
- Técnica empleada : backscatter.	- Alimentación : batería interna de litio
- Polarización : horizontal.	- Opcional (TAG sin Bateria)
- Alcance : regulable desde 0.9 a 9 metros para una velocidad máxima de 150 km/hora con el uso de tags con batería.	- Polarización : Horizontal.
- Temperatura de trabajo : -40°C a 50 °C	- Grado de protección : IP-56
- Humedad : 100 % con condensación.	- Superficie de montaje : No Metálica
- Vibración : 0,5 G de 10 a 500 Hz.	- Temperatura de trabajo : -40° - 85° c.
- Grado de protección : IP-66	
- Alimentación : 220 V, 50Hz/ 30 W.	

Figura 2.2. Especificaciones Técnicas del lector SicFlotas

Tal como se muestra en la Figura anterior, el uso de esta tecnología se basa en la utilización de detección de objetos de forma inalámbrica y está diseñada para el control de vehículos de carga, es un diseño de una empresa Sic TransCore.

Compañía GIGA-TMS²

Otra empresa que se encarga también de crear lectores para la detección de vehículos, es GIGA-TMS [GIGA-TMS 2007].

Con un modelo GP90-A, controla el acceso, con detección de vehículos a una distancia de 90 centímetros. Utiliza etiquetas RFID de 64 bits.

Posee un led indicador de encendido y lectura, posee un diseño capaz de soportar cambios de temperatura. La Figura 2.3. muestra el lector GP90-A de esta compañía.



Figura 2.3. Lector GP90-A de la compañía GIGA-TMS

² GIGA-TMS: Es una empresa con 20 años de experiencia, la cual es encargada de crear dispositivos RFID, actualmente su mercado de trabajo está en Europa, se encarga de importar y exportar dispositivos de seguridad, manufacturar tarjetas inteligentes. Fundada en 1987.

Las especificaciones técnicas del lector GP90-A se muestran en la Tabla 2.1.

Tabla 2.1. Especificaciones Técnicas del lector GP90-A

Especificaciones:	
Interfaces	MSR ABA Track2 RS232 RS485 Ó especificaciones especiales para los clientes.
Rango de lectura	Más de 90 cm. con tarjeta ISO en condiciones ideales ó más de 130 cm. con tarjeta clamshell especial.
Tarjetas de RFID aceptadas	125 Khz., 64 bits, codificación Manchester.
Nivel de energía	Más de 24 V / 2 A
Temperatura de trabajo	-10 a +60°C
Indicaciones Audio/Visuales	Luz azul brillante y timbre
Dimensiones	420 (W) x 320 (L) x 45 (D) mm.

2.2. Limitaciones de los trabajos relacionados

Dentro de las limitaciones que se detectan en las dos empresas es que están diseñadas para montarse en camiones de carga, con la ayuda de placas (que no es muy práctico ya que debe ser portable para aquellos vehículos que no requieren de una etiqueta fija).

Para el caso de la compañía Sic TransCore – SicFlotas:

- El principal problema que se encontró fue la estandarización, debido a que Sic TransCore no sigue un estándar (por ejemplo: EPC), con ello lograr que el software, sea aceptado fácilmente por parte de los usuarios, debido a que podrá hacer uso de etiquetas o lectores debidamente documentados, probados por empresas que trabajen con el mismo estándar.

Para el caso de la compañía GIGA–TMS:

- El principal problema que se observó fue su rango de lectura, debido a que trabaja con etiquetas RFID de alta frecuencia (125Khz), esto hace que la distancia de lectura entre la etiqueta y el lector sea muy reducida

aproximadamente un metro, aunque esta empresa asegura que su rango de lectura es de 130 centímetros., no es suficiente.

2.3. Solución propuesta

La solución es utilizar un lector comercial que se encuentre dentro del estándar EPC y diseñar una aplicación que permita administrar el acceso de los vehículos y proporcione una comunicación con una base de datos, para llevar el registro de estos.

Un ejemplo del diseño propuesto es el que se muestra en la Figura 2.4, donde se puede apreciar cómo el lector emite ondas de radio frecuencia para detectar la etiqueta que está en el vehículo.

De ese modo la etiqueta puede responde al estímulo regresando información de su identificación, los datos son recibidos por el lector, ser consultados en una base de datos de una computadora y de esa forma se puede mostrar información contenida en la base de datos de ese vehículo en particular.

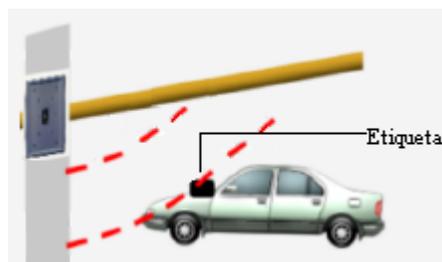


Figura 2.4. Detección de Vehículos por Radio Frecuencia

Debido a que se pretende utilizar un estándar para este sistema, no se puede utilizar el diseño de la compañía Sic TransCore – SicFlotas, y como la detección de vehículos generalmente no se hace en distancias cortas, no se puede utilizar el sistema de la empresa GIGA–TMS.

2.3.1. Consideraciones del proyecto

Es factible implementar el proyecto debido a que es posible asociar la información obtenida del lector RFID-EPC con una base de datos, además de que la implementación de las etiquetas en los vehículos no es muy costosa.

Si se utilizan etiquetas de clase 1 (RFID - EPC), para la identificación de vehículos, donde éstas, tienen un rango máximo de lectura de 3 metros, entonces es factible emplearlas para el diseño de la aplicación, debido a que si se considera que un

vehículo cuando pretende acceder a una empresa, escuela, etc. (lugar donde se encuentra el software a desarrollar) debe aproximarse a la entrada (es donde debe colocarse el lector), es donde el lector hace su trabajo. Obteniendo la identificación de la etiqueta y comparándola con la de la base de datos, para esta aproximación del vehículo, la distancia de 3 metros es más que suficiente.

2.3.2. Análisis técnico

Para llevar a cabo este proyecto se utilizó:

1. Un lector RFID-EPC, marca AWID, modelo 2010R.
2. Una etiqueta EPC Clase 1 (por vehículo).
3. Una computadora con puerto serial (R232).
 - 3.1. Un Convertidor R232 a USB (usado solo si la computadora no tiene un puerto R232).
4. La plataforma de desarrollo (Sistema Operativo Windows)
5. El lenguaje de programación (Java de SUN Microsystems)
6. El software creado.
 - 6.1. La base de datos creada por el software.

2.3.3. Interfaz del entorno

Se propone una interfaz de visualización en ambiente Windows, debido a que el mayor número de computadoras que se encuentran en el mercado tienen el Sistema Operativo Windows.

El software, trabaja en un ambiente virtual Java³.

2.3.4. Evaluación de la fabricación

Debido a que el Software es de fácil implementación y manejo, se puede utilizar en casi cualquier lugar que cuente con una persona que lo controle y a la vez permita el paso a los vehículos, ejemplos: en empresas privadas con personal de seguridad en la entrada, en escuelas con personal de vigilancia que permita el acceso, pensiones de vehículos, almacenes, etc.

³ Java: Lenguaje desarrollado por Sun Microsystems para la elaboración de aplicaciones exportables a la red y capaces de operar sobre cualquier plataforma a través, normalmente, de visualizadores.

En el próximo capítulo:

Se muestra un análisis de lo necesario para la construcción del software, también se describe la forma en que se desarrolló, la descripción del ciclo de vida utilizado, también se presenta a través de diagramas cuáles son los elementos necesarios para hacer funcionar (internamente y externamente) el software.

Capítulo 3

Análisis y diseño de la aplicación

En este capítulo se presenta la forma en que se diseñó la aplicación, en el se muestran diferentes diagramas de modelado los cuales sirven para describir de una forma clara el funcionamiento del software.

3.1 Análisis de la solución

Se considera que los clientes pueden ser todas aquellas empresas, escuelas, almacenes o simplemente aquellos lugares que cuenten con un servicio de vigilancia para control de acceso a él, la aplicación se encarga de registrar las entradas y salidas de los vehículos, la ventaja de este diseño es que ayuda a la pronta identificación del automóvil.

El software incluye lo siguiente:

1. Una interfaz gráfica eficiente.
2. Un desarrollo que cubra los requerimientos establecidos.
3. Seguridad en su base de datos.
4. Automatización de procesos.
5. Ofrecer un estándar de funcionalidad.
6. Una implementación en un lenguaje de programación orientado a objetos.
7. Bajo costo, debido a que se ocupa equipo comercial.
8. Un alcance suficiente de lectura.

Las entradas al software desde el punto de vista del usuario son las siguientes:

1. El operador puede buscar etiquetas para leer, desde la ventana principal.
2. El operador puede consultar la temperatura del lector de etiquetas.
3. El operador puede verificar la versión del lector de EPC y de la versión del software.
4. La aplicación puede actualizar la base de datos, cuando lee una etiqueta.
5. El administrador puede modificar la base de datos del sistema.

Las salidas del software vistas desde el usuario son:

1. Información de lecturas realizadas.
2. Registros encontrados en la base de datos.
3. Todas las respuestas de los comandos o consultas de las entradas del software.

Todas las entradas y salidas son importantes que existan y tengan interacción con el software de manera directa.

Las especificaciones que a continuación se mencionan, hacen referencia a aspectos importantes que se deben considerar para la elaboración del sistema.

Especificaciones funcionales

El software interactúa con el usuario de manera directa, proporciona información de todas las consultas realizadas, muestra información alojada en la base de datos, la

cual sirve para que el usuario del sistema verifique que el vehículo puede pasar por la puerta del lugar donde se utiliza este sistema. (Hay que recordar que la decisión final es del vigilante).

Especificaciones de información

La información debe ser clara y precisa, el usuario debe ingresar la menor cantidad de información al sistema, todo se hará por medio de opciones o botones (exceptuando la interacción con la base de datos, esta opción solo está disponible para el administrador del sistema).

Especificaciones de eventos

Cuando se aproxima un vehículo al rango de alcance del lector, se mandará una alerta con el Id de la etiqueta y si se encuentra información en la base de datos de esta etiqueta, se mostrará esa información también.

Además se mandarán mensajes o alertas de comandos empleados y resultados de acciones realizadas.

Especificaciones no funcionales

Se considera que el software debe cubrir con todas las funciones necesarias para la correcta funcionalidad y visualización de la información, por ello no existen funciones que no sean tomadas en cuenta.

3.2 Arquitectura de la aplicación

La representación de la arquitectura del software, facilita la comunicación entre todas las partes interesadas en el sistema, además de destacar decisiones tempranas de diseño.

Se incluye el conjunto de modelos y técnicas de especificación, utilizadas para especificar los requisitos anteriormente descritos.

Algunos enfoques que se utilizan para el análisis del software, son:

- Enfoque desde el punto del usuario.
- Enfoque desde el punto de vista del sistema y sus mensajes entre procesos.
- Enfoque desde el punto de vista de objetos y clases.
- Enfoque desde el punto de vista de las actividades que se realizan.

Estos análisis permiten ver el problema desde varios enfoques, los cuales proporcionan una mejor solución, para con ellos hacer un análisis detallado, ayudando a desarrollar la aplicación, sin que se escape algún aspecto importante.

Para el diseño de la aplicación se elige un ciclo de vida Lineal Secuencial [Pressman 2002], el cual dictará las fases para la construcción del software.

El ciclo de vida Lineal Secuencial sugiere un enfoque sistemático, secuencial, para el desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

3.2.1. Casos de Uso General y Principales

Los diagramas de caso de usos se emplean para visualizar el comportamiento de un sistema, un subsistema o una clase, de forma que los usuarios puedan comprender cómo utilizar ese elemento y de forma que los desarrolladores puedan implementarlo.

Para comprender como trabaja el software, se desarrolló un diagrama de Caso de Uso que muestre cómo interactúan los usuarios con el sistema, este es mostrado en la Figura 3.1.

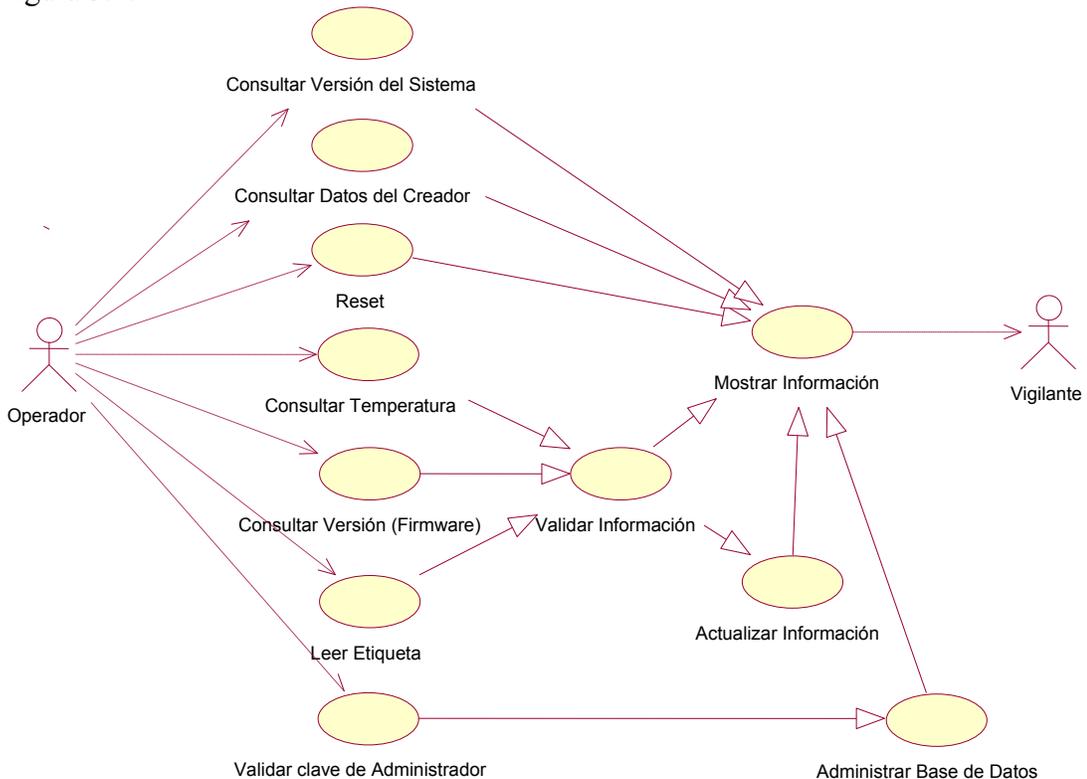


Figura 3.1. Diagrama de Caso de Uso General

El operador del software, entre las acciones que puede realizar son:

- Consultar la versión del sistema.
- Consultar los datos del creador.
- Realizar un Soft Reset en el lector.

Las siguientes tres operaciones son mostradas en el sistema una vez ejecutas las instrucciones correspondientes para cada uno de ellas.

- Consultar la temperatura el lector.
- Consultar la versión del lector (Firmware).
- Leer una etiqueta.

Estas tres operaciones requieren de procesos extras para visualizar el correcto resultado de cada uno de ellas

Una de estas operaciones extras es verificar que la información recibida se valide antes de mostrarse al vigilante, otra operación es la de actualizar la información que se refiere a registrar entradas y salidas en la base de datos, esta operación se utiliza en la opción de Leer Etiqueta.

Tanto el operador como el vigilante son los actores que interactúan con el software, puede darse el caso de que una sola persona realice estas actividades.

Existe un caso particular del operador, es cuando esté es el administrador del sistema en ese caso solo él tiene acceso a la ventana que administra la base de datos y solo él puede modificarla, esto se consigue después de una validación de su clave de acceso.

Como el proceso de Leer Etiqueta, es el más complejo, a continuación se describe a detalle, en la Figura 3.2. el diagrama de Caso de Uso del proceso de lectura que presenta el proceso de Leer Etiqueta, aquí se hace una validación de la información obtenida del lector con el fin de mostrar un mensaje de alerta de error si es que el resultado no es el esperado (Mostrar Información), en caso contrario, se busca la existencia del Id de la etiqueta con los registros almacenados en la base de datos del software y se procede a crear un registro en la base de datos el cual corresponde a una entrada o salida del vehículo según sea el caso (Actualización Información), después se procede a mostrar la información almacenada en la base de datos referente a la etiqueta que fue leída, presentando los resultados al vigilante. Esta información es usada para la toma la decisión de dar acceso al vehículo o no.

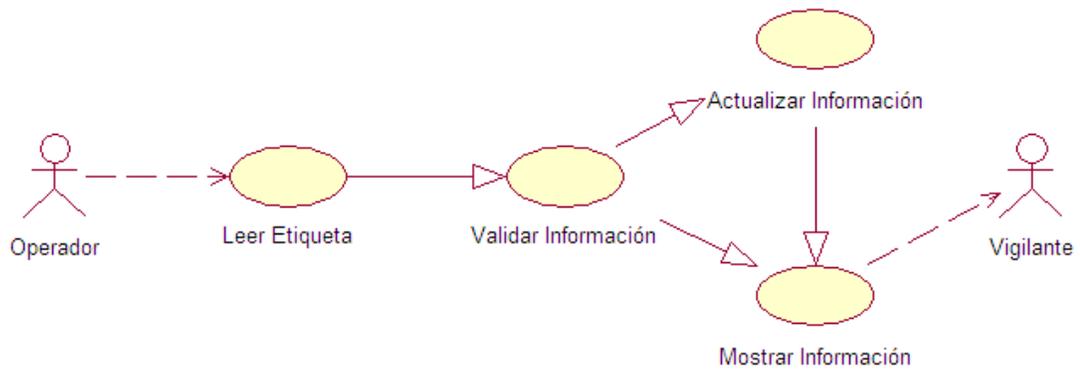


Figura 3.2. Diagrama de Caso de Uso del Proceso de Lectura de una Etiqueta

3.2.2. Diagramas de Secuencias

En un diagrama de secuencia se destaca la ordenación temporal de los mensajes.

Debido a que se quiere mostrar a detalle la forma en que se envían los mensajes entre las clases, a continuación se describen los Diagramas de Secuencias de acuerdo a los Diagramas de Casos de uso anteriores.

Este tipo de diagramas permiten mostrar el orden y sentido que toman los mensajes cuando se encuentra una instrucción o proceso.

La Figura 3.3. Muestra el diagrama de Secuencias del proceso de Lectura de una Etiqueta

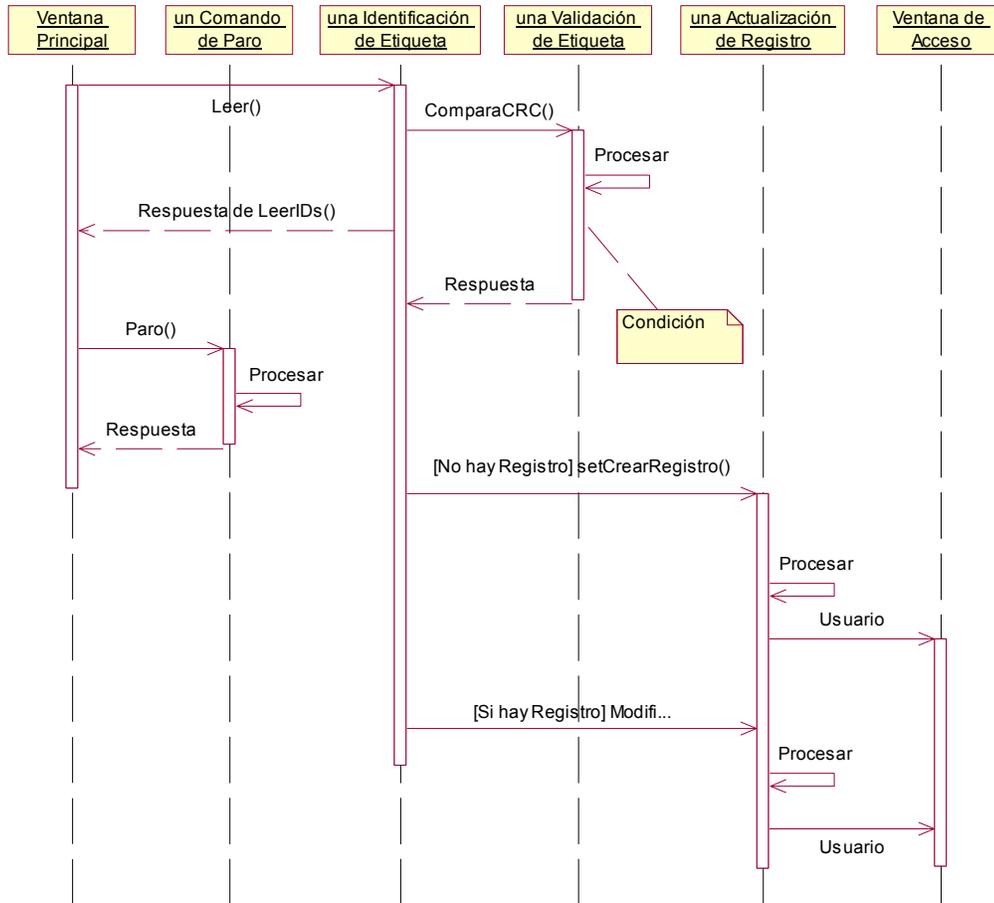


Figura 3.3. Diagrama de Secuencias del Proceso de Lectura de una Etiqueta

Desde una ventana principal, se ejecuta la orden de Leer, la cual se encarga de identificar la etiqueta, después la orden verifica que la información obtenida sea válida y que se encuentre la información de la Id en la base de datos, y se actualizará el registro de acceso, mostrando la información al vigilante quien dará o no acceso al vehículo.

Este diagrama muestra un comando adicional, el comando de Paro, el cual permite detener múltiples lecturas por parte del lector, ya que éste no se detendrá de enviar la información leída hasta recibir este comando. (De acuerdo al manual del lector).

A continuación la Figura 3.4. muestra cómo se realiza el proceso de consultar la temperatura del lector y puede apreciarse que al ejecutar el comando Temperatura(),

se espera recibir una respuesta 00 ó FF (ambos en hexadecimal). En caso de ser 00 se recibirán los bytes necesarios para procesar la temperatura deseada, después se valida esta información (ComparaCRC()) y de la respuesta obtenida se realiza el cálculo de la temperatura. En caso de recibir un FF se manda un mensaje de error al usuario del sistema, indicando que hubo un problema en la operación realizada.

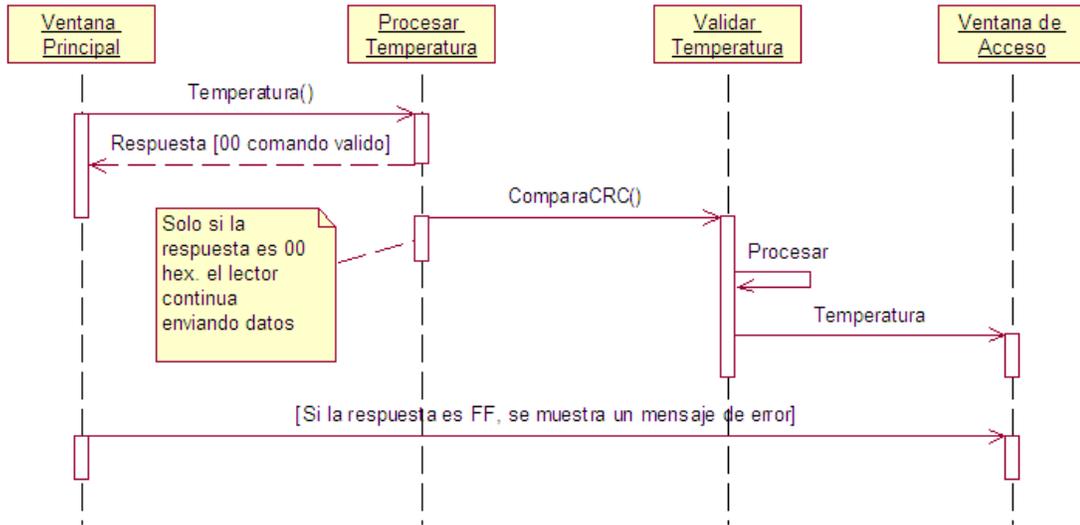


Figura 3.4. Diagrama de Secuencias del Proceso de Consultar Temperatura

La Figura 3.5. Muestra el diagrama de secuencia del proceso de consultar la versión del lector (Firmware).

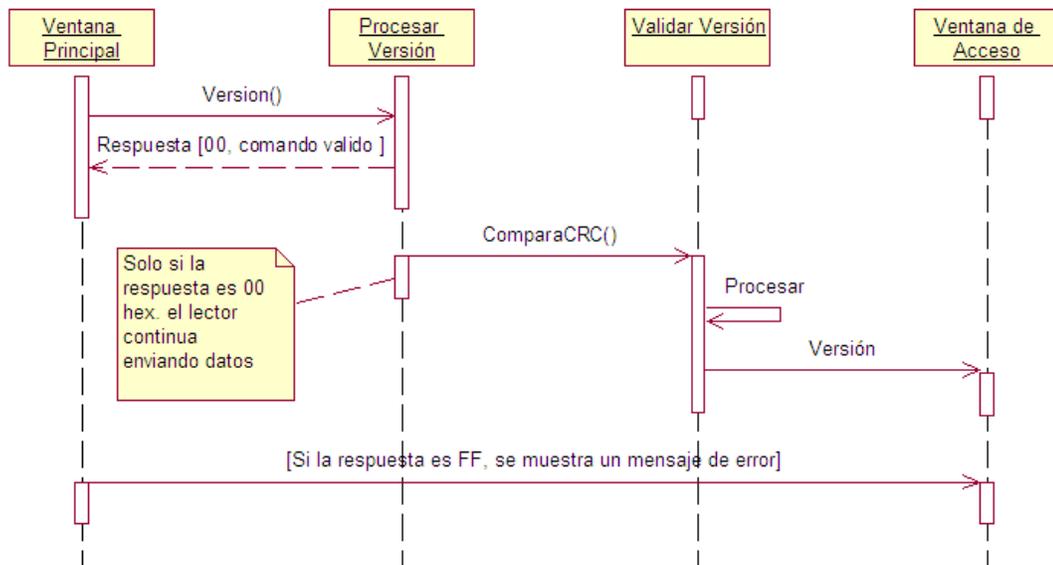


Figura 3.5. Diagrama de Secuencias del Proceso de Consultar Versión

Al ejecutarse el comando Versión(), el lector responderá con un 00 ó un FF (hexadecimal). Al recibir un 00 se espera que a continuación se envíen los bytes necesarios para mostrar el Firmware, pero antes se debe validar esta información (ComparaCRC()), después se envía el mensaje correspondiente a la ventana de acceso (mensaje que contiene la versión del lector). En caso de recibir un FF se envía un mensaje indicando que ocurrió un error en este proceso.

A continuación se presenta el Diagrama de Secuencias del proceso de Soft Reset del lector, este se muestra en la Figura 3.6. que después de ejecutar el comando Reset (Soft Reset), el lector responde con un 00 si recibe el comando correctamente e inmediatamente después ejecuta el comando y manda un mensaje el cual se muestra en la ventana de acceso.

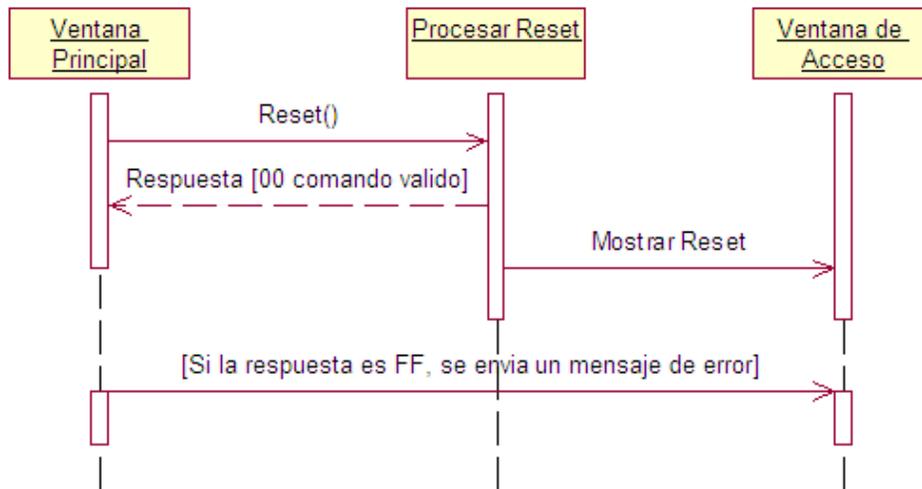


Figura 3.6. Diagrama de Secuencias del Proceso de Soft Reset del lector.

Para los casos de los procesos consultar la versión del software y consultar los datos del creador, el proceso es muy similar al caso descrito anteriormente, solo que con una diferencia, como en estos casos la comunicación no es con el lector sino con la misma aplicación, solo es enviado el comando y se procede a mostrar la información solicitada.

A continuación se presenta el Diagrama de Secuencias del proceso de validar la clave de administrador, este proceso permite que el operador (en este caso el administrador del sistema), pueda modificar la base de datos del sistema, con la ayuda de la ventana de Registros Almacenados, véase la Figura 3.7.

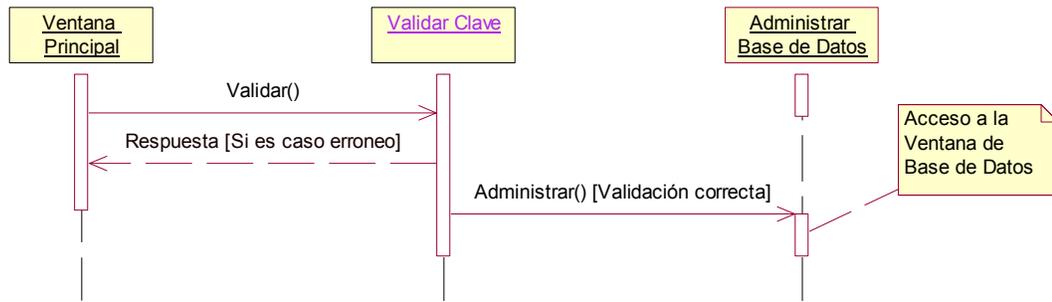


Figura 3.7. Diagrama de secuencias del proceso de validar clave de administrador.

3.2.3. Diagramas de Colaboración

Además de los mensajes entre procesos el Diagrama de Colaboración muestra los mensajes que se envían los objetos¹ entre sí.

La siguiente Figura 3.8., muestra el Diagrama de Colaboración del proceso de Leer Etiqueta. En él se aprecia la secuencia que tienen los mensajes. La secuencia numérica facilita la comprensión.

¹ Un objeto es una representación detallada, concreta y particular de un "algo". Tal representación determina su identidad, su estado y su comportamiento particular en un momento dado. (<http://es.wikipedia.org/>)

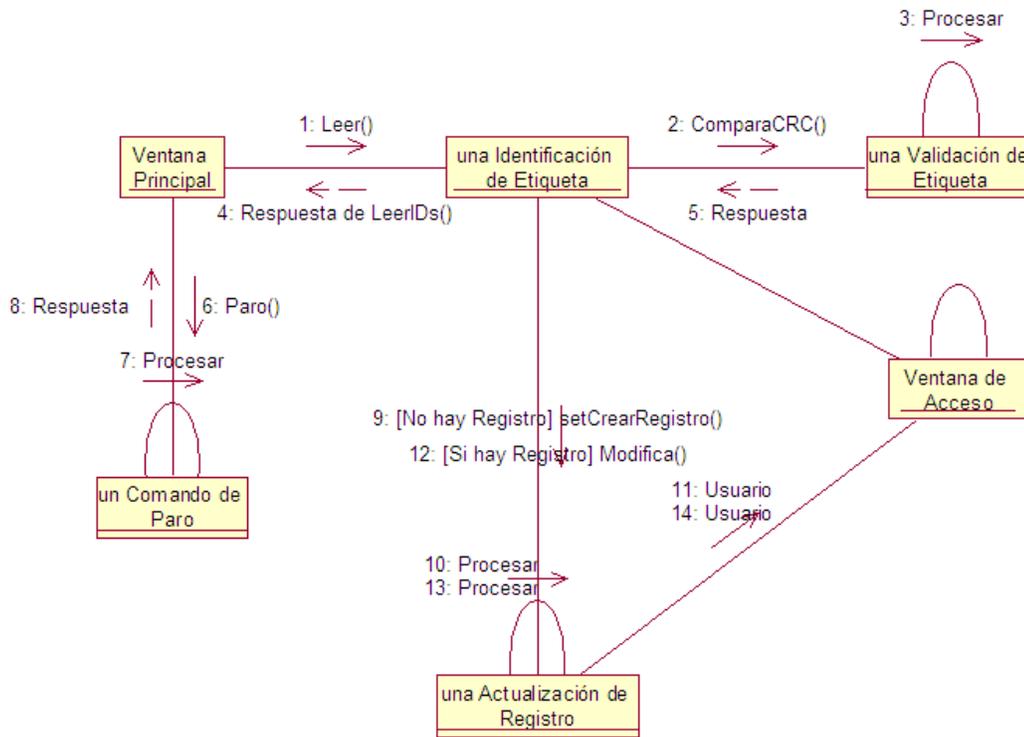


Figura 3.8. Diagrama de Colaboración del Proceso Lectura de una Etiqueta

La Figura 3.9. presenta la secuencia de mensajes entre el proceso de Temperatura. En este caso la secuencia numérica ayuda a comprender cuál es el orden que siguen los mensajes.

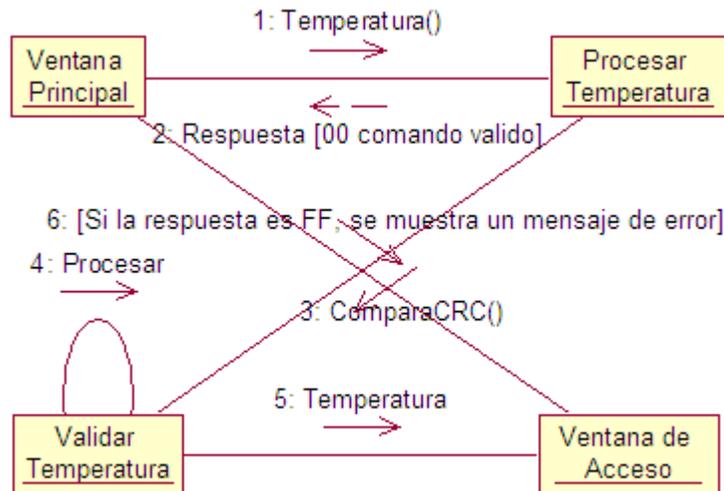


Figura 3.9. Diagrama de Colaboración del Proceso de Consultar Temperatura

La Figura 3.10. muestra el Diagrama de Colaboración del Proceso de Consultar Versión. Siguiendo la numeración que presenta la Figura, se puede ver la secuencia de los mensajes que se siguen en el proceso de consultar Versión.

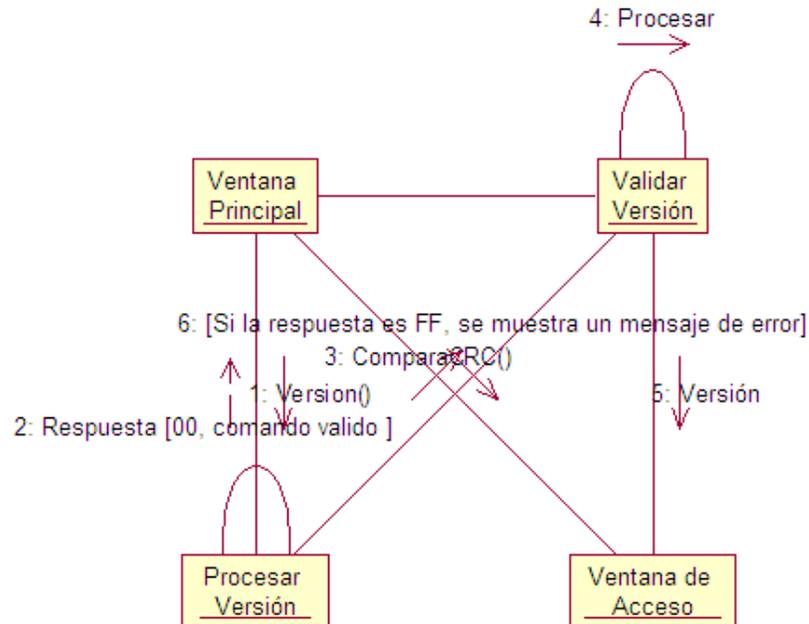


Figura 3.10. Diagrama de Colaboración del Proceso de Consultar Versión

La Figura 3.11. presenta el Diagrama de Colaboración del Soft Reset, este diagrama no tiene los mensajes 2 y 4 de la Figura 3.10. debido a que el comando de Reset no espera respuesta del lector, solo muestra la información enviada por el lector.

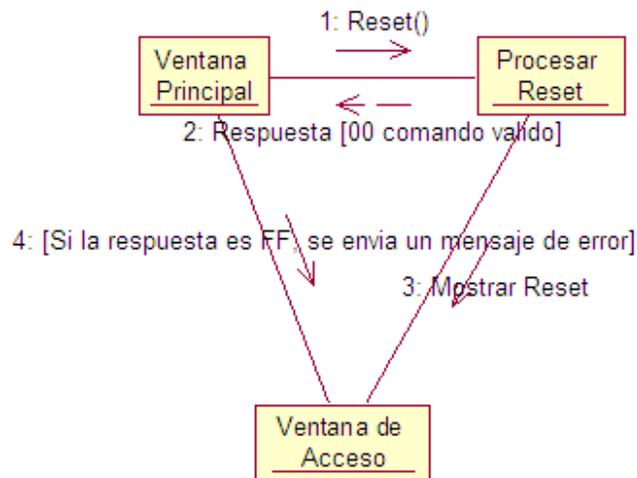


Figura 3.11. Diagrama de Colaboración del Proceso de Soft Reset del lector

El Diagrama de Colaboración del proceso de validar clave de administrador es mostrado en la Figura 3.12. en este diagrama se puede observar que solo el administrador del sistema tiene acceso a la base de datos del sistema.

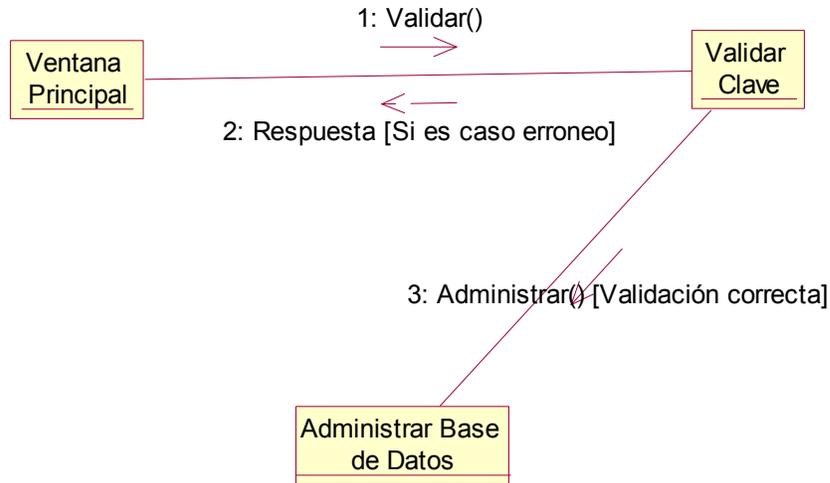


Figura 3.12. Diagrama de Colaboración del Proceso de validar clave de administrador.

3.2.4. Diagrama de Clases

Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema. Esta vista soporta principalmente los requisitos funcionales de un sistema, los servicios que el sistema debe proporcionar a los usuarios finales.

Cuando se modela la vista de diseño estática de un sistema, normalmente se utilizarán los diagramas de clases de unas de estas tres formas:

- Para modelar el vocabulario de un sistema.
- Para modelar colaboraciones simples.
- Para modelar el esquema lógico de una base de datos

La Figura 3.13. presenta el Diagrama de Clases del software, el cual muestra la estructura general del sistema, el cual contiene las siguientes clases:

- Acceso
- BDLector
- CRC16
- EscrituraPuerto
- EtiquetaClase1
- Fecha

3.2.5. Diagrama de Actividades

Un Diagrama de Actividades ha sido diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o proceso.

La Figura 3.14. muestra cómo funciona el sistema, a través de un Diagrama de Actividades, el cual proporciona una vista de los caminos seguidos para ejecutar una instrucción en cualquier proceso.

También se puede apreciar que tanto al iniciar como para terminar la ejecución del sistema, la Ventana Principal, es la última mostrada, esto es debido a que contiene los controles de entrada y salida de la aplicación.

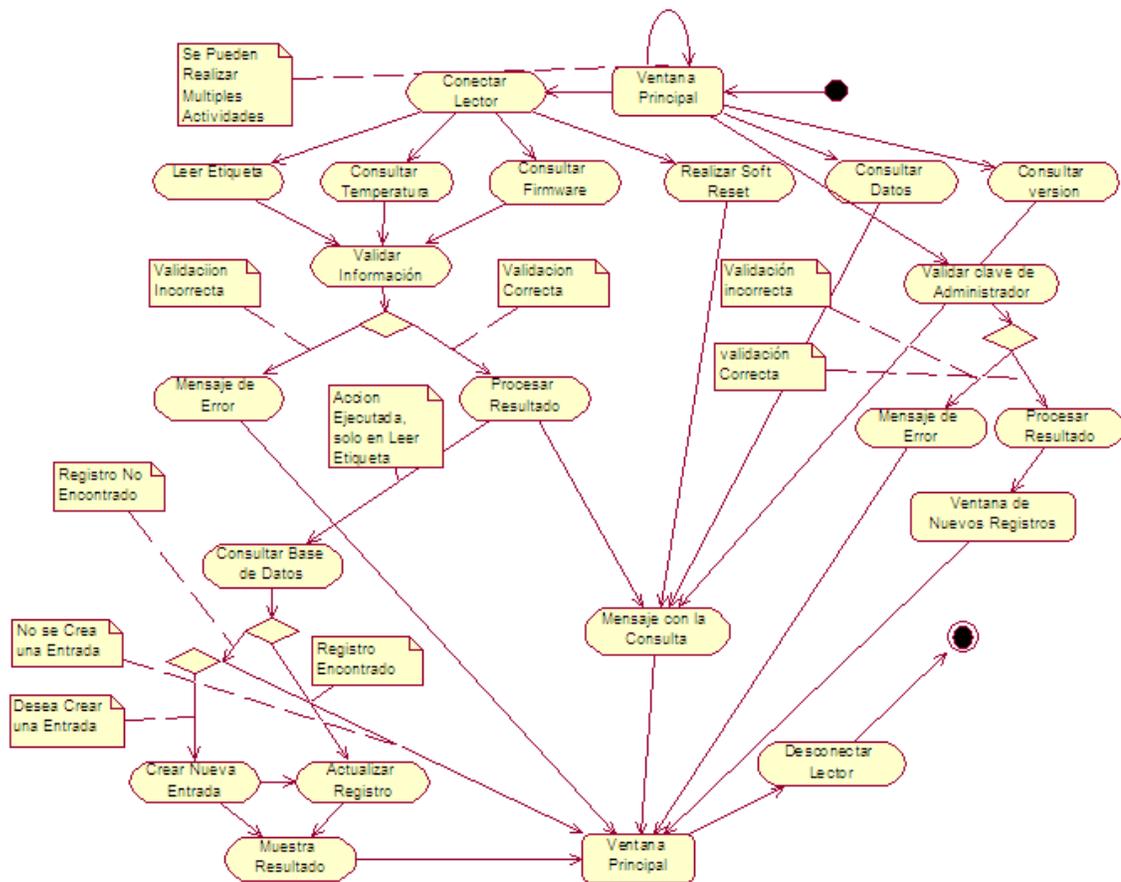


Figura 3.14. Diagrama de Actividades

3.3. Puntos clave del diseño

La Figura 3.15. representa un Grafo Convencional, el cual tiene la finalidad de mostrar los mensajes emergentes² que se visualizan en el software.

Los óvalos de la figura son llamados nodos, los cuales representan las ventanas y las flechas son llamadas aristas, las cuales representan el sentido en que va el flujo de la información.

El propósito de este grafo es ayudar con el diseño de la aplicación, mostrando los mensajes (o ventanas) emergentes, de errores y de éxitos en los procesos o comandos ejecutados por el sistema.

La numeración localizada dentro de cada nodo corresponde a las siguientes ventanas:

1. Ventana principal.
2. Desconexión del lector.
3. Conexión del lector.
4. Leer etiqueta.
5. Muestra Firmware.
6. Muestra temperatura.
7. Muestra Soft Reset.
8. Muestra datos del creador.
9. Muestra versión del software.
10. Validación de clave de administrador.
11. Validación de respuesta del lector.
12. Muestra el resultado ejecución de comando.
13. Etiqueta encontrada, muestra los datos relacionados con el Id de la etiqueta.
14. Etiqueta no encontrada en la base de datos, pregunta si se desea crear registro.
15. Ventana de registros almacenados, administra la base de datos del sistema.
16. Crea registro, muestra el registro de entrada o salida según sea el caso.
17. Crea nueva entrada de información en la base de datos, de la nueva etiqueta.

Los números que se encuentran en las aristas del grafo (flechas que salen de los nodos), representan las interconexiones o transiciones entre cada mensaje emergente.

Los * representan un camino obligatorio del sistema y se realiza de forma automática.

Las E, representan un mensaje de error el cual después de ser mostrado regresa al nodo que genero el mensaje.

² Los mensajes emergentes, son cajas de texto las cuales ayudan a mostrar cierta información al usuario de la aplicación, contienen información referente al resultado de una operación o un error en la ejecución de una instrucción.

La **S**, representa la salida o terminación de la aplicación.

La **A**, representa la entrada o inicio de la aplicación.

El nodo que se encuentra dentro del rectángulo punteado, representa la ventana de Registros Almacenados, sirve para modificar la base de datos (solo el administrador tiene acceso a ella).

En los nodos 13, 14, 16 y 17, los mensajes solo se presentarán si es que se está en la acción de Leer Etiqueta.

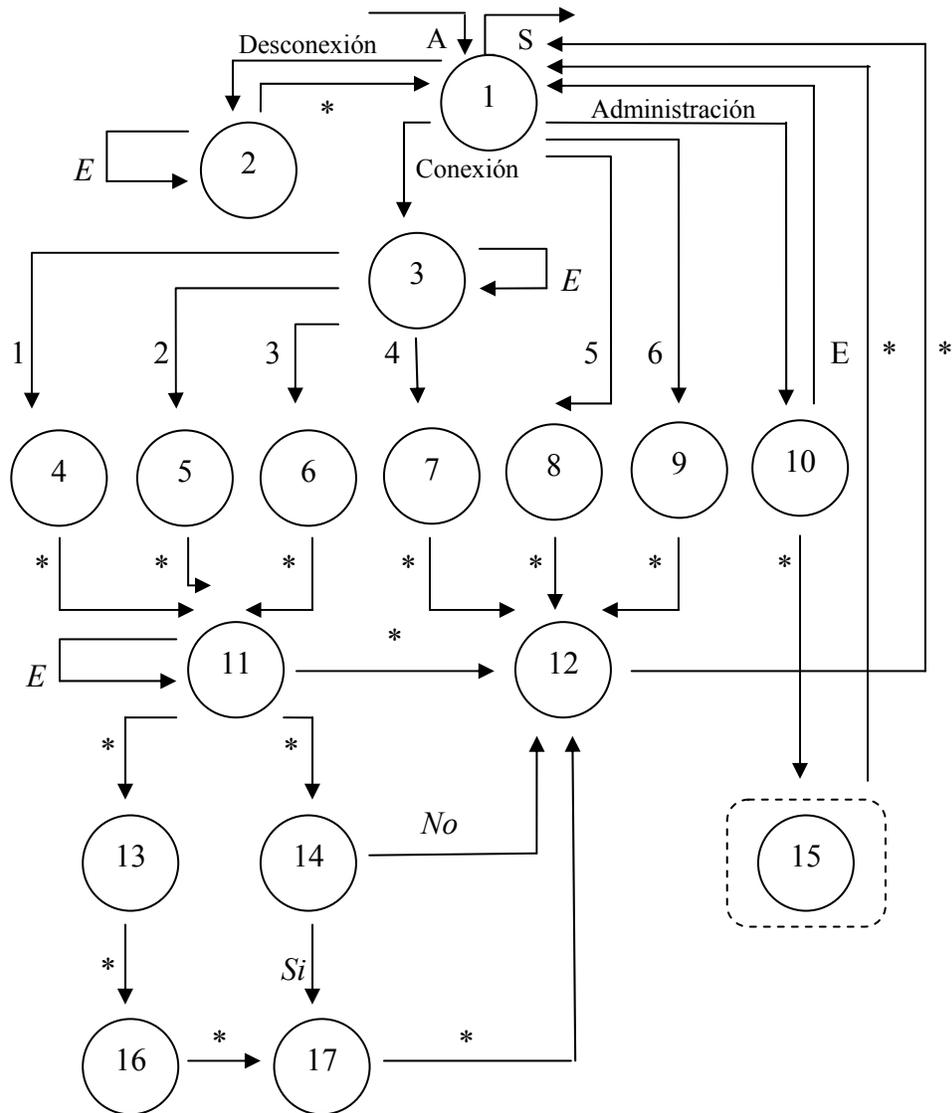


Figura 3.15. Grafo convencional del software

En el próximo capítulo:

Se presenta la implementación del software, algunas características principales que se deben tomar en cuenta referentes al lector y a las etiquetas, se explica a detalle la comunicación entre el lector y la computadora, así como la construcción del software y su base de datos.

Capítulo 4

Implementación de la aplicación

Este capítulo presenta el diseño de la base de datos y la implementación del sistema, así como la comunicación entre los diferentes elementos del sistema (base de datos, lector EPC, etiquetas EPC y computadora).

4.1. Creación de la base de datos

Estructuras de archivo y bases de datos.

Un Diagrama Entidad – Relación, es un lenguaje gráfico para describir conceptos. Informalmente, son simples dibujos o gráficos que (si se saben interpretar) describen la información que trata un sistema de información y el software que lo automatiza.

La Figura 4.1. muestra el contenido de la base de datos del sistema, para lo cual se construyó un Diagrama Entidad – Relación.

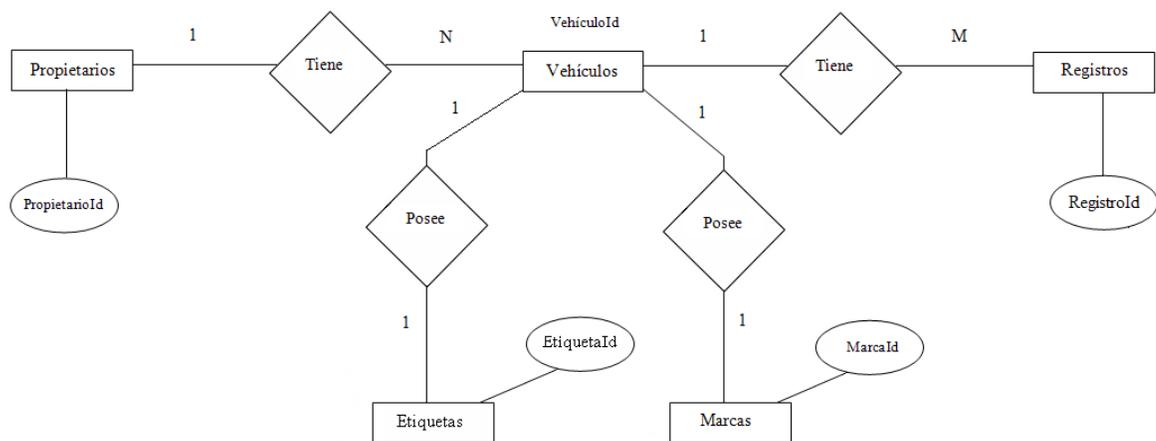


Figura 4.1. Diagrama Entidad Relación

De acuerdo al Diagrama Entidad Relación presentado anteriormente, se puede observar que un *propietario* posee al menos un *vehículo*, mientras que el *vehículo* puede tener varios *registros*.

Un *vehículo* posee una *marca* a la cual pertenece, así como una *etiqueta*, la cual sirve para identificarlo.

Para la elaboración de la base de datos, tuvo que pasar por varias fases de desarrollo, las cuales hacen que ésta no tenga redundancias y que todo su contenido pueda ser accedido por el sistema, por ello se desarrolla con un método llamado:

Normalización de la base de datos

Básicamente, las reglas de Normalización están encaminadas a eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas.

La Tabla 4.1. muestra los atributos que se consideran importantes que aparezcan en la base de datos del sistema.

Nivel Cero de Formalización / Normalización.

1. Se identifican los Atributos que deben tener las tablas de la base de datos.
2. Agruparlos de acuerdo a grupos similares.

Tabla 4.1. Nivel 0 de Normalización.

1	Marca
2	Modelo
3	Anio
4	Color
5	EtiquetaId
6	Propietario
7	Departamento
8	Password
9	FechaEntrada
10	HoraEntrada
11	FechaSalida

La Tabla 4.2. muestra el Nivel 1 de Normalización.

Primer nivel de Formalización/Normalización.

1. Eliminar los grupos repetitivos de las tablas individuales.
2. Crear una tabla separada por cada grupo de datos relacionados.
3. Identificar cada grupo de datos relacionados con una clave primaria.

Tabla 4.2. Nivel 1 de Normalización

Vehículo								
VehículoId	Marca	Modelo	Anio	Color	EtiquetaId	Propietario	Departamento	Password

Registro				
RegistroId	FechaEntrada	HoraEntrada	FechaSalida	HoraSalida

La Tabla 4.3. muestra el resultado de la Normalización de Nivel 2.

Segundo nivel de Formalización/Normalización.

1. Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
2. Relacionar estas tablas mediante una clave externa.

Tabla 4.3. Nivel 2 de Normalización

Vehículos						
VehículoId	relPropietarioId	Marca	Modelo	Anio	Color	EtiquetaId

Registros					
RegistroId	relVehículoId	FechaEntrada	HoraEntrada	FechaSalida	HoraSalida

Propietarios			
PropietarioId	Propietario	Departamento	Password

Una normalización más es presentada en la Tabla 4.4. la cual muestra el resultado final de la normalización, quedando las tablas que contendrá la base de datos del sistema.

Tercer nivel de Formalización/Normalización.

1. Eliminar aquellos campos que no dependan de la clave.

Tabla 4.4. Nivel 3 de Normalización

Vehículos						
VehículoId	relPropietarioId	relMarcaId	Modelo	Anio	Color	relEtiquetaId

Registros					
RegistroId	relVehículoId	FechaEntrada	HoraEntrada	FechaSalida	HoraSalida

Propietarios			
PropietarioId	Propietario	Departamento	Password

Marcas	
MarcaId	Marca

Etiquetas	
EtiquetaId	Etiqueta

Esta tabla muestra el contenido de las tablas de la base de datos, en la cual se puede apreciar que se emplean cinco tablas para almacenar toda la información para que el sistema pueda realizar su función.

Este diseño facilita el trabajo con la información que maneja el sistema, ya que al estar organizada de esta forma, permite que no exista redundancia en las bases de datos.

4.2. Implementación del software

Para la construcción del sistema, se utiliza un ambiente de trabajo que soporte Java, por lo cual se optó por NetBeans en su versión 5.5 [NetBeans 2007].

La decisión de porqué utilizar esta herramienta, es porque es la propietaria de la tecnología Java y por lo cual facilita la programación del software.

Esta plataforma es de libre distribución.

Otra herramienta necesaria es:

- J2SE Development Kit 5.0, Update 10 ó superior.

El J2SE de Java no es más que una aplicación que contiene:

- El Java Runtime Environment (JRE), que es la plataforma sobre la cual corren las aplicaciones realizadas en Java, en el ambiente Windows.
- Líneas de comandos y herramientas necesarias para programadores.

Éste se puede descargar libremente de la página de Sun Microsystems [Sun 2007]; la herramienta NetBeans ya contiene este programa, así que solo basta revisar la versión que se está instalando de NetBeans y del J2SE que viene incluido con él.

La Figura 4.1. muestra la instalación del software NetBeans.

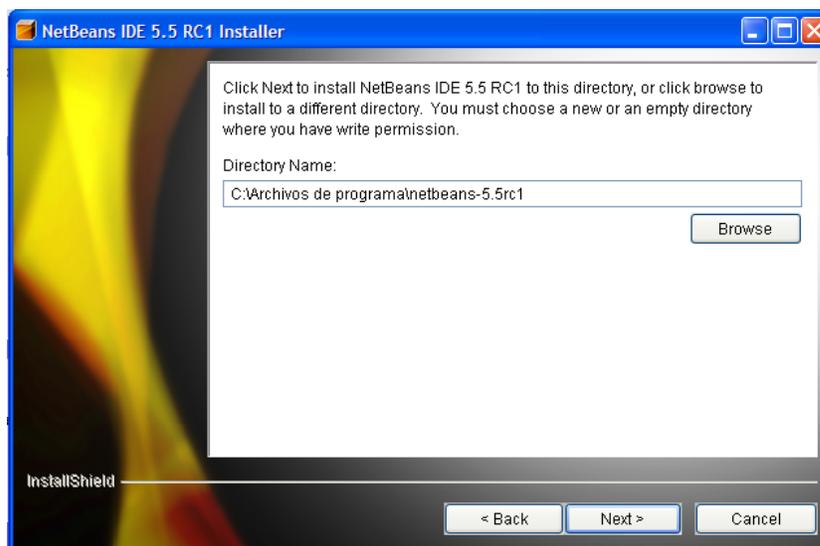


Figura 4.2. Instalación de NetBeans en Windows.

Otra herramienta necesaria es:

- El API de comunicaciones, que no es más que una extensión estándar que permite realizar comunicaciones con los puertos serie RS-232 de la computadora, en su versión 2.0.

El API, es un conjunto de archivos, distribuidos por Sun Microsystems, el cual ayuda a interactuar con el puerto serie de la PC.

Básicamente la descarga consiste de un solo archivo (en formato Zip), el cual contiene tres archivos más su documentación de uso.

Una vez desempaquetado el fichero se procede a copiar el fichero *Win32com.dll* a `\jre\bin`. Se coloca el archivo *comm.jar* al directorio `\jre\lib\ext`. Del mismo modo se coloca el archivo *javax.comm.properties* al directorio `\jre\lib`, éste contiene los drivers que soporta el API.

Estos directorios que se mencionan, se encuentran dentro del directorio raíz donde se instaló el J2SE.

Otra herramienta necesaria es:

- El conector JDBC para MySQL (Conector/J).

El conector/J es un driver nativo de Java que establece la conexión JDBC (Java Database Connectivity, Conectividad con las Base de Datos Java) que utiliza los protocolos necesarios para utilizar las bases de datos MySQL [Mysql 2007].

Éste es de libre distribución y se puede descargar de la página de su creador [Mysql 2007] el cual se encuentra actualmente en una versión 5.0.4.

El conector/J es distribuido en un solo archivo (formato Zip), el cual contiene entre otras cosas un archivo en formato JAR (Archivos Java Empaquetado). Éste debe ser colocado en el directorio `jre\lib\ext`, del directorio raíz donde fue instalado el J2SE.

Otra herramienta necesaria es:

- Base de Datos MySQL en su versión 5.0.27

MySQL es un sistema de gestión de base de datos, el cual es de libre distribución y se descarga de la página de su autor [Mysql 2007].

Esta herramienta se utiliza para construir la base de datos que ya fue modelada previamente en el tema de Creación de Base de datos (4.1.).

La Figura 4.3. muestra la instalación de MySQL, no es necesario instalar la versión completa, ya que solo instala ejemplos de uso de este programa, con la versión típica es suficiente.

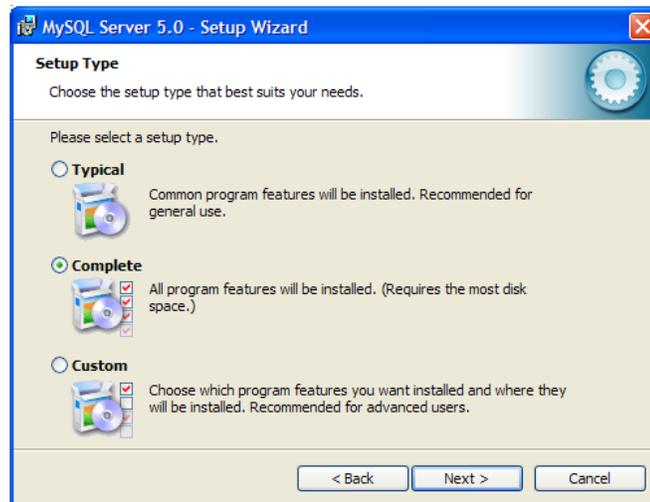


Figura 4.3. Instalación de MySQL en Windows.

4.3. Características del lector y las etiquetas

Los sistemas “long-range” [RFID 2007], que como su propio nombre indica son de largo alcance, (generalmente) mayores a 1 m. Estos sistemas se basan en el uso de ondas electromagnéticas en el rango de UHF o microondas.

La Figura 4.4. muestra el esquema utilizado, el sistema long-range, facilita el implementar los estándares EPC [Auto ID 2007], debido a que trabaja en el rango de Ultra Alta Frecuencia (UHF – 860 a 960 MHz).

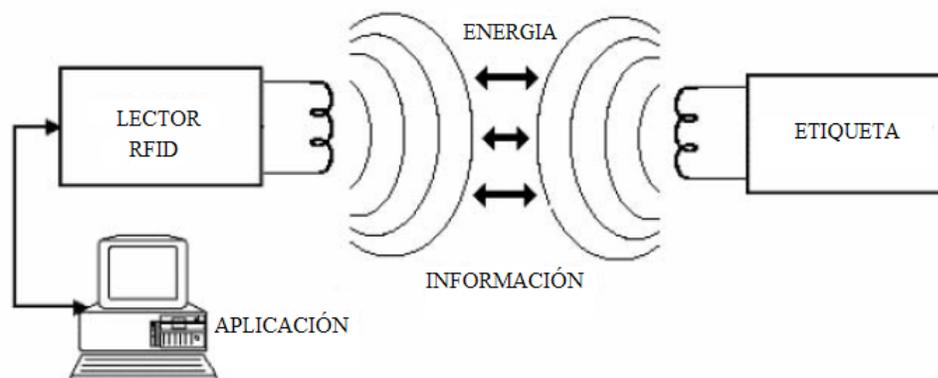


Figura 4.4. Comunicación entre el lector y las Etiquetas.

Para comunicarse con las etiquetas se necesita establecer previamente una comunicación con el lector y éste se encargará de interactuar con las etiquetas (utilizando el comando adecuado), para esto se utiliza el manual del lector, donde se muestra la estructura del paquete que se utiliza para interactuar con el lector.

Los comandos enviados desde la computadora al lector [AWID 2005], así como la respuesta del lector, siguen la estructura mostrada en la Tabla 4.5.

Tabla 4.5. Estructura del paquete

LONGITUD	TIPO	COMANDO	DATOS	CHECKSUM
(1) byte	(1) byte	(1) byte	(<= 27) bytes	(2) bytes

Donde:

- LONGITUD: Número total de bytes en el paquete.
- TIPO: Tipo de comando: categorizados en comandos de sistema (0x00), y específicos de tipo de etiqueta (0x16).
- COMANDO: Identificación de comando dentro de la categoría respectiva.
- DATOS: Dependiendo si es comando o respuesta, los datos van de 0 a 27 bytes. Si el valor es 0, este campo es omitido.
- CHECKSUM: CRC-16 (Véase el siguiente apartado)

Los comandos de sistema que se utilizan, son mostrados en la Figura 4.5.

Version del Firmware (0x00)

Del	A	Ejemplo de Mensaje	Ejemplo de Respuesta
PC	Lector	05 00 00 xx xx	00 ó FF 17 00 00 46 69 72 6D 77 61 72 65 20 56 65 72 20 32 2E 31 37 4D xx xx

Temperatura (0x01)

Del	A	Ejemplo de Mensaje	Ejemplo de Respuesta
PC	Lector	05 00 01 xx xx	00 ó FF 07 00 01 01 1D xx xx

Soft Reset (0x80)

Del	A	Ejemplo de Mensaje	Ejemplo de Respuesta
PC	Lector	05 00 80 xx xx	00 ó FF

Stop (Comando de Parada) (0x00)

Del	A	Ejemplo de Mensaje	Ejemplo de Respuesta
PC	Lector	00	00

Figura 4.5. Comandos del Lector

Por ejemplo; el comando “Versión del Firmware” se emitiría como: 05 00 00 xx xx.

Donde el primer byte denota el número de bytes del paquete, en este caso 05 bytes; “00” en el segundo byte denota un comando de sistema; “00” en el tercer byte denota el comando ‘leer el firmware’. Note que para este comando no hay datos, por lo que se omite este campo. Los dos bytes “xx xx” finales representan el CHECKSUM. En este caso, de acuerdo a la especificación CRC-CCITT¹ resulta “D8 93” Hex.

Como respuesta a este comando el lector envía primero un “00” o un “FF”, denotando que el comando se recibió y entendió correctamente, o que dicho comando se recibió con error. Si sucedió esto último, el lector ya no envía ninguna otra información adicional. Si por el contrario el comando se recibió correctamente, se envía una respuesta. Para este caso, la respuesta sigue la misma estructura señalada anteriormente:

17 00 00 **46 69 72 6D 77 61 72 65 20 56 65 72 20 32 2E 31 37 4D** 12 5A

Donde **46 69 72 6D 77 61 72 65 20 56 65 72 20 32 2E 31 37 4D** son los caracteres ASCII de “Firmware Ver 2.17M”.

Para enviar los bytes que el lector necesita para ejecutar el comando se utiliza un método llamado setEscribir, el cual se encuentra en la clase Lector, que recibe el arreglo de bytes que serán enviados por el puerto de la computadora.

```
//Método de escritura en el puerto utilizado solo por esta clase
private void setEscribir(byte [] mensaje) {
    iIndice = 0; //Inicializamos el contador de bytes leídos

    // Inicializamos el arreglo con valores negativos
    for(int i = 0; i < iTamBuffer; i++)
        bLeidos[i][0] = (byte)-1;

    // Si está habilitado el puerto para escritura, escribimos.
    if(escribir != null)
        escribir.setEscribe(mensaje);
}
```

Otro caso particular es el comando de temperatura, donde se consigue con la siguiente fórmula descrita en la Figura 4.6.

$$(Temp1 * 256 + Temp2) / 10$$

Figura 4.6. Fórmula ocupada para obtener la temperatura del lector.

¹ CCITT son las siglas de Comité Consultivo Internacional Telegráfico y Telefónico (Consultative Committee for International Telegraphy and Telephony).

Donde:

- Temp1: es el 4º byte obtenido del comando enviado por el lector.
- Temp2: es el 5º byte obtenido del comando enviado por el lector.

El resultado es un valor entero que representa la temperatura en grados Celsius.

El comando de Soft Reset reinicia el lector.

El comando de “Stop”, es utilizado para detener el lector después de haber habilitado el comando de lectura de etiquetas EPC clase 1, este comando (“IDs”, véase la Figura 4.7.), provee la habilidad de leer múltiples etiquetas simultáneamente, debido a que al ser invocado, el lector no detiene su búsqueda de etiquetas hasta recibir el comando de “Stop”.

El comando que se utiliza para poder leer las etiquetas de clase 1 es el mostrado en la Figura 4.7. este tiene en su 2º Byte un 16 (Hexadecimal), lo que indica que es un comando usado para leer etiquetas clase 1.

IDs (0x04)

Del	A	Ejemplo de Mensaje	Ejemplo de Respuesta
PC	Lector	05 16 04 xx xx	00 ó FF 11 16 04 11 22 33 44 55 66 77 88 99 AB CD EF xx xx

Figura 4.7. Comandos de las Etiquetas EPC clase 1

Los comandos de los mensajes de envió y de respuesta, que se muestran en las Figuras 4.5. v 4.7., tienen al final de cada uno, unos símbolos “xx xx”, los cuales simbolizan el CHECKSUM (CRC, Control Redundante Cíclico), éste es utilizado para detectar fallas tanto en el envió de los mensajes (computadora - lector) como en los mensajes de respuesta (lector - computadora), esto se explica con detalle en el apartado siguiente.

4.4. Comunicación lector – computadora

La interactividad entre ambos elementos (lector y computadora), se consigue creando el entorno de trabajo necesario para que ambos logren comunicarse, esto se hace de acuerdo a las especificaciones del lector EPC.

Lo primero es configurar la computadora para leer su puerto serie (RS-232), esto se logra haciendo una conexión con los siguientes valores:

Velocidad en Baudios	9600
Bits de Datos	8
Paridad	Ninguna
Bits de Parada	1
Control de Flujo	Ninguno

El código usado para esta configuración se encuentra dentro de la clase Lector en el método setComunicar, se coloca dentro de una estructura try y catch para recibir cualquier error que se pudiera presentar:

```
// Se fijan los parámetros de comunicación del puerto
try {
    puertoSerie.setSerialPortParams( 9600,
    SerialPort.DATABITS_8,
    SerialPort.STOPBITS_1,
    SerialPort.PARITY_NONE );
} catch( UnsupportedOperationException e ) {
    e.printStackTrace();
    return 2; //No se soportan los parámetros de operación
}
```

Corrección de errores

El siguiente paso es determinar el algoritmo Checksum, que se encarga de detectar si existe un error en la información que es enviada o recibida.

El algoritmo de transmisión mostrado en la Tabla 4.6., debe estar formado de acuerdo al manual del lector y a lo que determina el CCITT, al cual pertenece el CRC que utilizamos [Auto-id 2002].

Tabla 4.6. CRC utilizado para la transmisión

Tipo de CRC	Tamaño	Polinomio	Ajustar a	Residuo
CCITT 16	16 bits = 2 bytes	0x1021	0xFFFF	0

La Tabla 4.7. muestra el tipo de CRC que se utiliza en los mensajes de respuesta del lector a la computadora.

Tabla 4.7. CRC utilizado para la recepción.

Tipo de CRC	Tamaño	Polinomio	Ajustar a	Residuo
CCITT 16	16 bits = 2 bytes	0x1021	0xFFFF	0xFFFF

Cada vez que se envía un mensaje se espera una respuesta de parte del lector; éste es representado por un 00 ó un FF (Hexadecimal), donde el 00 representa que el comando fue recibido correctamente y el FF es el caso erróneo.

Estos mensajes son recibidos una vez transferido el comando completo y se deben esperar en un tiempo aproximado de 100 ms. (milisegundos).

También se debe determinar el control de flujo de la información, es decir la forma en que la aplicación envía información al lector.

La Figura 4.8. muestra este control de flujo, en ella se puede apreciar cómo el lector responde a un comando específico (IDs, que sirve para leer las etiquetas) mandando un 00 (Hexadecimal, indica que el comando es recibido correctamente) y comienza a enviar la lectura de los identificadores de las etiquetas que encuentre en su alcance de lectura y esto no se detiene hasta que es recibido el comando de parada (Stop).

Dentro de esta Figura se pueden apreciar algunas abreviaturas las cuales simbolizan lo siguiente:

- CMD:** Comando (IDs).
- ACK:** Respuesta de comando recibido (00, en este caso).
- RESP \geq 1:** Respuesta de acción del comando, una o más respuestas (por parte del lector).
- RESP:** Respuesta de acción del comando, una respuesta (por parte de la/s etiqueta/s).
- STOP:** Comando de Parada.

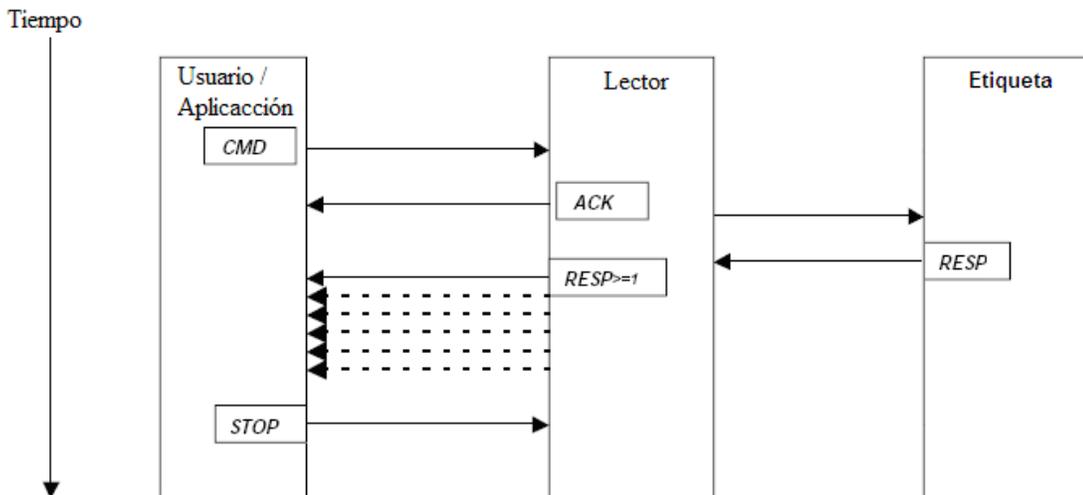


Figura 4.8. Control de Flujo.

4.5. Interacción con la base de datos

La Figura 4.9. es una captura de la pantalla del sistema desarrollado, donde se visualiza la Tabla de Registros de los vehículos donde se observa cómo se almacenan las entradas y salidas.

RegistroId	relVehiculoId	FechaEntrada	HoraEntrada	FechaSalida	HoraSalida
1	1	11/10/2006	20:30	12/10/2006	12:00
41	1	06/11/2006	18:35		
42	1	06/11/2006	18:35		
43	1	06/11/2006	18:35		
44	1	06/11/2006	18:41		
45	1	06/11/2006	18:41		
46	1	06/11/2006	18:52		
47	1	06/11/2006	18:52		
48	1	06/11/2006	18:55		
49	1	06/11/2006	18:55		
50	1	06/11/2006	18:55		
51	1	06/11/2006	18:55		
52	1	06/11/2006	18:55		
53	1	06/11/2006	19:8		
54	1	06/11/2006	19:37		
55	1	06/11/2006	21:34		
56	1	06/11/2006	21:44		

Figura 4.9. Tabla de Registros

La Figura 4.10. es una captura de la pantalla mostrada por el sistema, en la cual se puede apreciar la interfaz vista por el administrador que maneja la aplicación. En esta Figura se puede observar que existe un formulario para poder ingresar nuevos registros (crear), así como las opciones de editar, borrar un elemento, mientras que del lado derecho se muestran los registros ya almacenados en la base de datos.

Propietarioid	Nombre	Departamento
1	Juan Jose Lopez Martinez	Finanzas
2	Sonia Mendez cruz	Recepción
3	Saul Lopez Arellano	Logística

Figura 4.10. Tabla de Propietarios.

La Figura 4.11. es un caso parecido a la Figura 4.10., donde se muestra la interfaz que visualiza el administrador del sistema al interactuar con la Tabla de Etiquetas, al igual que en el caso anterior también se pueden agregar, editar o borrar registros de esta tabla.

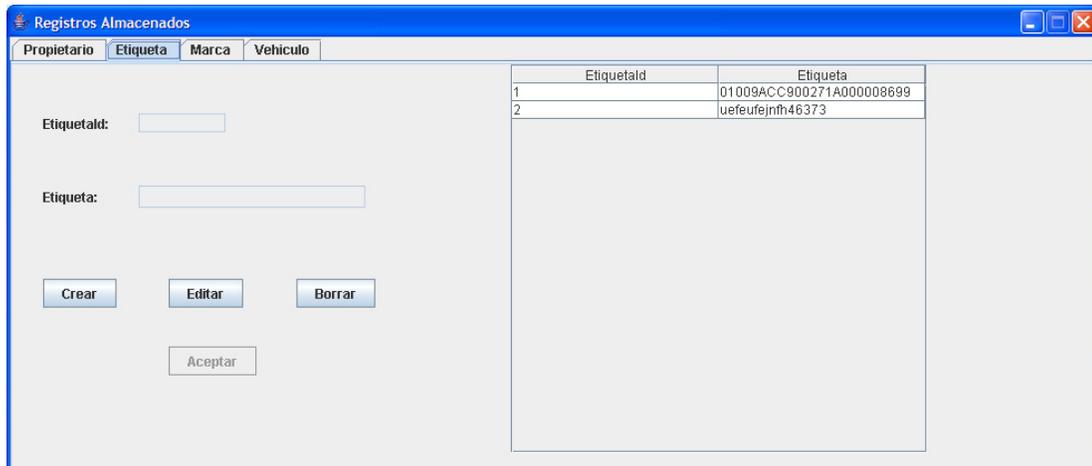


Figura 4.11. Tabla de Etiquetas.

La Figura 4.12. muestra la interfaz que visualiza el administrador del sistema al acceder a la tabla de marcas, esta tabla también puede ser utilizada para poder agregar, editar o borrar registros, esto permite que se puedan mantener actualizados los registros de la bases de datos.

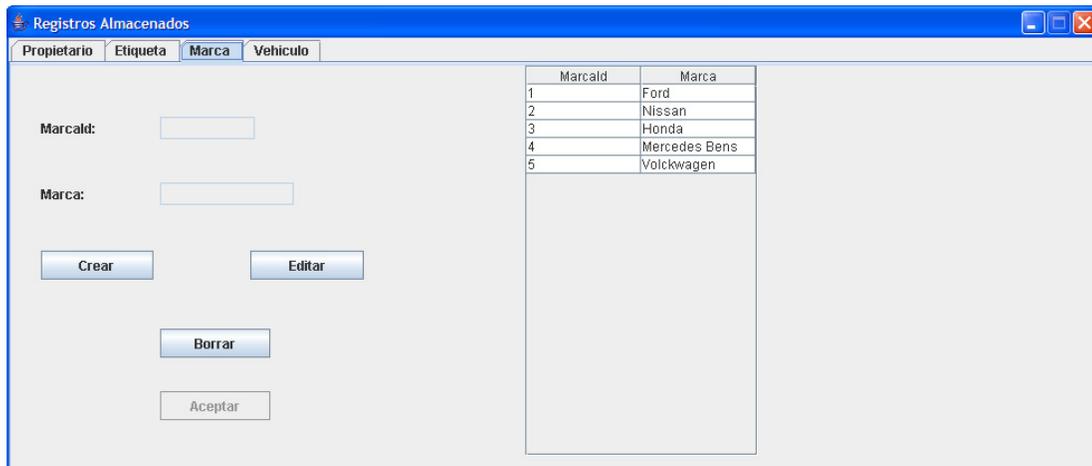


Figura 4.12. Tabla de Marcas (de vehículos).

En la Figura 4.13. se presenta la tabla de vehículos, de la base de datos ya modelada en el apartado 4.1. Esta Figura al igual que las anteriores muestra como el administrador interactúa con el sistema.

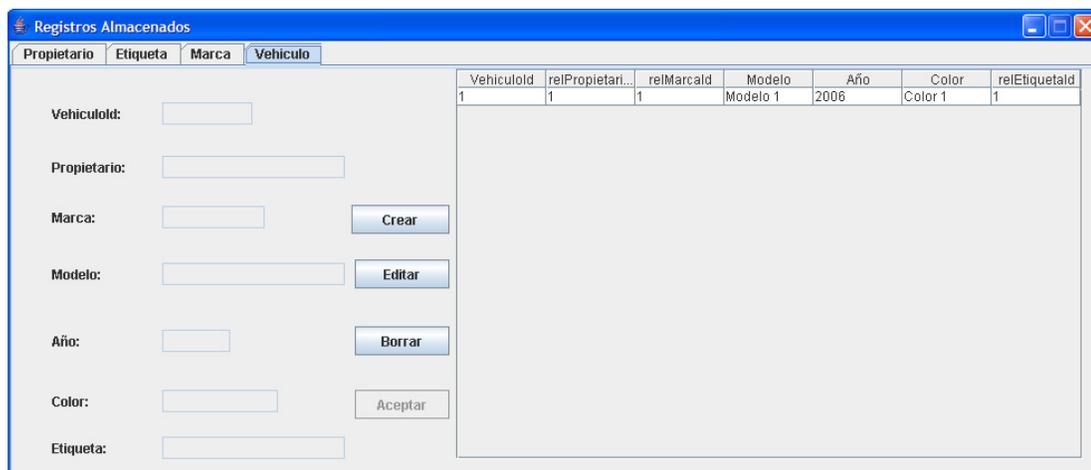


Figura 4.13. Tabla de Vehículos.

La aplicación puede interactuar con la base de datos (realizada en MySQL), gracias al Conector/J que provee de un conjunto de clases que permite la comunicación con ella.

El código encargado de interactuar con la base de datos se encuentra en la clase BDlector, en el método setBDlectorConecta es el encargado de utilizar el Conector/J y de establecer comunicación con la base de datos:

```
try {
    // Creamos una nueva instancia del driver para trabajar con mysql
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    conn = DriverManager.getConnection(sUrl,sLogin,sPassword);

    if (conn != null) //Si se logra conexión con la base datos
        stmt = conn.createStatement();
} catch(SQLException ex) {
    System.out.println(ex);
    return 0;
}
```

La programación del sistema es realizado en este capítulo, pero este código es mostrado en el Anexo A.

Una vez terminada la codificación [Arnold 2001] se procede a ejecutar el sistema, pero para ello se debe tener la seguridad de que el sistema está libre de errores, para ello se necesita hacer una serie de pruebas, mismas que se presentan en el capítulo 5.

En el próximo capítulo:

Se muestran las pruebas realizadas al software, así como lo que se obtuvo de la implementación de éstas, del mismo modo se reportan todos los resultados obtenidos por la implementación y pruebas en el momento de hacer funcionar el sistema terminado.

Capítulo 5

Pruebas y resultados

Este capítulo presenta las pruebas realizadas al sistema, muestra la forma en que los procesos son ejecutados e interpretados por el puerto de la computadora, también se hace una comparación entre lo esperado y lo obtenido.

5.1. Diseño de pruebas

Para obtener resultados confiables, se necesita desarrollar una fase de pruebas, que tiene como objetivo encontrar errores de los siguientes tipos:

- Procesos incorrectos o inexistentes.
- Errores relativos en las clases.
- Errores en estructuras de datos o en la base de datos.
- Errores en el rendimiento.
- Errores de inicialización o terminación.

Estas pruebas, se deben realizar en las funciones principales del sistema, para con ello verificar el correcto funcionamiento de cada una de las acciones que ejecuta el software creado.

Las funciones que se verifican son:

- Consultar la versión del software.
- Consultar los datos del creador y nombre de la aplicación
- Consultar la versión del lector (Firmware).
- Consultar la temperatura del lector.
- Realizar un Soft Reset
- Lectura de una etiqueta.
 - No almacenada en la base de datos.
 - Almacenada en la base de datos.

Para las dos primeras funciones se necesita seleccionar la pestaña de Información en el sistema, véase la Figura 5.1.

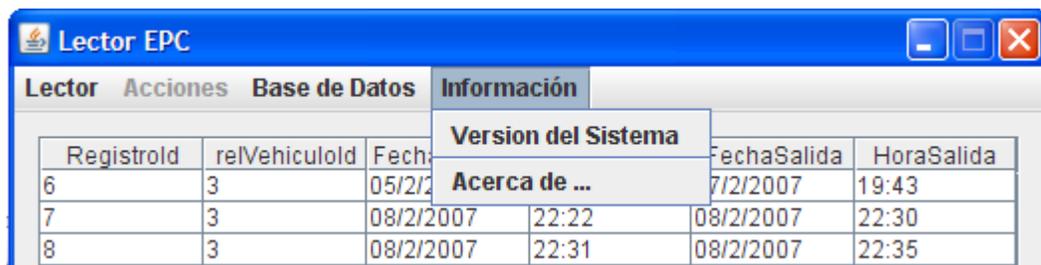


Figura 5.1. Funciones en la pestaña de Información.

Las cuatro funciones que restan son seleccionadas de la pestaña Acciones, véase la Figura 5.2.

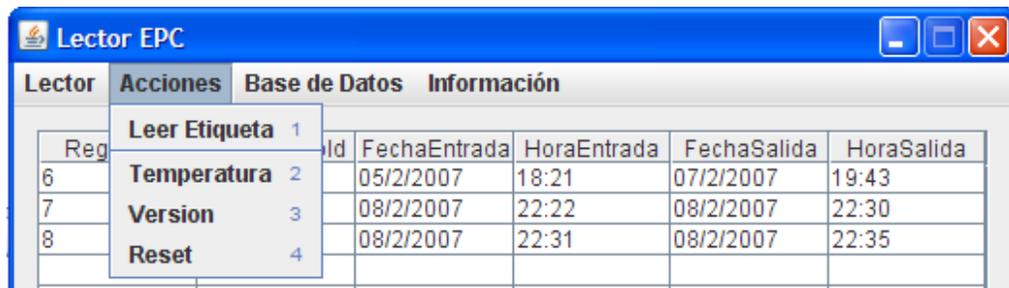


Figura 5.2. Funciones en la caja de Acciones

Del mismo modo se realizan pruebas en las operaciones propias de la base de datos:

- Crear un registro en la base de datos.
- Editar un registro en la base de datos.
- Borrar un registro en la base de datos.

Para acceder el contenido en la base de datos, se necesita seleccionar de la caja Base de Datos, la opción de Administración tal y como se presenta en la Figura 5.3.



Figura 5.3. Funciones en la caja de Base de Datos

Después de la selección, se mostrará una nueva ventana (Figura 5.4.), la cual valida si se trata del administrador del sistema, en caso de serlo se presenta la ventana con las opciones de la base de datos (Figura 5.5.), en caso contrario se regresa a la ventana de Acceso (Figura 5.3).



Figura 5.4. Validación de acceso a Base de Datos.

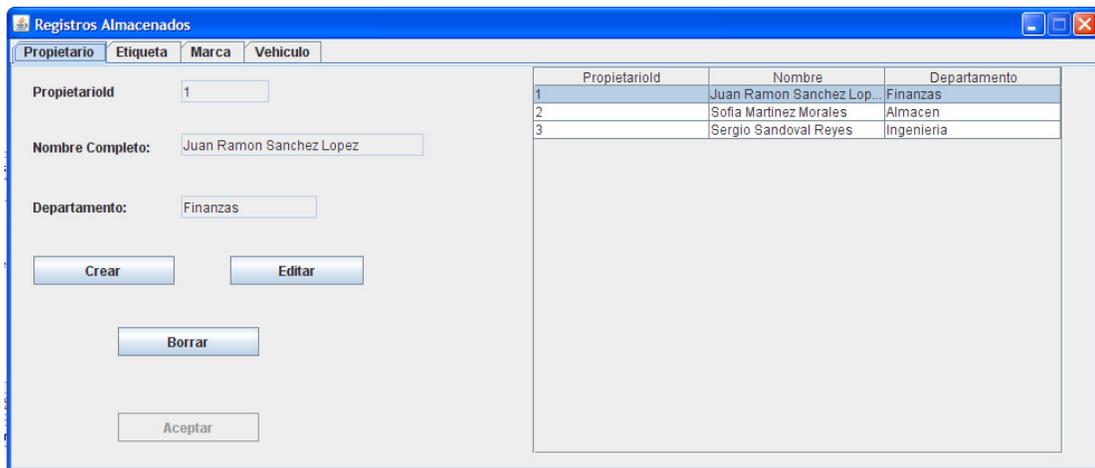


Figura 5.5. Administración de la Base de Datos

A modo de corroborar los datos que son enviados al lector y provenientes de este mismo, se necesita la ayuda de una herramienta; *Portmon* que se puede localizar en la pagina del autor [Sysinternals 2007] es de libre distribución e instalación y sirve para monitorear puertos de la computadora (para este sistema se ocupó el puerto serie COM 6). Véase la Figura 5.6.

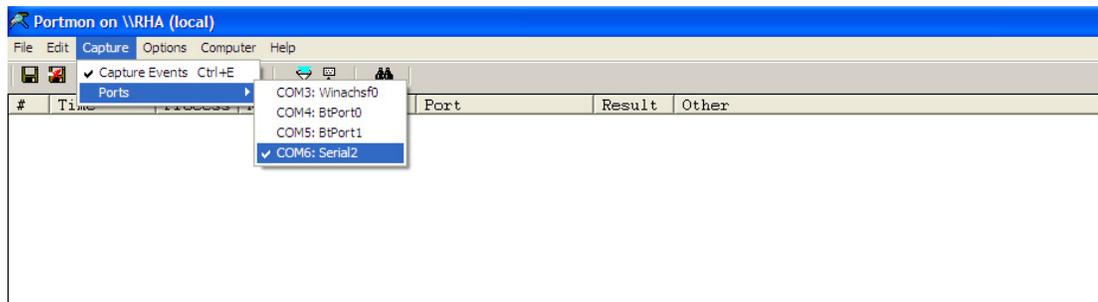


Figura 5.6. Conexión del Portmon con el puerto que utiliza el Lector.

A continuación se describe de manera general el funcionamiento de cada uno de los procesos que se realizan en la aplicación, para mayor descripción de su uso se debe consultar el manual de operaciones (véase el Anexo B).

5.1.1. Consultar la versión del software.

Este es un proceso propio de la misma aplicación, permite verificar la versión del software creado.

Este proceso ejecuta un llamado a la clase `javax.swing.JOptionPane` (Clase de Java), a la cual se le pasan los parámetros necesarios para que muestre el mensaje con la información solicitada.

La Figura 5.7. muestra el resultado de la ejecución de este proceso.

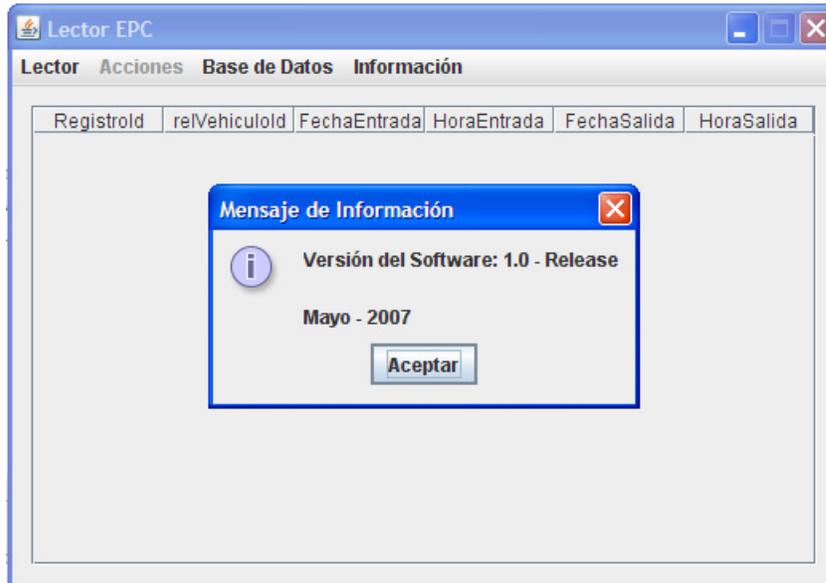


Figura 5.7. Resultado del comando *Versión del Sistema*.

La información obtenida es propiamente: el número de versión de la aplicación y la fecha en que fue creado el sistema.

5.1.2. Consultar los datos del creador y nombre de la aplicación.

Este proceso es similar al descrito en el apartado 5.1.1., se muestra la información del desarrollador del sistema, así como el nombre del proyecto.

También ejecuta un llamado a la clase `javax.swing.JOptionPane`, a la cual se le transfieren los parámetros necesarios para que se muestre el mensaje deseado.

El resultado de ejecutar este proceso es mostrado en la Figura 5.8.

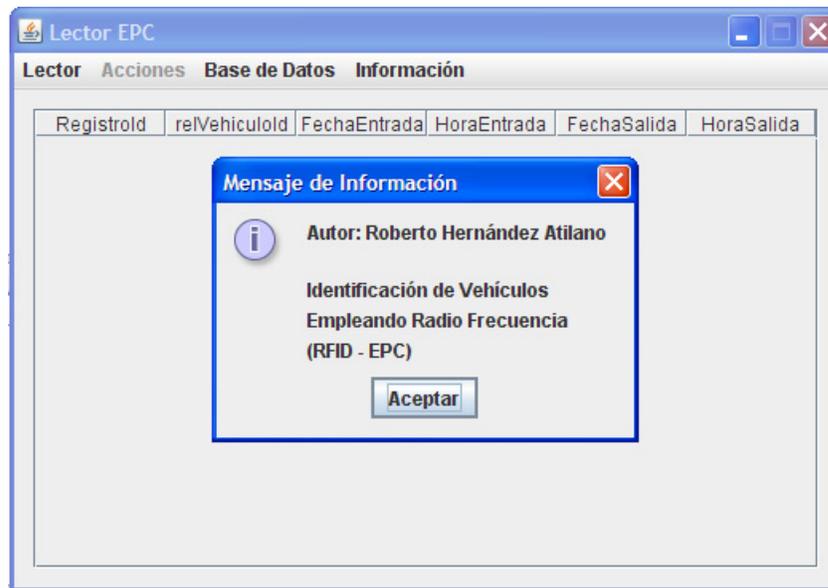


Figura 5.8. Resultado del comando *Acerca de*.

La información obtenida del sistema es propiamente: el nombre del desarrollador de la aplicación y el nombre del proyecto.

5.1.3. Consultar la versión del lector (Firmware).

Este proceso se encarga de comunicarse con el lector, le envía el comando necesario, espera respuesta e interpreta la información recibida.

El proceso ejecuta un llamado a la clase Lector, la cual se encarga de establecer comunicación con el puerto de la computadora, así como enviar y recibir los comandos entre el sistema y el lector.

La clase Lector, se encarga de verificar la información recibida del lector, esta clase muestra la versión solicitada. Véase la Figura 5.9.

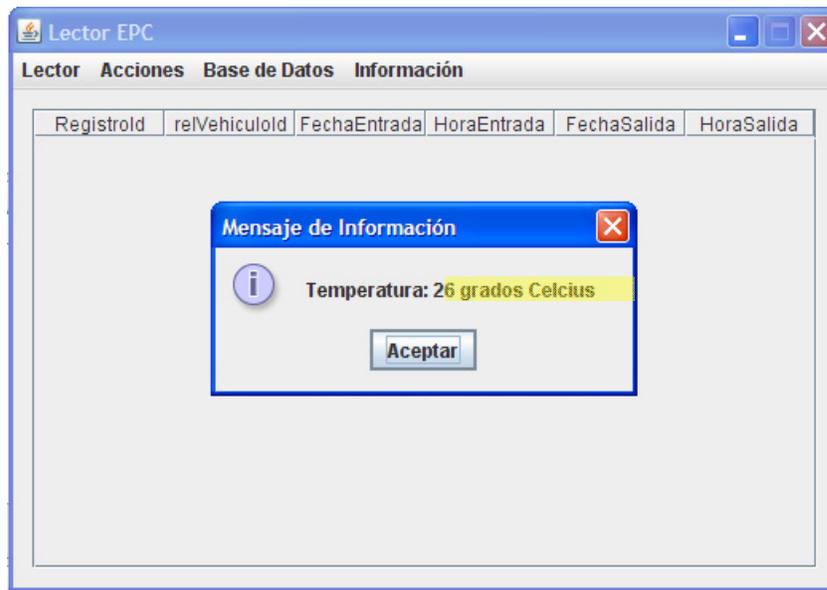


Figura 5.9. Resultado del comando Versión del Lector (Firmware).

La Figura muestra el número de versión del lector (de color amarillo), se puede corroborar que esta información es recibida desde el puerto de la computadora con ayuda del programa *Portmon*, véase la Figura 5.10.

Secuencia	Tiempo	Proceso	Puerto	Acción	Tamaño	Bytes de Resultado	Cod. ASCII
0	0.00029557	IRP_MJ_WRITE	Serial6	SUCCESS	Length 5:	05 00 00 D8 93	
5	0.00000643	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	00	
8	0.00000419	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	17	
20	0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	00	
28	0.00000531	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	00	
36	0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	46	F
44	0.00000615	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	69	i
52	0.00000531	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	72	r
60	0.00000615	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	6D	m
68	0.00000503	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	77	w
76	0.00000587	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	61	a
84	0.00000615	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	72	r
92	0.00000587	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	65	e
100	0.00000559	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	20	
108	0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	56	V
116	0.00000615	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	65	e
124	0.0000067	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	72	r
132	0.00000615	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	20	
140	0.0000067	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	32	2
148	0.00000503	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	2E	.
156	0.00000587	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	31	1
164	0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	37	7
172	0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	4D	M
180	0.00000559	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	12	
188	0.00000559	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	5A	

Figura 5.10. Captura de Portmon del proceso de Versión del Lector

La Figura anterior muestra la información recibida del lector, de color amarillo está el código ASCII¹ que es el equivalente numérico de los Bytes de Resultado.

5.1.4. Consultar la temperatura del lector

La temperatura del lector es presentada en la Figura 5.11.

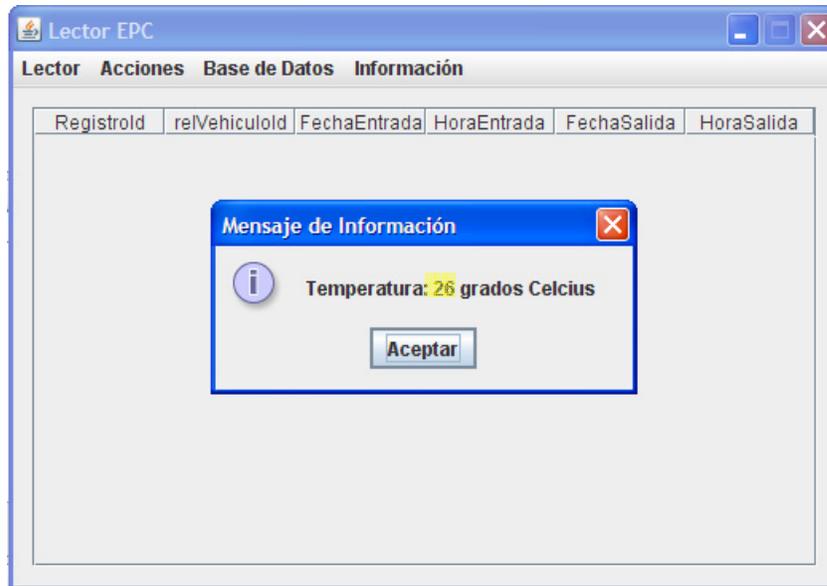


Figura 5.11. Resultado del comando Temperatura.

A continuación se presenta la información capturada por el programa *Portmon*, donde se muestran los datos obtenidos del lector. Véase la Figura 5.12.

Secuencia	Tiempo	Proceso	Puerto	Accion	Tamaño	Bytes de Resultado
156	0.00114791	IRP_MJ_WRITE	Serial2	SUCCESS	Length 5:	05 00 01 C8 B2
161	0.00000615	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	00
169	0.00000531	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	07
177	0.00000559	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	00
185	0.00000559	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	01
193	0.00000615	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	01
201	0.00000615	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	13
209	0.00000587	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	50
217	0.00000559	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	8B

Figura 5.12. Captura de Portmon del proceso de Temperatura.

¹ El código ASCII (American Standard Code for Information Interchange, Código Estadounidense Estándar para el Intercambio de Información), es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y en otras lenguas occidentales.

La Figura 5.11. muestra el resultado de la ejecución del comando de temperatura en la aplicación, de color amarillo se encuentra el valor numérico (26), el cual representa la temperatura en grados Celsius.

Mientras que en la Figura 5.12. es presentada la información obtenida de la aplicación *Portmon*, después de la ejecución de este mismo comando.

De color amarillo son mostrados los dos bytes necesarios para obtener la temperatura (bytes 4 y 5).

Utilizando la fórmula descrita en el capítulo 4 (Figura 4.6.), se obtiene:

Temp1 = 01 hexadecimal.

Temp2 = 13 hexadecimal.

$$(01 \times 256 + 13) / 10 = 269 / 10 = 26.9$$

Con esta información es obtenida la temperatura del lector, misma que es presentada por la aplicación.

5.1.5. Lectura de una etiqueta

Para la lectura de una etiqueta se necesita la utilización de varias de las clases del sistema, debido a que requiere de la comunicación con el puerto de la computadora, de la verificación de los datos, de algunas consultas en la base de datos y los mensajes de información.

Para corroborar el correcto funcionamiento de las instrucciones y comandos se utiliza el programa *Portmon*, el cual muestra la captura de los mensajes entre puerto y lector del proceso de lectura de una etiqueta.

Esto se aprecia en la Figura 5.13. las líneas de color amarillo, muestran el Id de la etiqueta leída, mismo que se ocupa para consultar la base datos en búsqueda de la información del vehículo al que le fue asignada la etiqueta en cuestión.

Secuencia	Tiempo	Proceso	Puerto	Accion	Tamaño	Bytes de resultado
0	0.00088084	IRP_MJ_WRITE	Serial2	SUCCESS	Length 5:	05 16 04 31 C2
5	0.00000643	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	00
13	0.00000643	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	11
21	0.00000447	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	16
29	0.00000475	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	04
37	0.00000419	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	00
45	0.00000419	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	00
53	0.00000419	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	00
61	0.00000419	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	00
69	0.00000447	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	00
77	0.00000419	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	00
85	0.00000419	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	61
93	0.00000419	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	01
101	0.00000447	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	19
109	0.00000447	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	23
117	0.00000475	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	66
125	0.00000475	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	44
133	0.00000447	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	48
141	0.00000447	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	EB
145	0.00089062	IRP_MJ_WRITE	Serial2	SUCCESS	Length 1:	00
150	0.00000559	IRP_MJ_READ	Serial2	SUCCESS	Length 1:	00

Figura 5.13. Captura de Portmon del proceso de Leer Etiqueta.

Una vez localizada la etiqueta en la base de datos, se procede a recuperar toda la información referente al propietario del vehículo, ésta se muestra en un mensaje al usuario que opera el sistema. Véase la Figura 5.14.

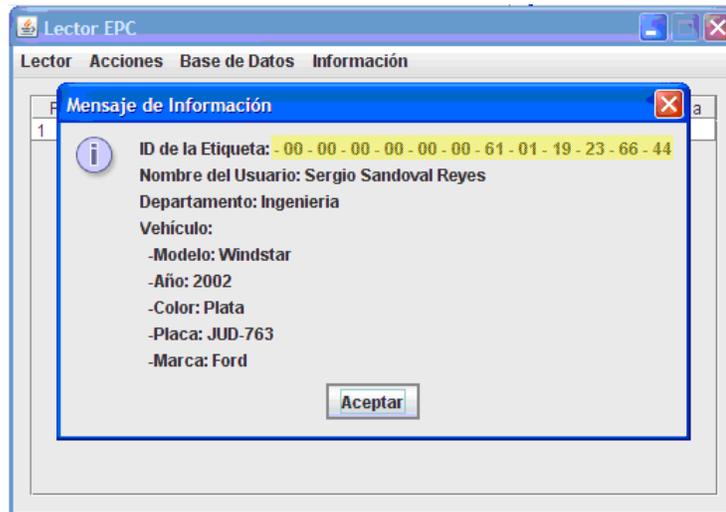


Figura 5.14. Resultado del comando de Leer Etiqueta.

En las Figuras 5.13. y 5.14. se muestran los mensajes con el Id de la etiqueta, en ambos se puede apreciar que este identificador de la etiqueta es el mismo. De este

modo se puede probar que el software está funcionando adecuadamente, ya que la respuesta obtenida (en la Figura 5.14), es la esperada, puesto que muestra todos los datos recolectados en la base de datos y que están relacionados con la etiqueta leída.

5.1.6. Realizar un Soft Reset

Este comando reinicia el lector, al restablecerse manda un mensaje avisando que está listo, esta información es capturada por el sistema y es el que se muestra. Véase la Figura 5.15.

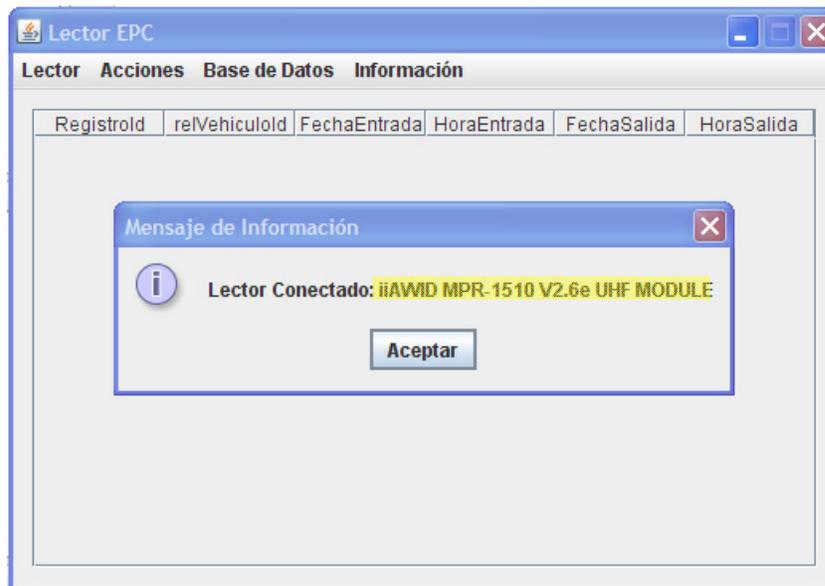


Figura 5.15. Resultado del comando Soft Reset.

Para corroborar que la información que se manda del lector a la computadora es la correcta, se utiliza el programa *Portmon*, en la Figura 5.16. se muestran los bytes de respuesta y de color amarillo se presenta su equivalente en código ASCII, los cuales forman el mensaje que es enviado por la aplicación.

Tiempo	Proceso	Puerto	Acción	Tamaño	Bytes de Resultado	Cod. ASCII
0.00070149	IRP_MJ_WRITE	Serial6	SUCCESS	Length 5:	05 00 80 49 1B	
0.00000643	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	00	
0.00000643	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	69	i
0.00000615	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	69	i
0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	41	A
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	57	W
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	49	I
0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	44	D
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	20	
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	4D	M
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	50	P
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	52	R
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	2D	-
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	31	1
0.00000587	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	35	5
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	31	1
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	30	0
0.00000503	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	20	
0.00000503	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	56	V
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	32	2
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	2E	.
0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	36	6
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	65	e
0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	20	
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	55	U
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	48	H
0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	46	F
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	20	
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	4D	M
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	4F	O
0.00000503	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	44	D
0.00000447	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	55	U
0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	4C	L
0.00000475	IRP_MJ_READ	Serial6	SUCCESS	Length 1:	45	E

Figura 5.16. Captura de Portmon del proceso de Soft Reset.

5.1.7. Crear un registro

Crear un registro en la base de datos, es básicamente la ejecución de una instrucción de SQL, esta sintaxis es llamada desde la forma de “Administración” y donde los parámetros son tomados de los JTextFields (Clases predefinidas, para una caja de texto en Java).

Para corroborar que las clases y los métodos funcionen correctamente, se verifica que se muestre el registro que se quiere insertar después de darle la instrucción Crear e insertar los datos.

Para ejemplificar el funcionamiento de este proceso en cualquiera de las tablas de la base de datos, se presenta la Figura 5.17. la cual muestra ventana de Registros Almacenados, antes de la creación del registro.

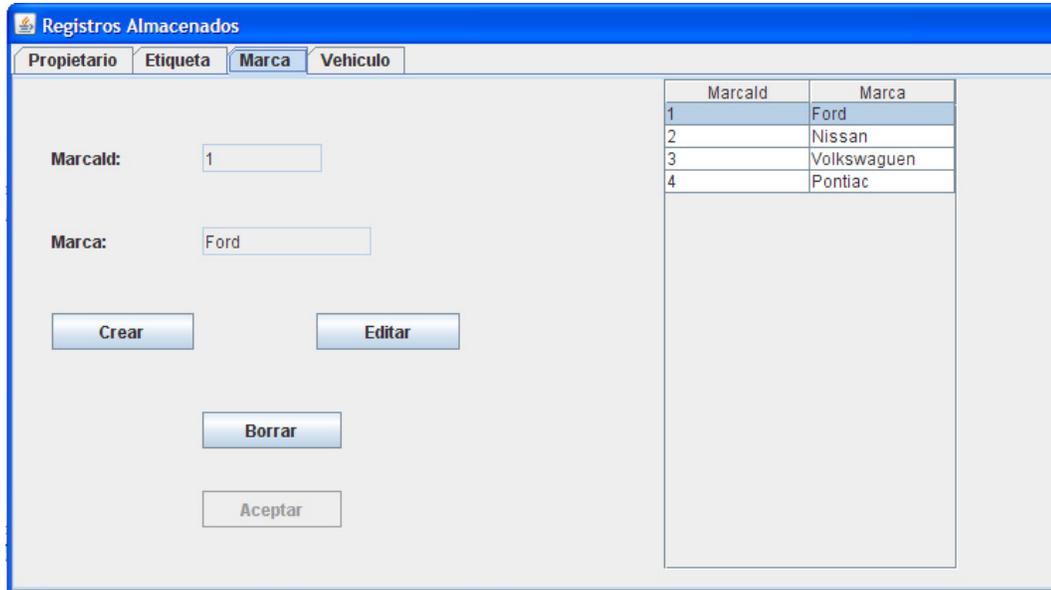


Figura 5.17. Registros almacenados en la tabla Marca.

La Figura 5.18. muestra el resultado de crear un registro, de color azul se presenta el nuevo registro.

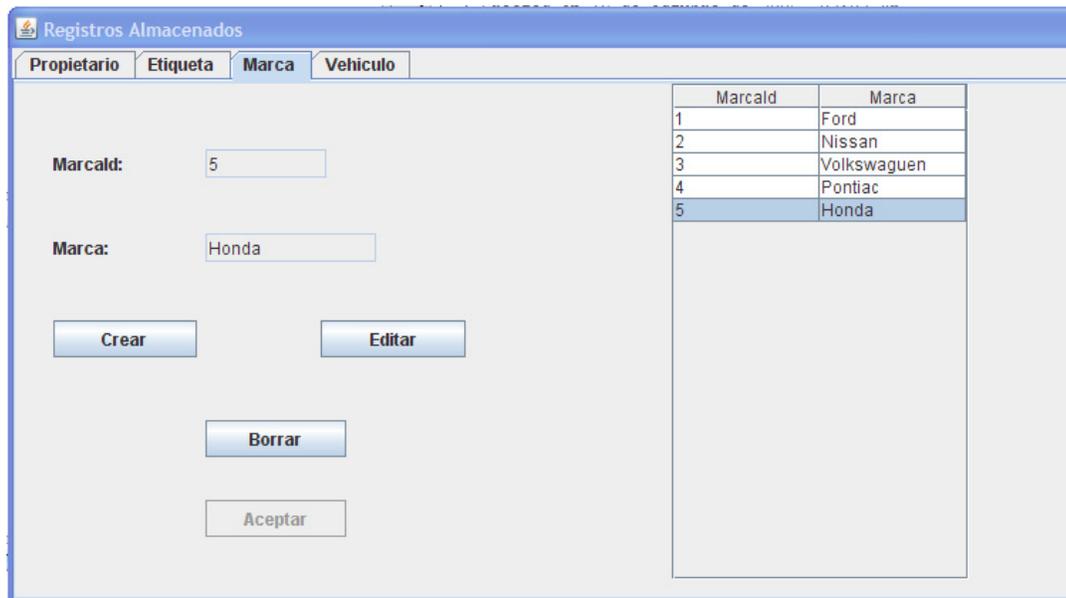


Figura 5.18. Creación de un registro en la tabla Marca.

5.1.8. Editar un registro

El proceso de edición de un registro es similar al descrito para crear una etiqueta, solo se diferencia en que al presionar el botón de editar, las cajas de texto se convierten en editables (que se les puede cambiar su valor actual) y que la modificación de un registro en la tabla vehículos tiene una opción más, en la cual se muestra una forma extra donde son exhibidos los registros almacenados de otras tablas, esto es para facilitarle la inserción de los datos al administrador del sistema. Véase la Figura 5.19.

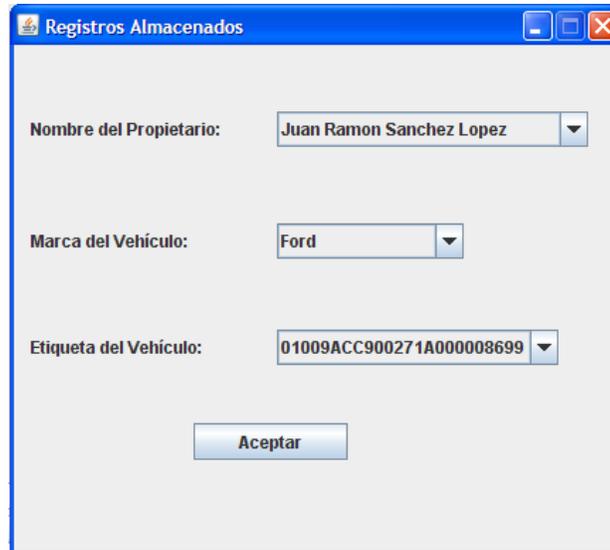


Figura 5.19. Ventana de opciones en la edición de un registro de la tabla vehículos.

Para ejemplificar la edición de un registro se presenta una captura de pantalla de la ventana de Registros Almacenados antes de ser modificado. Véase la Figura 5.20.

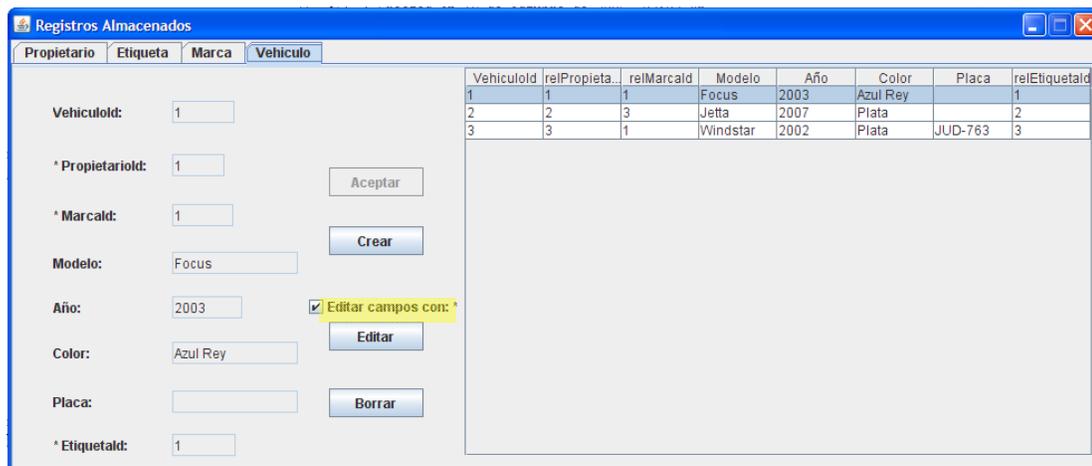


Figura 5.20. Registros almacenados en la tabla Vehículos.

La opción de “Editar campos con *” (color amarillo) de la Figura anterior, corresponde solo a la tabla de vehículos, en las otras tablas no aparece esta opción (Tablas: propietarios, marcas y etiquetas).

El resultado de la edición es presentado en la Figura 5.21. se puede observar que al inicio (Figura 5.20), en el primer registro, el Año es 2003 y el Color es de Azul Rey, mientras que en esta Figura, el Año es 2005 y el Color es verde.

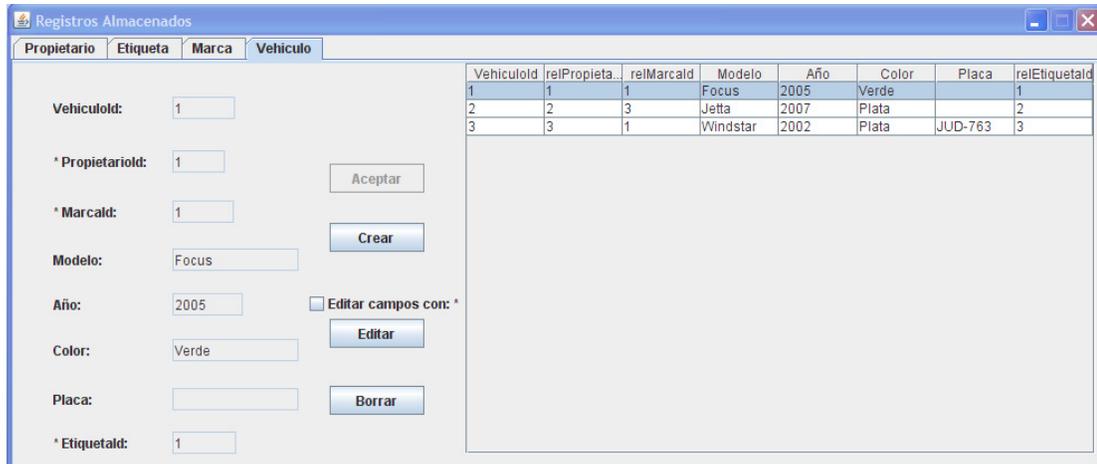


Figura 5.21. Edición de un registro en la tabla Vehículos.

5.1.9. Borrar un registro

Este proceso permite la modificación de la base de datos del sistema, ya que solo es necesario presionar el botón de Borrar para la ejecución de una instrucción SQL en la base de datos, para que se elimine el registro deseado.

Para corroborar que el sistema realice bien su función, se presenta la tabla propietarios antes de ser borrado un registro (Véase la Figura 5.22.). Este proceso es similar para todas las tablas de la base de datos (vehículos, propietarios, marcas y etiquetas).

El registro de color azul es el que será borrado.

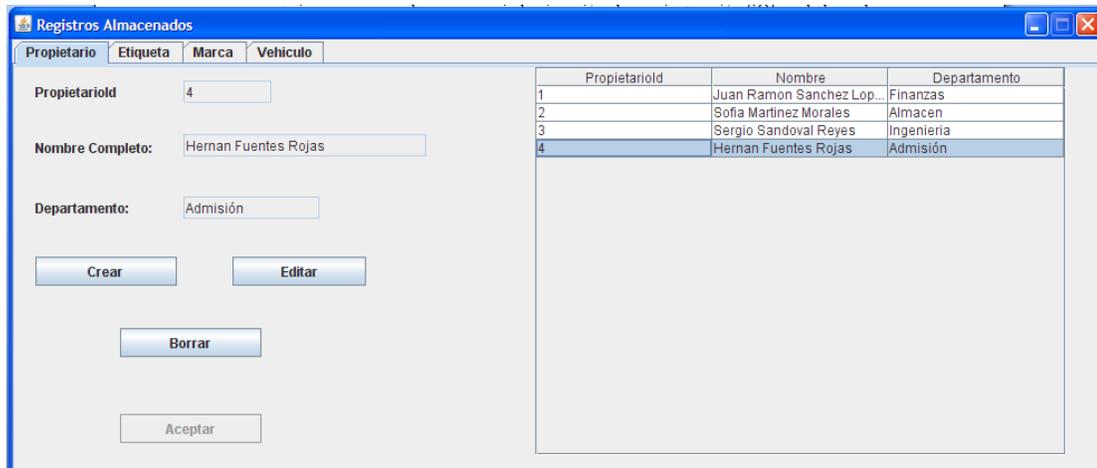


Figura 5.22. Registros almacenados en la tabla Propietarios.

La Figura 5.23. muestra la tabla Propietarios después de haberse borrado un registro.

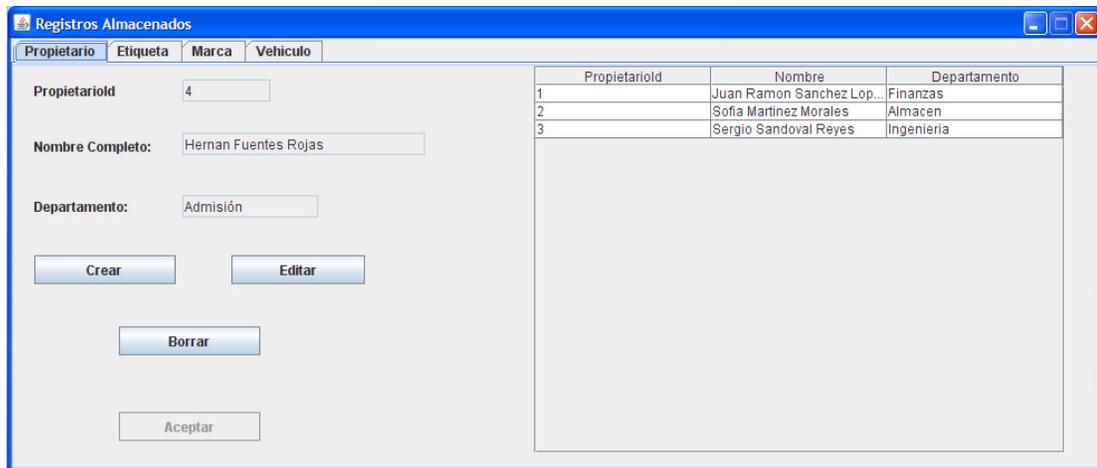


Figura 5.23. Borrado de un registro en la tabla Propietarios.

5.2. Resultados obtenidos

De acuerdo a los resultados obtenidos, se puede concluir que la etapa de pruebas se termino con éxito, al no encontrar errores en los procesos que se ejecutan en la aplicación.

Los comandos fueron verificados con la ayuda del programa *Portmon*, con ello se observa que tanto las respuestas enviadas del lector, como lo que se presenta por la aplicación, es la misma información.

Capítulo 6

Conclusiones

El capítulo presenta las conclusiones del trabajo desarrollado, se muestran los objetivos logrados, también se menciona una pequeña descripción de mejoras a futuro que se pueden realizar a partir de este proyecto.

6.1. Objetivos logrados

El software resultante de esta investigación funciona de la manera esperada, mostrando e interactuando con los medios que se requieren conectar entre sí.

Este trabajo de investigación cubre los objetivos descritos anteriormente (capítulo 1) de la siguiente manera:

De manera general:

Se desarrolló un software que permite la detección de etiquetas EPC colocadas en vehículos, utiliza la tecnología de identificación por radio frecuencia (RFID-EPC) y asocia a una base de datos para registrar entradas y salidas.

De manera específica:

Se implementó un software para leer los datos de una etiqueta EPC localizada en un vehículo.

Se incorpora una base de datos para guardar información sobre las etiquetas leídas.

Se desarrolló una relación entre la base de datos y la información obtenida del lector.

Se realiza una comparación con los datos leídos del lector con los de la base de datos y se muestran.

Se utiliza un estándar de calidad (EPC Global).

Se obtuvo seguridad en el sistema, permitiendo que solo el administrador tenga permiso de modificar el contenido de la base de datos.

En el momento de la detección de una etiqueta, se almacena un registro del Id de la etiqueta del vehículo y de todos los datos relacionados con ese automóvil, este proceso se realiza de forma automática y aunque éste lo puede ejecutar un usuario sin ser administrador, los datos almacenados no pueden ser alterados sin antes validar que se trata de una persona con permiso de edición de la base de datos (administrador del sistema).

La utilización de aplicaciones de libre distribución como es el caso de MySQL, NetBeans y Portmon, permitieron la correcta implementación del sistema, además de ahorrar costo en el diseño y construcción del mismo, ya que no se pagan licencias por el uso de las herramientas empleadas.

Aspectos innovadores:

- Se utiliza la tecnología RFID, para resolver el problema de la identificación de vehículos.
- Se utiliza con éxito el estándar de calidad EPC, permitiendo que el sistema sea compatible con equipos que estén dentro este estándar.
- Se hace uso de etiquetas EPC de clase 1 pasivas, las cuales son capaces de ser utilizadas en vehículos.

Alcances y aplicaciones

- El sistema es usado con eficiencia para controlar el acceso a lugares que requieren de control de vehículos y que cuentan con personal de vigilancia para la seguridad de dicho lugar.
- El sistema es capaz de ser expandible, permitiendo la incorporación de nuevas funciones que estén dentro del estándar EPC, como es el caso de destruir la etiqueta o agrega un password en el Id de la etiqueta.

6.2. Mejoras a futuro

Las mejoras a futuro están propuestas teniendo como base lo aquí desarrollado. El presente trabajo está diseñado de tal forma que permite la actualización del sistema conforme sea necesario. A continuación se describen brevemente algunas de estas mejoras.

6.2.1. Incorporación de nuevas etiquetas de otras clases

El modelado del sistema cuenta con un diseño que permite la incorporación de nuevas etiquetas como por ejemplo el caso de etiquetas EPC clase 1 en su versión 2, del mismo modo se pueden utilizar etiquetas de clase 0. (Clases de acuerdo a las especificaciones de EPC Global).

Cabe recordar que siempre que se quiera incorporar una nueva etiqueta en cualquiera de sus versiones, debe considerarse que el lector que es utilizado debe ser capaz de leer esta clase de etiquetas.

La opción de agregar etiquetas de otras versiones distintas a las que se usaron en este trabajo, permite al usuario, tener versatilidad en el uso del sistema.

Para esto solo bastaría con agregar una nueva clase donde se definan los métodos que se utilizarán, los cuales proporcionan los bytes que serán enviados al lector y recibidos del mismo, (enviados y recibidos en la clase “Lector”) y a su vez,

incorporar en el menú de inicio (clase “Acceso”), las opciones que permitan llamar estos métodos.

6.2.2. Acceso vía Web a la base de datos del sistema.

Una de las razones por las cuales se decidió el uso de una base de datos de libre distribución, es que el usuario del sistema puede no solo modificarla a su conveniencia, sino que además puede utilizarla en otras aplicaciones diferentes de las que fueron destinadas inicialmente.

Cabe mencionar que algunas compañías que proporcionan bases de datos, limitan la utilización de estas, incluso en ocasiones hacen mención que si se usa en otros sistemas diferentes para el que está destinado inicialmente, se pierde la garantía que se ofrece en el momento de contratación.

Se propone un sistema vía Web capaz de mostrar y modificar (si es necesario), la base de datos del sistema, para con ello permitir a los propietarios de los vehículos conocer la información de entrada y salida, registrada por el sistema al momento del pase del vehículo (que cuenta con una etiqueta) frente al lector EPC.

Además el acceso vía web, permite también al administrador del sistema consultar su información a distancia (gracias al internet).

Glosario

El presente glosario, no pretende ser un diccionario de la investigación científica, simplemente reúne las palabras más usadas en el presente documento.

A

Análisis de Sistemas:

Se denomina análisis de sistemas a una de las etapas de construcción de un sistema informático, que consiste en relevar la información actual y proponer los rasgos generales de la solución futura.

API:

Es la abreviatura de Application Programming Interface (Interfaz Programable de la Aplicación), no es más que una serie de servicios o funciones que el sistema operativo ofrece al programador, como por ejemplo, imprimir un caracter en pantalla, leer el teclado, escribir en un fichero de disco, etc.

Arista:

Una arista puede ser:

- En Teoría de grafos, las aristas, junto con los vértices, forman los elementos principales con los que trabaja esta disciplina, siendo consideradas las aristas las uniones entre nodos o vértices.
- En Geometría, las aristas son las líneas o segmentos en los que se unen dos caras.

ASCII:

El código ASCII utiliza 7 bits para representar los caracteres, aunque inicialmente empleaba un bit adicional (bit de paridad) que se usaba para detectar errores en la transmisión.

Casi todos los sistemas informáticos actuales utilizan el código ASCII o una extensión compatible para representar textos y para el control de dispositivos que manejan texto.

AUTO ID CENTER:

Equipo de investigación del MIT (Massachusetts Institute of Technology) dedicado al estudio de RFID.

BBase de Datos:

Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

CCajas de texto:

Estas permiten introducir información de parte del usuario, son parte también de los componentes más utilizados existen dos tipos de cajas de texto las de una sola línea y las de varias líneas, las primeras son conocidas como: JTextField y las segundas como JTextArea.

Caso de uso:

Es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios. Ese alguien o algo es el actor. Tienen una representación gráfica de óvalos. Como características principales, los casos de uso:

- Están expresados desde el punto de vista del actor.
- Se documentan con texto informal describiendo la interacción.
- Están acotados al uso de una funcionalidad claramente definida del sistema.

Situaciones que pueden darse:

- Un actor se comunica con un caso de uso.
- Un caso de uso extiende otro caso de uso.
- Un caso de uso usa otro caso de uso.

CCITT:

Acrónimo de Comité Consultatif Internationale de Télégraphie et Téléphonie (Comité de Información Internacional de Telecomunicaciones), organización con base en Ginebra, Suiza, establecida como parte de la ITU (United Nations International

Telecommunications Union, Asociación Internacional de Telecomunicaciones de las Naciones Unidas). La CCITT recomienda el uso de los tipos de comunicaciones oficiales reconocidos en todo el mundo. La publicación de estas recomendaciones tiene lugar cada cuatro años. Cada actualización se distingue por el color de la portada. Los protocolos establecidos por la CCITT se aplican a los módems, las redes y las transmisiones de fax.

CHECKSUM:

Una suma de verificación o checksum es una forma de control de redundancia, una medida muy simple para proteger la integridad de datos, verificando que no hayan sido corrompidos. Es empleado para comunicaciones (internet, comunicación de dispositivos, etc.) tanto como para datos almacenados (archivos comprimidos, discos portátiles, etc.).

Clase:

Una clase es un contenedor de uno o más datos (variables o propiedad miembro) junto a las operaciones de manipulación de dichos datos (funciones/métodos). Las clases pueden definirse como estructuras (struct), uniones (unión) o clases (class) pudiendo existir diferencias entre cada una de las definiciones según el lenguaje de programación que se utilice.

CRC:

Los códigos cíclicos también se llaman CRC (Códigos de Redundancia Cíclica) o códigos polinómicos. Su uso está muy extendido porque pueden implementarse en hardware con mucha facilidad y son muy potentes.

Estos códigos se basan en el uso de un polinomio generador $G(X)$ de grado r , y en el principio de que n bits de datos binarios se pueden considerar como los coeficientes de un polinomio de orden $n-1$. A estos bits de datos se le añaden r bits de redundancia de forma que el polinomio resultante sea divisible por el polinomio generador. El receptor verificará si el polinomio recibido es divisible por $G(X)$. Si no lo es, habrá un error en la transmisión.

D

Diagrama de Clases:

El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El

modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.

Diagrama de Colaboración:

Un diagrama de colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizadas entorno a los objetos y los enlaces entre ellos.

Diagrama de Secuencias:

Los diagramas de secuencia proporcionan una forma de ver el escenario en un orden temporal - qué pasa primero, qué pasa después -. Los usuarios entienden fácilmente este tipo de diagramas, por lo que resultan útiles en las primeras fases de análisis. Por contra los diagramas de colaboración proporcionan la representación principal de un escenario, ya que las colaboraciones se organizan entorno a los enlaces de unos objetos con otros. Este tipo de diagramas se utilizan más frecuentemente en la fase de diseño.

Diseño de Sistemas:

En la construcción se aplican ciertas técnicas y principios con el propósito de definir un sistema, con suficientes detalles como para permitir su interpretación y realización física.

La etapa del Diseño del Sistema encierra cuatro etapas:

- El diseño de los datos: Transforma el modelo de dominio de la información, creado durante el análisis.
- El Diseño Arquitectónico: Define la relación entre cada uno de los elementos estructurales del programa.
- El Diseño de la Interfaz: Describe cómo se comunica el Software consigo mismo, con los sistemas que operan junto con él y con los operadores y usuarios que lo emplean.
- El Diseño de procedimientos: Transforma elementos estructurales de la arquitectura del programa.

La importancia del Diseño del Software se puede definir en una sola palabra, *calidad*.

E*EEPROM:*

Memoria más usada en los sistemas con acoplamiento inductivo. Tiene unos ciclos de escritura limitados y un consumo alto de energía. Sus siglas son en inglés: Electrically Erasable Programmable Read Only Memory (Memoria de solo lectura programable y borrrable eléctricamente).

EPC:

El Código Electrónico de Producto (EPC, Electronic Product Code), es la siguiente generación de identificación de productos. Al igual que el Código barras, es un número único que identifica a un artículo específico en la cadena de suministro y mantiene una estructura estándar. El EPC se almacena en una etiqueta de identificación por radio frecuencia (RFID), que combina un chip de silicio y una antena.

EPC Global:

Esta nueva norma global, abierta, combina la tecnología de Identificación por Radio Frecuencia (RFID), la infraestructura de las redes de comunicaciones existentes, y el Código Electrónico de Producto (un número para identificar un artículo de manera única) para crear información eficiente en costos, en tiempo real y exacta sobre la ubicación de los artículos, su historial y el número de artículos en la cadena de suministro. Se basa en la investigación conducida a través del Auto-ID Center con el apoyo de más de 100 compañías líder.

Etiqueta:

Las etiquetas RFID pueden ser activas, semi-pasivas (o semi-activas) o pasivas.

Las etiquetas RFID pasivas no tienen fuente de alimentación propia.

Las etiquetas RFID semi-pasivas incorporan una pequeña batería. Esta batería permite al circuito integrado de la etiqueta estar constantemente alimentado.

Las etiquetas RFID activas, deben tener una fuente de energía, y pueden tener rangos mayores y memorias más grandes que las etiquetas pasivas, así como la capacidad de poder almacenar información adicional enviada por el transmisor-receptor.

F*Firmware:*

Firmware o Programación en Firme, es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria tipo ROM, que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Al estar integrado en la electrónica del dispositivo es en parte hardware, pero también es software, ya que proporciona lógica y se dispone en algún tipo de lenguaje de programación. Funcionalmente, el firmware es el intermediario (interfaz) entre las órdenes externas que recibe el dispositivo y su electrónica, ya que es el encargado de controlar a ésta última para ejecutar correctamente dichas órdenes externas.

G*Grafo:*

En matemáticas y en ciencias de la computación, un grafo es el objeto abstracto básico de estudio en teoría de los grafos. Informalmente, un grafo se concibe y se representa como un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas. Las aristas pueden tener dirección (grafo dirigido).

J*Java:*

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas de punta.

L*Lector:*

Un lector de RFID tiene una o varias antenas, de forma que, por ejemplo, cuando una agrupación de mercancía identificada con EPC cruza a través de este dispositivo RFID, las antenas activan cada uno de los tags recogiendo simultáneamente la información de productos contenida en cada uno de ellos. Los lectores EPC están

situados en puntos estratégicos de la cadena de suministro con el fin de poder localizar los movimientos de los artículos.

O

Objeto:

Un objeto es una representación detallada, concreta y particular de un "algo". Tal representación determina su identidad, su estado y su comportamiento particular en un momento dado.

El estado de un objeto es el conjunto de valores concretos que lo caracterizan en un momento dado, como peso, color, precio, etc.

Orientado a Objetos:

La Programación Orientada a Objetos es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

P

PC:

Son las siglas en inglés de Personal Computer, que traducido significa Computadora Personal. Hay otras que se denominan Computadoras de escritorio, que son la gama de equipos utilizados en el hogar o en las oficinas y que no son portátiles, aunque esta categoría también podría considerarse una computadora personal.

Proceso:

Un proceso puede ser definido como un conjunto de actividades enlazadas entre sí que, partiendo de uno o más inputs (entradas) los transforma, generando un output (resultado).

RReader:

El Reader o lector son los encargados de enviar una señal RF para detectar las posibles etiquetas en un determinado rango de acción.

Registro:

Los datos de una base de datos se encuentran almacenados en campos, a cada uno de los cuales se le asigna un rótulo numérico que indica su contenido. Puede pensarse que el número de rótulo es el nombre interno del campo.

El conjunto de campos que contienen todos los datos de una unidad de información, se denomina registro.

Requisitos Funcionales:

Se basan en casos de uso, son listas que tienen el objetivo de:

- Capturar el comportamiento, sin especificar
- Comprensión común
- Validar arquitectura y sistema

RFID:

La identificación por radiofrecuencia o RFID (Radio Frequency Identification) es una tecnología de identificación, seguimiento y recuperación de datos mediante el uso de ondas de radio. Requiere tres elementos básicos: una etiqueta electrónica, un lector de etiquetas y una base de datos. La etiqueta consta de un circuito electrónico y una antena.

SSistema:

Un sistema es un conjunto de elementos relacionados. El concepto tiene dos usos muy diferenciados, que se refieren respectivamente a los sistemas de conceptos y a los objetos reales más o menos complejos y dotados de organización.

Software:

Se denomina software, equipamiento lógico o soporte lógico a todos los componentes intangibles de un ordenador o computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). Esto incluye aplicaciones informáticas tales como un procesador de textos, que permite al usuario realizar una tarea, y software de sistema como un sistema operativo, que permite al resto de programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de aplicaciones.

TTranspondedor:

La palabra Transponder deriva de TRANSmitter / resPONDER, lo cual explica su funcionamiento, en español se le conoce como Transpondedor (transmisor / receptor). Término utilizado para describir una etiqueta de RFID.

Anexos

Anexo A

El código fuente del sistema, se encuentra en el CD anexo a este trabajo, en la carpeta de nombre Código Fuente.

Anexo B

El Manual de Operaciones, se encuentra en el CD anexo a este trabajo con el nombre: ManualOperacional.pdf

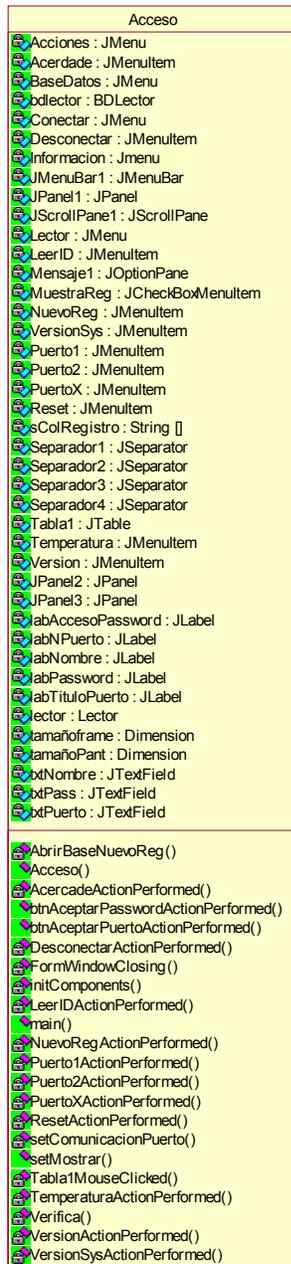
Anexo C

Las Especificaciones Técnicas del Lector se encuentran en el CD anexo a este trabajo con el nombre: MPRSerialCommunicationProtocol.pdf

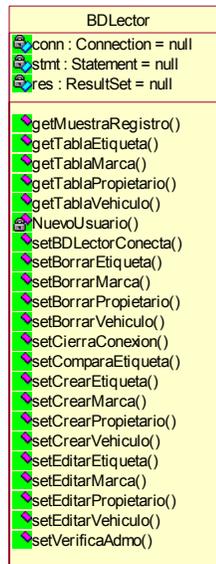
Anexo D

El Diagrama de Clases, está conformado por 10 Clases las cuales se muestran a continuación:

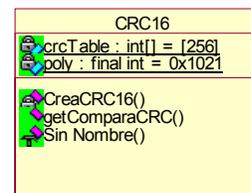
Clase Acceso



Clase BDLector



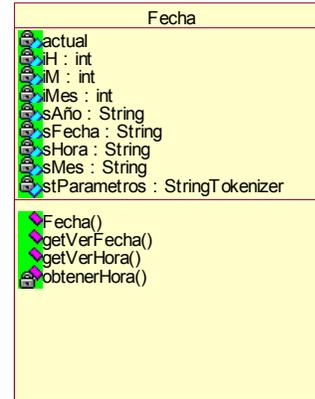
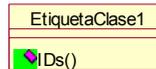
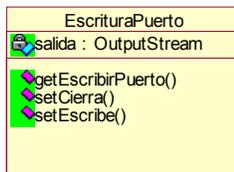
Clase CRC16



Clase EscrituraPuerto

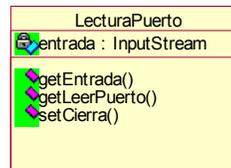
Clase EtiquetaClase1

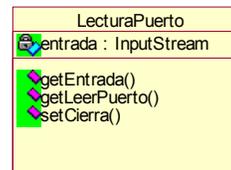
Clase Fecha



Clase Lector

Clase LecturaPuerto



*Clase Lector**Clase LecturaPuerto*

Clase NuevoRegistro (Seccionado en 3 partes)

NuevoRegistro		
acceso : Acceso	labPropietario : JLabel	btnAceptarEtiquetaActionPerformed()
bdlector : BDLector	labPropietarioID : JLabel	btnAceptarMarcaActionPerformed()
bbSalir : boolean	labPropietarioOV : JLabel	btnAceptarOVAActionPerformed()
btnAceptarEtiqueta : JButton	labPropietarioV : JLabel	btnAceptarPropietarioActionPerformed...
btnAceptarMarca : JButton	labVehiculoid : JLabel	btnAceptarVehiculoActionPerformed()
btnAceptarOV : JButton	Marca : JPanel	btnBorrarEtiquetaActionPerformed()
btnAceptarPropietario : JButton	Mensaje2 : JOptionPane	btnBorrarMarcaActionPerformed()
btnAceptarVehiculo : JButton	OpcionVehiculo : JPanel	btnBorrarPropietarioActionPerformed()
btnBorrarEtiqueta : JButton	Propietario : JPanel	btnBorrarVehiculoActionPerformed()
btnBorrarMarca : JButton	sColEtiqueta : String []	btnCrearEtiquetaActionPerformed()
btnBorrarPropietario : JButton	sColMarca : String []	btnCrearMarcaActionPerformed()
btnBorrarVehiculo : JButton	sColPropietario : String []	btnCrearVehiculoActionPerformed()
btnCrearEtiqueta : JButton	sColVehiculo : String []	btnCrearPropietarioActionPerformed()
btnCrearMarca : JButton	sIdEtiqueta : String	btnEditarEtiquetaActionPerformed()
btnCrearPropietario : JButton	sIdMarca : String	btnEditarMarcaActionPerformed()
btnCrearVehiculo : JButton	sIdPropietario : String	btnEditarPropietarioActionPerformed()
btnEditarEtiqueta : JButton	sIdVehiculo : String	btnEditarVehiculoActionPerformed()
btnEditarMarca : JButton	Tabla2 : JTable	cargarOpcionesVehiculo()
btnEditarPropietario : JButton	Tabla3 : JTable	cerrar()
btnEditarVehiculo : JButton	Tabla4 : JTable	initComponents()
comEtiquetaOV : JComboBox	Tabla5 : JTable	obtieneElementos()
comMarcaOV : JComboBox	tamañoFrame : Dimension	opcionesVehiculo()
comPropietarioOV : JComboBox	tamañoPant : Dimension	setControlarEtiqueta()
Etiqueta : JPanel	txtAño : JTextField	setControlarMarca()
Accion : int	txtColor : JTextField	setControlarPropietario()
VarbotonEtiqueta : int	txtDepartamento : JTextField	setControlarVehiculo()
VarbotonMarca : int	txtEtiqueta : JTextField	setNuevaEtiqueta()
VarbotonPropietario : int	txtEtiquetaId : JTextField	setOcultarMarca()
VarbotonVehiculo : int	txtEtiquetaV : JTextField	setOcultarEtiqueta()
CheckBox1 : JCheckBox	txtMarca : JTextField	setOcultarPropietario()
ScrollPane1 : JScrollPane	txtMarcaId : JTextField	setOcultarVehiculo()
ScrollPane2 : JScrollPane	txtModelo : JTextField	setValores()
ScrollPane3 : JScrollPane	txtPlaca : JTextField	Tabla2MouseClicked()
ScrollPane4 : JScrollPane	txtPropietario : JTextField	Tabla3MouseClicked()
TabbedPane1 : JTabbedPane	txtPropietarioId : JTextField	Tabla4MouseClicked()
labAño : JLabel	txtPropietarioV : JTextField	Tabla5MouseClicked()
labColor : JLabel	txtVehiculoid : JTextField	verificaEdicion()
labDepartamento : JLabel	Vehiculo : JPanel	
labEtiqueta : JLabel		
labEtiquetaId : JLabel		
labEtiquetaOV : JLabel		
labEtiquetaV : JLabel		
labMarca : JLabel		
labMarcaId : JLabel		
labMarcaOV : JLabel		
labMarcaV : JLabel		
labModelo : JLabel		
labPlaca : JLabel		

Clase Usuario

