

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE CIENCIAS



APLICACIONES DE LA TECNOLOGÍA RFID

TESIS

QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN CIENCIAS COMPUTACIONALES

PRESENTA:

JOSÉ ALBERTO IBARRA HERNÁNDEZ

Ensenada, Baja California. Junio de 2005

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE CIENCIAS

APLICACIONES DE LA TECNOLOGÍA RFID

TESIS PROFESIONAL

QUE PRESENTA

JOSÉ ALBERTO IBARRA HERNÁNDEZ

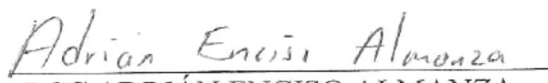


DR. JOSÉ ANTONIO GARCÍA MACÍAS
PRESIDENTE

APROBADO POR:



MC ADRIÁN VÁZQUEZ OSORIO
SECRETARIO



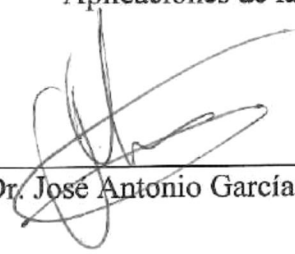
LCC ADRIÁN ENCISO ALMANZA
1er. VOCAL

Ensenada, Baja California. Junio de 2005

RESUMEN de la tesis de José Alberto Ibarra Hernández presentada como requisito parcial para la obtención de la Licenciatura en ciencias Computacionales. Ensenada, Baja California, México. 09 de Junio de 2005.

Aplicaciones de la Tecnología RFID

Resumen aprobado:


Dr. José Antonio García Macías
M.C. Luis Enrique Vizcarra Corral

En este trabajo se caracteriza la tecnología RFID (Radio Frequency IDentification), se señalan algunos de sus padecimientos para el fácil desarrollo de aplicaciones sobre esta misma tecnología. Toda esta investigación da como resultado el desarrollo de un conjunto de clases bases que facilitarán al usuario el desarrollo de aplicaciones RFID.

En los primeros capítulos del documento se abarca desde la problemática cubriendo el marco teórico de la tecnología RFID hasta la evaluación de tecnologías que auxiliaron en el entendimiento de la funcionalidad de los dispositivos RFID. En el capítulo IV se muestra el análisis y diseño de la herramienta. El capítulo V abarca la implementación de la herramienta. Los demás secciones y apéndices se muestra la instalación de la herramienta y paquetes de desarrollo tanto para la PC de escritorio y laptops como para PDAs (*Personal Digital Assistant*).

Dedicatoria

Esta tesis se la dedico con mucho cariño, amor y respeto a:

Primeramente a Dios por las fuerzas dadas para no desfallecer en el intento.

Mis padres Francisco Ibarra Pérez y Emilia Hernández Rosales porque con su ayuda, apoyo y cariño incondicional me dieron las fuerzas y motivación para cumplir con otra meta en vida.

Mis hermanos porque ellos supieron alentarte en momentos difíciles aunque ellos no lo supieran.

A mis sobrinos (los gemelos) Francisco y José Fernando quienes a pesar de sus dos años de vida me han enseñado que hay que esforzarse día a día para cumplir objetivos y alcanzar metas.

Y por último, al señor tiempo que me permitió cumplir esta meta.

Agradecimientos

Con mucha gratitud y honor menciono a las siguientes personas:

Carlos Fernando Caloca, Luis Pablo, Erika Vázquez, José Antonio González, Zunilda Kelinet y Mónica Pamela, sólo por mencionar algunos. Por la amistad ofrecida incondicionalmente.

Especialmente a:

Luis Demetrio Moreno León, el cual me apoyo con hardware y sin su ayuda no hubiera podido realizar mi tesis.

Mónica Anaya, quien paso vergüenzas al conseguirme una parte de mi kit de desarrollo.

M.C. Adrián Vázquez Osorio, por ser un amigo más y por el apoyo dado en el transcurso de mi formación académica.

CONACYT, por el apoyo económico dado.

Mis directores de tesis: **Dr. José Antonio García Macias** por haber confiado en mí en este proyecto y al **M.C. Luis Enrique Vizcarra Corral** quien me apoyo en todo momento en el transcurso de la tesis.

Aunque me faltaron muchos por mencionar para todos ellos mi más grande gratitud y afecto...

... gracias amigos.

CONTENIDO

I.	Introducción	1
I.1	Planteamiento del Problema	2
I.2	Objetivo(s)	4
I.3	Metodología	4
II	Marco Teórico	6
II.1	¿Qué es un Sistema RFID?.....	9
II.2	Algoritmo Anticolisión SID	13
II.3	Área de lectura entre el lector y la Etiqueta RFID	14
II.4	Ambientes de Aplicación de la Tecnología RFID	15
II.5	Software de RFID de <i>OpenSource</i>	17
II.5.1	RFDUMP*	17
II.6	Máquinas Virtuales para PDA's.....	18
II.6.1	KVM	19
II.6.2	Jeode EVM.....	21
II.6.3	SuperWava	22
II.7	CESerial.jar, Comm.jar y commapi.dll	22
II.8	Hyperterminal.....	23
III	Evaluación de Tecnologías	24
III.1	Kit RFID para el desarrollo del Proyecto	24
III.1.1	Lector RFID Syscan con Compact Flash Tipo II	24
III.1.2	Etiquetas Tag-it ISO 15693 de Texas Instruments	26
III.2	Análisis del Software Propietario	27
III.3	Análisis de la etiqueta RFID	29
III.3.1	Protocolos de los comandos de lectura y escritura	30
III.3.2	Pruebas de concepto en lenguaje Java.....	31
III.3.3	Comunicación entre Lector y Etiqueta	34
III.4	Análisis de la Herramienta RFDUMP	35
IV	Desarrollo de Clases Base de Java para RFID	37
IV.1	Análisis de la Arquitectura del sistema.....	38
IV.1.1	Abstracciones Clave	38
IV.2	Análisis.....	39
IV.2.1	Requerimientos.....	39
IV.2.2	Casos de Uso Generales	40
IV.2.1.1	Caso de Uso Manejo de Información	40
IV.2.1.2	Caso de Uso Control de Comunicación	42
IV.2.1.3	Caso de Uso Configurar	43
IV.2.3	Casos de Uso Específicos.....	45
IV.2.2.1	Caso de uso Apagar Lector.....	46

Diagrama de Secuencia.....	47
Diagrama de Colaboración	48
Diagrama de Clases.....	48
IV.2.2.2 Caso de uso Prender Lector.....	49
Diagrama de secuencia	50
Diagrama de Colaboración	51
Diagrama de Clases.....	51
IV.2.2.3 Caso de uso Detener lectura	52
Diagrama de secuencia	53
Diagrama de Colaboración	54
Diagrama de Clases.....	54
IV.2.2.4 Caso de uso: Lectura Continua	55
Diagrama de secuencia	57
Diagrama de Colaboración	58
Diagrama de Clases.....	58
IV.2.2.5 Caso de uso Lectura Simple	59
Diagrama de secuencia	60
Diagrama de Colaboración	62
Diagrama de Clases.....	62
IV.2.2.6 Caso de uso Escribir datos.....	63
Diagrama de secuencia	65
Diagrama de Colaboración	66
Diagrama de Clases.....	66
IV.2.2.7 Caso de uso Obtener Versión Lector	67
Diagrama de secuencia	68
Diagrama de Colaboración	69
Diagrama de Clases.....	69
IV.2.2.8 Caso de uso Configurar	70
Diagrama de secuencia	71
IV.3 Diseño.....	72
IV.3.1 Arquitectura de la Herramienta	72
IV.3.2 Diagrama General de Clases	74
IV.3.3 Criterios de Diseño	75
IV.3.4 Diagramas de Interacción en la Realización del Caso de Uso.....	76
IV.3.4.1 Caso de Uso: Apagar Lector.....	76
IV.3.4.2 Caso de Uso: Prender Lector	77
IV.3.4.3 Caso de Uso: Detener Lectura	78
IV.3.4.4 Caso de Uso: Lectura Continua	79
IV.3.4.5 Caso de Uso: Lectura Simple	80
IV.3.4.6 Caso de Uso: Escribir Datos	81
IV.3.4.7 Caso de Uso: Obtener Versión	82
IV.3.5 Vista de Clases Participantes en la Realización del Caso de Uso.....	83
IV.3.5.1 Caso de Uso: Apagar Lector.....	83
IV.3.5.2 Caso de Uso: Prender Lector	84
IV.3.5.3 Caso de Uso: Detener Lectura	84
IV.3.5.4 Caso de Uso: Lectura Continua	85
IV.3.5.5 Caso de Uso: Lectura Simple	85
IV.3.5.6 Caso de Uso: Escribir datos.....	86
IV.3.5.7 Caso de Uso: Obtener Versión del Lector	86
V Implementación.....	87

V.1	Implementación para PDA.....	87
V.1.1	Protocolo del Puerto Serie.....	89
V.2	Implementación para entornos Win32.....	90
V.2.1	Protocolo del Puerto Serie.....	90
VI	Conclusiones y Perspectivas	91
VII	Apéndice A. Tipos de Lectores	93
VII.1	Lector RFID para Alta Frecuencia.....	93
VII.2	Lector RFID para Baja Frecuencia	93
VII.3	Lectores RFID para Ultra Alta Frecuencia.....	94
VII.4	Lector para PDA.....	96
VII.5	Lectores para Identificadores de Animales	97
VIII	Apéndice B. Tipos de Transponders.....	99
VIII.1	Tipos de Etiquetas	99
VIII.2	Modos de Operación.....	100
VIII.3	Formas de las Etiquetas	102
VIII.4	Transponders Pasivos TIRIS HDX 134,2 Khz. (LF).....	102
VIII.4.1	Ventajas clave	103
VIII.4.2	Capacidades de Memoria de Datos.	103
VIII.4.3	Modelos de Transponder	104
VIII.5	Transponders Pasivos Tag-it HF 13,56 MHz.....	107
VIII.5.1	Ventajas clave	108
VIII.5.2	Capacidades de memoria de datos.....	108
VIII.5.3	Modelos de Transponder	110
VIII.6	Tipos de Identificadores para animales.....	111
VIII.7	Tipos de Transponder Pasivos en Frecuencia Ultra Alta (UHF)	112
IX	Apéndice C: Paquetes para Puertos Serie	114
IX.1	Instalación y Ejecución en PDA.....	114
IX.1.1	Formas de Ejecutar una Aplicación Java en Jeode	115
IX.2	Instalación en Computadoras de Escritorio	117

Índice de Figuras

Figura 1 Funcionamiento de un sistema RFID.....	10
Figura 2 Comunicación Lector – Transponder RFID	11
Figura 3 Funcionamiento de las Etiquetas RFID	12
Figura 4 Petición por Demanda	12
Figura 5 Lectura Múltiple de Etiquetas	13
Figura 6 Organización de la Memoria de una etiqueta HF	30

<i>Figura 7 Casos de Uso Generales</i>	40
<i>Figura 8 Caso de uso: Manejo de Información</i>	45
<i>Figura 9 Caso de uso: Control de Comunicación</i>	45
<i>Figura 10 Caso de uso: Configurar</i>	45
<i>Figura 11 Diagrama de componentes: Entornos Windows CE para Pocket PC</i>	88
<i>Figura 12 Diagrama de componentes para la aplicación en entornos Windows 32</i>	90

Índice de Tablas

<i>Tabla 1 Distancias aproximadas dependiendo la frecuencia usada [Web 4 op. cit]</i>	15
<i>Tabla 2 Lista de paquetes de la KVM</i>	21
<i>Tabla 3 Estructura del Marco de la Petición del Transponder</i>	34
<i>Tabla 4 Estructura del Marco de la Respuesta del Transponder</i>	34

I. Introducción

Hoy en día ha surgido una nueva manera de autoidentificar casi cualquier tipo de objeto (carros, productos, animales, etc.) ya sea en las tiendas de ropa, mercados, seguimientos de productos y/o materiales en las líneas de producción de las industrias y en accesos de vehículos. Todo lo anterior se logra gracias a la tecnología RFID – **Radio Frequency IDentification**-, si bien, no es relativamente nueva, pero se ha estado acrecentando su uso de manera comercial y menos restringida en los sectores privados y públicos, pese a que existen grupos que están en contra del uso de esta tecnología ya que de alguna manera amenaza de cierta forma la privacidad de las personas, esto no ha podido parar su desarrollo y expansión en el mundo del comercio, en el sector detallista y el sector minorista.

Las aplicaciones (ver II.4) con tecnología RFID pueden tener una amplia gama de aplicaciones, algunas de ellas ya se encuentran en proceso de desarrollo, pero quizás las más interesantes sean las que todavía están por concebirse. En efecto, como ya se mencionó antes, apenas comienza el inicio de esta nueva era.

I.1 Planteamiento del Problema

Como se mencionó anteriormente es indispensable experimentar con este tipo de plataformas, probar capacidades actuales, imaginando y concibiendo las futuras.

Es evidente entonces que, al encontrarse en una etapa incipiente de desarrollo, el campo de la tecnología RFID ofrece muchos retos. Por mencionar algunos: falta de API's, middlewares, etc. De entre todos ellos se abordará el reto de proveer, a los desarrolladores, herramientas adecuadas para el desarrollo de aplicaciones utilizando como plataforma la tecnología RFID.

Sin embargo, al querer desarrollar las aplicaciones con la tecnología RFID los desarrolladores se enfrentan a varios problemas: el primero es que una solución con dispositivos RFID son muy costosas ya que tanto este tipo de dispositivos al igual que el desarrollo de aplicaciones son caros.

Otro problema es que cada empresa o proveedor de un lector RFID define su propio conjunto de funciones diferente a los demás lectores de las demás. Esto da pie a que la aplicación desarrollada sea específicamente para ese tipo de lector y cuando se cambie de lector se tenga que realizar la aplicación de nuevo.

Un tercer problema es el nivel bajo de abstracción respecto a la programación ya que algunos lectores RFID aún son manipulados bajo comandos enviados por el puerto serie del dispositivo que al que están conectados y además de que la información que manejan tanto el lector y la etiqueta RFID esta en formato hexadecimal.

Debido a la problemática antes presentada esta área de desarrollo de software para tecnología RFID constituye una oportunidad interesante para hacer contribuciones tanto a la sociedad de desarrolladores como a los sectores públicos y privados ofreciendo soluciones RFID.

I.2 Objetivo(s)

El objetivo principal es:

Proveer a los desarrolladores de un conjunto de clases bases que les faciliten la creación de aplicaciones con tecnología RFID a un nivel de abstracción alto respecto a la programación.

Así pues, algunos objetivos específicos que se derivan del objetivo principal, arriba mencionado, son:

OBJ1. Identificar el desarrollo actual de aplicaciones con tecnología RFID.

OBJ2. Análisis de las herramientas propietarias y *Open Source* (Código Abierto)

OBJ3. Conformar un Kit RFID para el desarrollo

OBJ4. Desarrollar un API que puedan utilizar los desarrolladores para construir aplicaciones con dispositivos RFID.

I.3 Metodología

Para la consecución de los objetivos ya descritos se usará una metodología que involucra aspectos tanto teóricos como empíricos, pero también generaremos pruebas de conceptos con base en prototipos funcionales. Los pasos a seguir para este desarrollo metodológico se enuncian a continuación.

1. **Contextualización.** Hacer una investigación bibliográfica para conocer las diferentes plataformas hardware para formar aplicaciones con tecnología RFID, determinar sus características. De igual forma, caracterizar el desarrollo de aplicaciones sobre dichas plataformas.

2. **Propuesta de alternativas.** Tras caracterizar el desarrollo sobre la plataforma RFID, se determinará el nivel de abstracción que resulta adecuado para los desarrolladores. Esto dará pie a la definición de interfaces de desarrollo (API's) que permitan el acceso a componentes simples que puedan ser integrados para el desarrollo de sistemas potencialmente complejos.
3. **Pruebas de concepto.** Con el fin de probar las propuestas hechas, se implementarán algunos de los componentes propuestos. Dichos componentes podrán integrarse con otros y probar su comportamiento.
4. **Validación y refinamiento.** Mediante las pruebas empíricas del paso anterior se podrán entonces validar nuestras herramientas, determinar sus capacidades y limitaciones, para entonces entrar en un ciclo donde se vayan refinando las propuestas.

II Marco Teórico

A pesar de que la tecnología RFID ha estado disponible desde los años cuarenta, en ése entonces, era utilizada para aplicaciones de radares principalmente. Es apenas hace algunos años que la tecnología RFID ha mejorado sustancialmente y bajado de precio lo suficiente como para ser considerada para múltiples aplicaciones e incorporada en múltiples productos [Malatto, *et. al.*; 2005].

RFID (*Radio Frequency IDentification*) es el término genérico dado al uso de ondas de radio para identificar objetos individuales, como productos en un almacén, animales o personas, reduciendo así los errores que pueda ocasionar la intervención humana [Malatto, *op. cit.*]. En si, es un sistema de auto identificación de bienes, el cual utiliza lectores y etiquetas para lograr su cometido [José Camilo Daccach T, 2005; Ann Cavoukian, 2004].

Entre los beneficios que ofrece la RFID se encuentran la disminución del número de etiquetas impresas, la reducción del número de impresores de códigos de barras y de etiquetadores en plantas, depósitos o plataformas de los centros de distribución y la simplificación en los procesos de lectura y escritura. De igual forma, la RFID permite la repetición de un tipo de dato en particular para productos iguales, pero con la diferencia de que se asigna un número diferente a cada artículo para evitar duplicidades en el identificador [Claudia M, 2004].

A diferencia del código de barras, la tecnología de RFID no requiere de un campo visual directo entre la etiqueta RFID y el lector. Los lectores de RFID pueden

leer etiquetas RFID a través de cajas de cartón o inclusive a través de paredes. La tecnología de RFID es mucho más versátil que el código de barras y en algunos casos permite implementar nuevas aplicaciones que anteriormente no eran posibles con la tecnología de código de barras.

En resumen las ventajas de las etiquetas electrónicas son las siguientes:

- A diferencia del código de barras, las etiquetas electrónicas no necesitan contacto visual con el módulo lector para que éste pueda leerlas. La lectura se puede hacer a una distancia de hasta 10 metros.
- Mientras el código de barras identifica un tipo de producto, las etiquetas electrónicas identifican cada producto individual. Es decir, dos yogures iguales llevan ahora el mismo código de barras y, por lo tanto, la misma identificación, pero si estuvieran equipados con etiquetas electrónicas se podrían identificar y gestionar de forma individual.
- La tecnología RFID permite leer múltiples etiquetas electrónicas simultáneamente. Los códigos de barras, por lo contrario, tienen que ser leídos secuencialmente. Esta característica del sistema de autoidentificación por radiofrecuencia ofrece diversas ventajas como, por ejemplo, la reducción del tiempo de espera en las colas de los supermercados.

- Las etiquetas electrónicas pueden almacenar mucha más información sobre un producto que el código de barras, que solo puede contener un código y, en algunos casos, un precio o cantidad.
- Mientras que sobre el código de barras se puede escribir solo una vez, sobre las etiquetas electrónicas se puede escribir todas las veces que haga falta.
- La tecnología RFID evita falsificaciones. Con una simple fotocopia se puede reproducir un código de barras. Las etiquetas electrónicas, en cambio, no se pueden copiar. Un tag sobre un artículo de marca garantiza su autenticidad.
- Un código de barras se estropea o se rompe fácilmente, mientras que una etiqueta electrónica es mas resistente porque, normalmente, forma parte del producto o se coloca bajo una superficie protectora y soporta mejor la humedad y la temperatura.

II.1 ¿Qué es un Sistema RFID?

Un sistema RFID consiste de tres componentes: una antena, un receptor, y un emisor. La antena transmite mediante ondas de radiofrecuencia una señal que activa el emisor. Cuando se activa, el emisor envía información hacia la antena, y con base en esta información se puede programar un controlador para que tome acción. En la actualidad la antena y el receptor están juntos y se le denomina “Lector”.

El lector y la etiqueta pueden ser tanto receptores como emisores, ya que el receptor ordena a la antena enviar una señal de radiofrecuencia, la cual es recibida por la etiqueta RFID y la misma etiqueta emite la respuesta que será recibida por el lector mediante la antena lectora.

En términos generales el lector envía una serie de ondas de radiofrecuencia al conjunto de etiquetas o en su defecto a una sola etiqueta, que esta capta a través de una pequeña antena [Web 7] (ver **Figura 1**). Estas ondas activan el microchip, que mediante la micro-antena y la radiofrecuencia transmite al lector su número único de identificación. Una vez que el lector ha recibido el código único de identificación del producto, lo transmite al dispositivo que lo manipula (en este caso la computadora), para que posteriormente se haga el procesamiento de los datos. La comunicación del lector con la computadora puede ser una red tipo Ethernet, puerto serial (interfaz RS232 entre otras interfaces), PCMCIA y Flash Card [Web 8]. Un ejemplo del procesamiento de datos sería que en la computadora conectada al lector contenga una base de datos, donde se han almacenado previamente las características del artículo

en cuestión: fecha de caducidad, material, peso, dimensiones, etc. De este modo se hace posible consultar la identidad de una mercancía en cualquier momento y fácilmente durante toda la cadena de suministro.

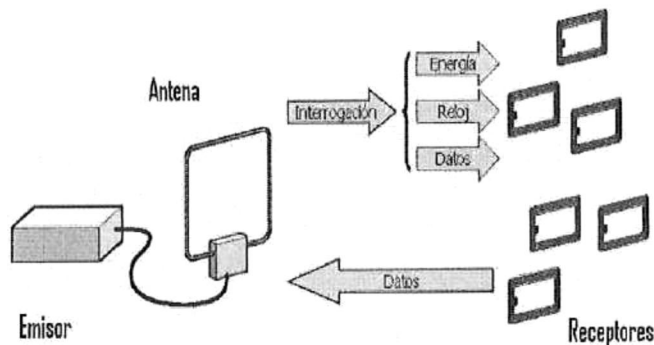


Figura 1 Funcionamiento de un sistema RFID.

Existen en el mercado etiquetas RFID que además de contar con un identificador único permiten almacenar información adicional que puede ser alterada dependiendo de las necesidades. El costo de las etiquetas RFID varía desde los 20 centavos de dólar hasta varios cientos de dólares dependiendo de la sofisticación de la etiqueta.

Hay dos tipos de sistemas RFID [Web 8 op. Cit, Web 4, Web 5]: los pasivos y los activos. Un sistema RFID pasivo utiliza etiquetas pasivas (ver Apéndice B. Tipos de *Transponders*), se utiliza en aplicaciones que requieren un área de lectura pequeña, y la tasa de lectura dependerá de la frecuencia que se utilice. Un sistema RFID activo utiliza etiquetas activas (ver Apéndice B. Tipos de *Transponders*), se

utiliza en aplicaciones que requieren un área de lectura más amplia, y la tasa de lectura de igual forma dependerá de la frecuencia que se utilice.

En la **Figura 2** se muestra como es la comunicación entre el lector RFID y el *transponder* en el área de radio frecuencia [Web 19]. Además se puede apreciar como es posible automatizar algún proceso o controlar los objetos que tengan adherido un *transponder* teniendo el lector conectado a la computadora.



Figura 2 Comunicación Lector – Transponder RFID

En la **Figura 3**, se muestra la conexión de un lector RFID a una computadora por medio de una interfaz RS232 ó RS422 ó también llamadas puerto serie. Una computadora podrá albergar en sus ranuras de expansión más de un lector, siempre de acuerdo a la posibilidad de ampliación de puertos de la computadora. Además, muestra el orden del módulo de control y el módulo de Radio Frecuencia, el primero controla la comunicación del lector con la computadora y el segundo controla la manipulación de la antena lectora / escritora.

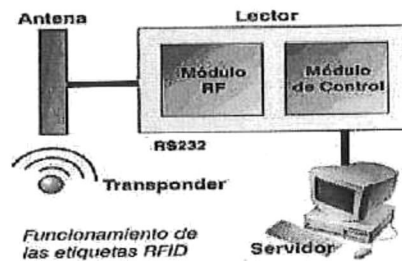


Figura 3 Funcionamiento de las Etiquetas RFID

En la Figura 4 es posible ver como la antena lectora emite una señal de radio, solicitando de este modo que el *transponder* responda a la señal de demanda de transmisión de datos. Ya que cualquier tipo de *transponder* no puede emitir su información a menos que le sea pedida por un lector.

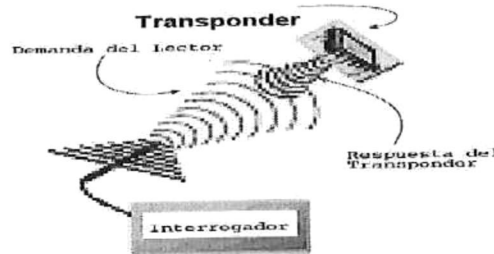


Figura 4 Petición por Demanda

La **Figura 5** permite ver como está determinada el área de lectura de una antena lectora. Todos los *transponder* que se encuentren dentro del área de lectura transmiten sus datos por demanda del Lector. No obstante, puede haber infinidad de *transponders* en dicha área, éstos en caso de colisión, se ordenan para transmitir según un algoritmo matemático denominado SID por el Transmisor de la Antena Lectora.

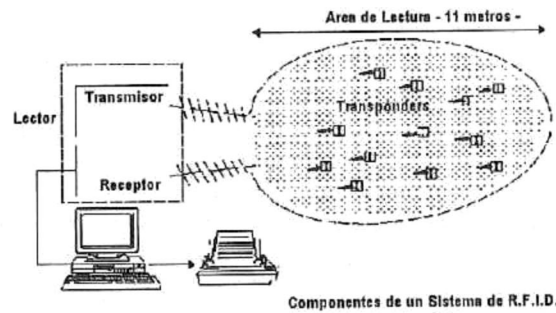


Figura 5 Lectura Múltiple de Etiquetas

II.2 Algoritmo Anticolisión SID

El mecanismo SID (*Simultaneous Identification*) ofrece la capacidad de inventariar un gran número de etiquetas por sus direcciones únicas en un corto tiempo, previendo las que están dentro del área operativa del lector [Web 7, op. cit.].

El mecanismo del SID está basado en un algoritmo administrado por el lector:

- Cada transponder tiene una dirección única
- Los *transponders* son interrogados por una secuencia de peticiones SID, usando un esquema de direccionamiento especial.
- Si dos o más *transponders* contestan a la misma interrogación, las colisiones son detectadas y almacenadas en la memoria del lector.
- Si solamente un transponder contesta, su dirección es registrada en la memoria del lector. Así, este no será interrogado de nuevo durante el actual ciclo del SID.

- El lector entonces ejecuta el algoritmo *collision-handling*, el cual resulta en la interrogación de *transponders* que todavía no están identificados por un esquema de direccionamiento modificado.
- El ciclo es continuo hasta que ya no se detecta la colisión, por ejemplo: todos los *transponders* han sido identificados y sus direcciones almacenadas en la memoria del lector. El inventario de los *transponders* está entonces completo.

El algoritmo *collision-handling* teóricamente permite un 100% de probabilidad de que los *transponders* que están presentes en el área del lector sean detectados. Sin embargo, en aplicaciones reales, la influencia mutua de los *transponders* puede afectar el resultado.

El algoritmo de SID no siempre utiliza toda su potencialidad, ya que se utiliza sólo cuando hay más de un *transponder* en un área determinada y siempre y cuando se quieran leer al mismo tiempo todos los *transponders* presentes. Por otro lado, aunque se encuentren demasiados *transponders* en el área de lectura del lector y sólo se pretenda leer de uno en uno, entonces una simple petición SID es necesaria para la lectura de cada *transponder*.

II.3 Área de lectura entre el lector y la Etiqueta RFID

Las áreas de lectura (ver Tabla 1) entre las etiquetas y lectores RFID dependen de tres cosas: del tipo de lector que se utilice, la antena asociada al lector y el tipo de

transponder. La frecuencia que se utilice también influye en el tamaño del área de lectura [Roy W., 2004].

Frecuencia	134 KHz	13.56 MHz	UHF	2.45 GHz	5.8 GHz
Distancia	40 - 60 cm.	20 - 40 cm.	3 - 6 m	1 - 3 m	0.5 - 1.5 m

Tabla 1 Distancias aproximadas dependiendo la frecuencia usada [Web 4 op. cit]

II.4 Ambientes de Aplicación de la Tecnología RFID

La tecnología RFID se puede utilizar para un sin número de aplicaciones. En general, dichas aplicaciones caen dentro de alguna de las tres categorías mencionadas a continuación:

- **Identificación de objetos (productos, documentos, etc.):** Con la tecnología RFID es posible identificar objetos a nivel individual con un número de serie único.
- **Autenticación de productos:** La tecnología RFID permite determinar si un objeto es genuino o ha sido falsificado. También se puede verificar si el objeto en cuestión, aunque sea genuino, ha sido robado.
- **Rastreo de productos:** Productos con tecnología RFID pueden ser rastreados de manera más efectiva y automática. Algunos expertos creen que la tecnología RFID permitirá crear una nueva capa encima de Internet en la cual, objetos con etiquetas RFID se comuniquen entre sí sin intervención humana. A esta visión la han llamado "El Internet de las cosas" (The Internet of things).

A continuación se muestra un tipo de aplicación RFID:

Pases de esquís y Parques de atracciones

Chile Locating System (Sistema de Localización de Niños)

SafeTzone Technologies Corporation, con domicilio en Laguna Hills, California, ha desarrollado un sistema de localización en tiempo real para parques temáticos. Este sistema permite la trazabilidad constante de personas que se encuentran dentro de un parque de atracciones, ya que son lugares donde se acumula gran cantidad de gente y son en estos lugares donde las personas menores son propicias a perderse. Este sistema se basa en etiquetas activas RFID las cuales tienen la forma de una pulsera de mano y en lectores de largo alcance. Por la gran eficacia y fiabilidad que proporciona este sistema actualmente está funcionando en algunos parques temáticos de los USA. Toda la información acerca de este sistema se puede encontrar en la siguiente liga http://dc.editricetemi.com/ed_temi/dcFr/articulo.jsp?idArt=17.

Las secciones II.5, II.6 y II.7 que a continuación se describen fueron realizadas con el objetivo de entender mejor los tipos de lenguajes utilizados para programar aplicaciones RFID y los limitantes de los dispositivos que contienen a estas aplicaciones. Y así, poder tener una vista mas clara del lenguaje que mas nos convenía para realizar nuestras clases bases.

II.5 Software de RFID de *OpenSource*

Se realizó una búsqueda, en los sitios de Internet *Freshmeat.net* [Web 21] y *SourceForge.net* [Web 22], de aplicaciones de libre distribución con licencia GPL (*General Public License*) desarrolladas para la tecnología RFID, sin embargo la gran mayoría de estas aplicaciones se encuentran en etapa de planeación y no están liberadas. En *SourceForge* se localiza: *JRFID*, la cual será una biblioteca de clases para Java, *Singularity* se pretende sea un middleware que se puede utilizar en las cadenas de producción. En *Freshmeat* están: *RFID:BASE* que plantea un conjunto de clases bases para desarrollar sistemas RFID, *RFID:Alien*, *RFID:MAtrics* y *RFID:EPC* todas ellas serán implementadas exclusivamente con lenguaje Perl.

En el sitio *RF-Dump.org* se encontró una aplicación liberada denominada **RF-DUMP** de código libre (*Open Source*).

II.5.1 RFDUMP*

Este software está basado en el lenguaje Java pudiendo así ser ejecutado en una máquina virtual de Java utilizando el paquete *javax.comm* para sistemas operativos

* RFDUMP es un software Open Source con licencia GPL realizado por Lukas Grunwal y Boris Wolf ;
<http://www.rf-dump.org>

win32 para realizar las llamadas por el puerto serie de la computadora. En las especificaciones de la aplicación RFDUMP se indica que la aplicación corre en PDA's con Linux, a partir de este hecho se formuló la interrogante de que si también funcionaría en una PDA que tuviera Windows CE.

RFDUMP es una herramienta para detectar etiquetas RFID y mostrar la información que contienen, la cual es: el identificador de la etiqueta, tipo de la etiqueta, la manufacturadora de la etiqueta, etc. Los datos que contiene la memoria de usuario de la etiqueta pueden ser desplegados. Los datos también pueden ser modificados usando un editor que le convierta los datos a cantidades hexadecimales. En adición, la *cookie** integrada en la aplicación RFDUMP demuestra lo fácil que es para una compañía abusar de la tecnología RFID para espiar a los usuarios. RFDUMP trabaja con el lector ACG Multi-Tag o un lector de tarjeta. En base a las características antes mencionadas, el objetivo de esta aplicación es dejar ver a flor de piel la vulnerabilidad de la seguridad de los sistemas RFID.

II.6 Máquinas Virtuales para PDA's

Una máquina virtual de Java (JVM) es un programa encargado de interpretar código intermedio (bytecode) de los programas Java precompilados a código máquina ejecutable por la plataforma. De esta forma, la JVM proporciona al programa Java independencia de la plataforma con respecto al hardware y al sistema operativo subyacente. Las implementaciones de las JVM's cubren desde computadoras

* Las "cookies" (que en inglés significa "galletitas") son pequeños archivos de texto alojados en el disco duro del usuario y su función es guardar datos

personales hasta dispositivos móviles. Por mencionar algunas JVM's para estos dispositivos están: Jeode, SuperWava, etc. Todas basadas en la máquina virtual KVM de Sun [Sergio Gálvez R, Lucas Ortega D.; 2003].

II.6.1 KVM*

KVM es la Máquina Virtual más pequeña desarrollada por Sun. Su nombre KVM proviene de Kilobyte (haciendo referencia a la baja ocupación de memoria, entre 40Kb y 80Kb). Se trata de una implementación de Máquina Virtual reducida y especialmente orientada a dispositivos con bajas capacidades computacionales y de memoria. La KVM fue diseñada para ser:

- Pequeña, con una carga de memoria entre los 40Kb y los 80 Kb, dependiendo de la plataforma y las opciones de compilación.
- Alta portabilidad.
- Modulable.
- Lo más completa y rápida posible y sin sacrificar características para las que fue diseñada.

Sin embargo, esta baja ocupación de memoria hace que posea algunas limitaciones con respecto a la clásica Java Virtual Machine (JVM):

* KVM fue desarrollada por Sun Microsystems; <http://www.sun.com>

1. No hay soporte para tipos en punto flotante. No existen por tanto los tipos `double` ni `float`. Esta limitación está presente porque los dispositivos carecen del hardware necesario para estas operaciones.
2. No existe soporte para JNI (*Java Native Interface*) debido a los recursos limitados de memoria.
3. No existen cargadores de clases (*class loaders*) definidos por el usuario. Sólo existen los predefinidos.
4. No se permiten los grupos de hilos o hilos *daemon*. Cuando se quiere utilizar grupos de hilos se utilizan los objetos colección para almacenar cada hilo en el ámbito de la aplicación.
5. No existe la finalización de instancias de clases. No existe el método `Object.finalize()`.
6. Limitada capacidad para el manejo de excepciones debido a que el manejo de éstas depende en gran parte de las API's de cada dispositivo por lo que son éstos los que controlan la mayoría de las excepciones.

La configuración de esta máquina virtual está clasificada en Configuración de Equipos Limitados por Conexión (CLDC, *Connected Limited Devices Configuration*). La CLDC está orientada a dispositivos dotados de conexión y con limitaciones en cuanto a capacidad gráfica, cómputo y memoria. Un ejemplo de estos dispositivos son: teléfonos móviles, localizadores (pagers), PDAs, organizadores personales, etc.

Esta máquina está limitada en cuanto a paquetes de desarrollo, los paquetes que incluye son:

Nombre del Paquete CLDC	Descripción
java.io	Clases y paquetes estándar de E/S. Subconjunto de J2SE
java.lang	Clases e interfaces de la Máquina Virtual. Subconjunto de J2SE.
java.util	Clases, interfaces y utilidades estándar. Subconjunto de J2SE.
Javax.microedition.io	Clases e interfaces de conexión genérica CLDC

Tabla 2 Lista de paquetes de la KVM

II.6.2 Jeode EVM+

Jeode EVM es la versión de la máquina virtual de Java para ambientes de Windows CE basada en la KVM de Sun, se utiliza para poder ejecutar *applets* y aplicaciones desarrollados con el lenguaje de Java. Jeode EVM soporta las versiones más actuales de Java, también soporta la mayoría de los paquetes que ofrece Java con excepción del paquete swing, paquete comm.jar –el cual sirve para la manipulación de la comunicación de los puertos serie- y los vectores.

Después de instalar el Jeode EVM en la PDA con Windows se instaló la aplicación RFDUMP para ver su funcionamiento sobre PDA's -aunque esta no estaba probada en una PDA con Windows- pero no se obtuvieron resultados favorables, una vez analizado los resultados obtenidos al respecto, se concluyó la necesidad de instalar una versión del paquete comm.jar para Windows CE. Después de la instalación del paquete en la PDA y correr de nuevo la aplicación se obtuvieron resultados favorables. En base a estos resultados de éxito de ejecución de

⁺ Jeode EVM Runtime fue desarrollada por Insignia Solutions; <http://www.insignia.com/>

aplicaciones desarrolladas en Java tanto en computadoras de escritorio, laptops y PDA's se determinó la factibilidad de la utilización del lenguaje Java como herramienta base de desarrollo.

El paquete comm.jar de Java utilizado en las versiones de la máquina virtual de Java para computadoras de escritorio no es compatible con Jeode EVM. Se hace una suposición de que las mínimas limitaciones que puede tener Jeode EVM son las que tiene la KVM de Sun.

II.6.3 SuperWava°

Es una versión de la máquina virtual de Java para ambientes de Windows CE, PalmOS y plataformas Windows. SuperWava no contiene los paquetes java.util, java.io, awt, swing, etc. La SuperWaba.pdb contiene los paquetes waba.lang y waba.io/afx/ui/util/sys. A pesar de que es una versión de la máquina virtual de Java, superwava no es Java y aunque es un 99% compatible con el lenguaje Java éste no puede ser llamado por la máquina virtual debido a las restricciones del copyright

II.7 CSerial.jar, Comm.jar y commapi.dll•

Los paquetes CSerial.jar, Comm.jar y commapi.dll son las versiones de la implementación del Java™ Communications API para ser utilizadas en las arquitecturas Pocket PC con Windows CE como sistema operativo. Este es conjunto de clases que sirven para manipular la comunicación a través del puerto serie de las PDA's y fueron desarrolladas por James N., 2000-2002.

° SuperWava es una versión de la máquina virtual de java desarrollada por SuperWava Companion;
<http://www.superwaba.com.br>

• CSerial.jar, Comm.jar y commapi.dll son propiedad de James Nord, 2000-2002.

II.8 Hyperterminal

La Hyperterminal es una herramienta desarrollada por Hilgraeve para Microsoft, a su vez éste la incluye en todas sus versiones de Windows 95, Windows 98, Windows Me, Windows NT 4.0, Windows 2000 y Windows XP. Esta herramienta ofrece muchas funcionalidades muy interesantes, una de ellas es que se pueden hacer llamadas por el puerto serie –previa configuración del puerto- y manipular él o los dispositivo(s) que esté(n) utilizando un puerto determinado en ese instante. Por las cualidades de esta herramienta de mandar y recibir datos por el puerto serie, la empresa proveedora del lector RFID la recomienda para llevar a cabo las pruebas de funcionalidad de este lector [Web 20]. Esta herramienta tiene la propiedad de poder crear en un archivo de texto una lista de comandos o un sólo comando que se desea ejecutar en el dispositivo y enviarlo por el puerto serie y así ejecutar varios comandos al mismo tiempo.

[†] Hilgraeve Inc., Copyright 1996, 1997, 1998, 1999, 2000, 2001; <http://www.hilgraeve.com/hpte.html>

III Evaluación de Tecnologías

III.1 Kit RFID para el desarrollo del Proyecto

El kit RFID para el desarrollo y pruebas de conceptos se conformó de 2 iPAQ's modelo H3900, un adaptador compact flash a PCMCIA, 5 etiquetas pasivas RFID número parte #RI-I03-112B, 5 etiquetas pasivas RFID número parte #RI-I02-112B, 5 etiquetas pasivas RFID número parte #RI-I01-112B, las etiquetas pasivas son *Tag-it ISO 15693 de Texas Instruments*, 1 lector RFID Syscan con compact flash tipo II y una laptop con procesador celeron de 433 MHz.

III.1.1 Lector RFID Syscan con Compact Flash Tipo II

Al buscar sitios de proveedores en Internet donde comprar este lector RFID se encontraron varios sitios tanto de México como de Estados Unidos de América y Canadá. La compra de este lector se realizó directamente al fabricante en Canadá ya que este tipo de lector RFID es el más accesible en el mercado.

Por las características que ofrece este lector, la compatibilidad con varios tipos de *transponders* y porque se adecuaba a los fines y presupuesto del proyecto, se optó por el lector *Syscan con Compact Flash Tipo II*. Este lector RFID funciona con interfaces Compaq Flash Tipo II o con adaptador PCMCIA. Este tipo de interfaces ofrecen que el lector pueda ser utilizado tanto en PDA's, computadoras de Escritorio, Organizadores y *laptops*. Este tipo de lector es compatible con la todas las versiones de Windows y algunas versiones de Linux. Además de que ofrece una amplia compatibilidad con muchos tipos de *transponders* RFID.

Los cuales son: I-Code2, ISO 15693, ISO 14443, Tag-it, I-Code SLI, Tag-it HFI, LR 512, SRF55VxxP, SRF55VxxS1, TempSense (KSW), Mifare Ultralight, ISO 14443C (Felica) 1,3, Mifare Standard 1,3, Mifare ProX1, 3, SLE55Rxx1, 3, SLE66CL160S1, 3.

El lector incluye software de demostración tanto para Pocket PC 2002 y 2003, los cuales están desarrollados con el compilador de Embedded Visual Basic 4.0, un conjunto de funciones empaquetados en un archivo dll, llamado "CFReader.dll" y una clase "CCFReader.cpp" que sirve como medio de acceso al archivo CFReader.dll, todo este conjunto de archivos sirve para dos tareas: la primera para poder acceder y manipular el lector, y la segunda para poder programar el lector. Además, también incluye los controladores del lector para las distintas versiones de Windows tanto para sistemas operativos existentes en computadoras de escritorio como para Pocket Pc (Windows CE 2002 y 2003).

Con respecto a la PDA este lector se introduce en la ranura del Compact Flash Tipo 2 y para poder utilizarlo en una computadora de escritorio se necesita un adaptador a Compact Flash Tipo 2 a PCMCIA. Este utiliza el archivo CFReader.dll para su funcionamiento, en este archivo están representados los comandos del lector en forma de funciones, las cuales utiliza para poder establecer la comunicación computadora-lector y lector-etiqueta. Estos comandos se envían por medio del puerto serie ó también denominado puerto COM de la computadora. Estas funciones contienen los comandos que son recibidos por el lector e interpretados como instrucciones. Después de esto el lector ejecuta la instrucción. Esta comunicación

entre el lector y la etiqueta se puede comprobar utilizando la aplicación llamada *Hyperterminal* siempre y cuando se tenga conectado el lector al puerto serie de una computadora de escritorio, ya que si se utiliza en la PDA, ésta trae una Terminal parecida a la Hyperterminal solo para probar los comandos mínimos del lector RFID.

III.1.2 Etiquetas Tag-it ISO 15693 de Texas Instruments

Al tratar de realizar la compra de estas etiquetas con el proveedor, con el cual ya había contacto, surgió un problema con la cantidad de etiquetas pedidas. Esto es, el proveedor tiene establecida una cantidad mínima de venta, la cual no cumplíamos debido al presupuesto del proyecto. Se contacto a una persona de otra institución universitaria ubicada geográficamente cerca del local del proveedor para que realizara la compra de las etiquetas RFID. Después de resolver los problemas de como comprar las etiquetas, finalmente se logró realizar la compra de las etiquetas RFID necesarias para poder realizar la herramienta.

Las razones por las que se optó por utilizar etiquetas RFID *Tag-it ISO 15693 de Texas Instruments* son muy sencillas de explicar, la primera razón y la más fuerte es que está en la lista de los transponders compatibles con el Lector RFID Handheld con Compact Flash Tipo 2, y por ende este tipo de etiquetas es uno de los modelos compatibles con la aplicación RFDUMP. Otras de las razones y a su vez ventajas es que este tipo de etiquetas RFID ya está muy desarrollada y también está estandarizada. Además de que es de costo bajo. Y por último este modelo de etiqueta RFID es el que soporta más funcionalidades del lector RFID adquirido.

III.2 Análisis del Software Propietario

Durante la fase de análisis y exploración realizada al software que incluye el lector, surgió un problema al tiempo de cargar la biblioteca dinámica CFReader.dll con el compilador de *Embedded Visual C++ 4.0*. Básicamente el problema era que Windows no aceptaba dicho archivo dll. Para resolver este inconveniente se contactó al equipo de soporte de la empresa *Syscan* para explicarles el problema y recibir una solución al mismo, sin embargo la solución propuesta era sólo remplazar dicho archivo, fundamentándose en que el archivo CFReader.dll estaba corrupto o dañado, tal acción no resolvió el problema.

Se buscaron alternativas más viables, como el tratar de registrar el archivo dll con el servidor de registro de archivos dll's de Windows llamado *regsvr32*, por la parte de versiones de Windows para computadoras de escritorio y con el *regsvrce* para la parte de la PDA, en ambos casos los resultados no fueron favorables, ya que no reconocieron dicho archivo y lo denotaron como una imagen inválida de Windows y Windows CE.

Como ya se mencionó que se utilizó el compilador de Embedded Visual C++ 4.0, dicho compilador se descarga gratuitamente de la página de Microsoft. Este software incluye el emulador estándar y el SDK de Windows CE, pero lo que no incluye es el emulador para Pocket PC 2002 ni sus librerías SDK, estas dos cosas son las necesarias para poder ejecutar y probar las aplicaciones que se realicen en el compilador antes de instalarlas en la PDA's. Estas herramientas pueden bajarse de

manera gratuita del sitio de Microsoft Internet. Al momento de utilizar el emulador de Pocket PC 2002 –previa instalación y configuración- con el compilador de embedded Visual C++ 4.0 se obtuvo que el emulador de Pocket PC 2002 no se puede ejecutar con la versión de este compilador. Lo que ocasionó que se buscarán nuevas alternativas de solución.

Se buscó una versión de dicho compilador en el cual funcionara el emulador de Pocket PC 2002 y por ende que funcionara en la PDA. Después de una exhaustiva búsqueda en Internet se encontró que la versión de 3.0 del compilador embedded Visual C++ ya incluye tanto el emulador para Pocket PC 2002 como su respectivo SDK, rápidamente se realizó una aplicación de prueba –sin incluir el archivo CFReader.dll- para una PDA y con gran acierto funcionó tanto en el emulador como en el dispositivo real. Después se trató de cargar el archivo .dll pero no funcionó.

Se probó la aplicación de demostración para Pocket PC 2002 sobre una IPAQ H3900. Al realizar esto, se observó que al utilizar el archivo CFReader.dll para desarrollar aplicaciones para el lector RFID, estas aplicaciones serían dependientes del archivo .dll, dando pie a una portabilidad limitada. Después de este análisis sobre el archivo CFReader.dll del lector, se determinó no utilizarse y buscar otro lenguaje como alternativa o bien una aplicación que sirviera de marco en la búsqueda de dicho lenguaje que ofreciera la portabilidad de nuestra herramienta.

III.3 Análisis de la etiqueta RFID

Las etiquetas vienen pregrabadas con un número de identificación de *Solo-Lectura* llamado EPC (*Electronic Product Code*). Este número es una dirección única a nivel mundial, la cual es programada en la fábrica que expide la etiqueta. El EPC es de 32 bits de longitud ($= 2^{32}$ direcciones diferentes) [Web 9, op. Cit.]. También cuentan con 256 bits de memoria no-volátil, arregladas en 8 bloques de 32 bits (ver Figura 6). Los bloques pueden ser programados por el usuario en forma temporal o también pueden ser clausurados de manera permanente para proteger los datos de modificaciones posteriores. Una vez activado, el *bit* de bloqueo no puede ser restaurado [Sergio S., 2004; Art 3]. Los bloques de memoria pueden ser programados en forma independiente. Existen dos niveles de clausurado de bloque que son soportados: *Clausurado individual del bloque por el usuario o clausurado individual del bloque de datos programados en fábrica durante la manufactura.*

Cada bloque de la memoria del usuario de la etiqueta contiene 15 páginas las cuales están numeradas con números hexadecimales, estos números están en el rango de 0 hasta F. Cada página puede albergar hasta 4 caracteres de información, por ello si la cadena de datos es muy larga, esta se tiene que dividir en cadenas de 4 caracteres. Además, para poder acceder a una página de un bloque determinado ya sea para lectura o escritura se tiene que cumplir con un cierto protocolo.

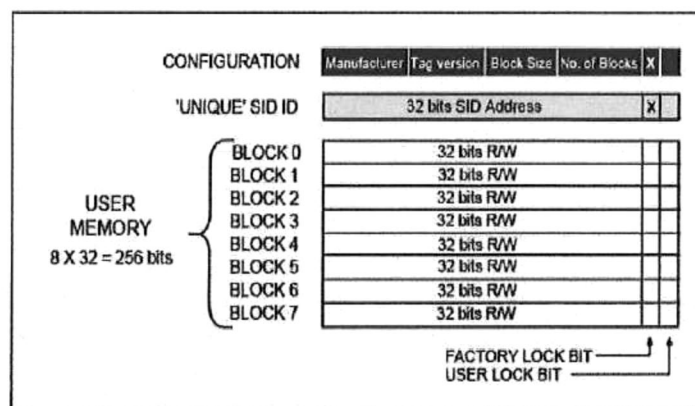


Figura 6 Organización de la Memoria de una etiqueta HF

III.3.1 Protocolos de los comandos de lectura y escritura

Se analizaron de los protocolos de los comandos utilizando etiquetas RFID *Tag-it ISO 15693 de Texas Instruments*. El protocolo para leer el contenido de una página es el siguiente:

Comando + número de bloque + número de página

Por ejemplo; si se quiere leer la información de la página 1 que está dentro del bloque 0, entonces el comando quedaría así:

r01

Donde la "r", es el comando de lectura y 01 es la dirección de la página

Como se puede apreciar en la Tabla 3 se están utilizando los parámetros "command code" y "parameters", para enviar la petición de datos del lector a la etiqueta.

El protocolo del comando de escritura es el siguiente:

Comando + dirección de la página + dato (formato hexadecimal)

Por ejemplo: si se quisiera escribir la palabra “Test” en la página 1 del bloque 0, el comando quedaría así:

w0154657374

Donde con “w” se le dice al lector que escriba la información “54657374” en la dirección “01” de la memoria de una etiqueta. El valor 54 65 73 74 es la palabra ‘Test’ en formato hexadecimal.

En ambos casos la etiqueta responde con la información pedida utilizando la estructura del comando de respuesta que se muestra en la Tabla 4.

III.3.2 Pruebas de concepto en lenguaje Java

Respecto a la lectura de las páginas de los bloques que se encuentran en la memoria de la etiqueta se muestra el siguiente código:

```
String leer_Memoria(String errMsg,String select){
//se selecciona la etiqueta RFID de la cual se leerá la información
    select = rfid.select();

    int adr=0;
    String datos="";
    while (errMsg == null) {

        String hexAdr = Integer.toHexString(adr);

        // Solo dos dígitos por número de página son posibles
        if (hexAdr.equals("100"))
            break;

        //si la dirección es un solo digito, se le agrega un 0 nada más
        if (hexAdr.length() == 1)
            hexAdr = "0" + hexAdr;

        //se da un tiempo de 5 milisegundos aproximadamente
        try {Thread.sleep(500);} catch (Exception e) {}

        // Se llama a la función readPage que esta en la clase RFID
        // Para que lea la página indicada por la dirección hexadecimal
        String page = rfid.readPage(hexAdr);
```

```

//se checa si no hay error al tatrar de leer la página
// de la memoria
errMsg = rfid.getErrorMsg(page);

if (errMsg != null) {

    // Se imprime en consola el error que ocurrio
    System.out.println(hexAdr + ":\t" + errMsg + " "+page);
    errMsg=null;

}

}else {
/* Sino ocurre un error al leer, entonces se desplegarán los datos
* en su forma de caracteres, ya que los datos almacenados en la
* etiqueta se encuentran en su valor hexadecimal */

    //codigo

    if(!page.equals("00000000"))
        datos += page;

    }//else
    // Se incrementa el número de la dirección de la página
    adr++;
}

//por último se regresan los datos en forma de caracteres ASCII
return fe.imprime_Datos(datos);
}

//función

```

El siguiente código muestra la estructura de la función *readPage*, la cual sirve para construir el protocolo del comando y así poder enviarle al lector RFID una instrucción válida. La clase RFID instancia a la clase *SerialConnection* para lograr sus fines.

```

public String readPage(String pageNo) {

    // El número de la página necesita estar en representación
    // hexadecimal. Además, se construye el comando cumpliendo con el
    // protocolo antes mencionado
    sc.write(CMD_READ_PAGE + pageNo);

    // Se lee la línea de información que entra por el puerto.
    String page = sc.readLine();

    // Si la página es nulo pues se lee la siguiente línea

```

```

        if (page == null)
            page = sc.readLine();
        //si la página es nulo también se lee la siguiente línea
        else if (page.equals(""))
            page = sc.readLine();

        return page;
    }

```

Respecto a la escritura de datos en las páginas de los bloques que se encuentran en la memoria de la etiqueta se muestra el siguiente código:

```

public boolean escribir_Tag(String datos){

    //se selecciona la etiqueta RFID donde se escribirá la información
    String select = rfid.select();

    //se verifica si no hay error al seleccionar la etiqueta
    String errMsg = rfid.getErrMsg(select);

    //si existe error se imprime en consola y la función se rompe
    if(errMsg != null){
        System.out.println("Error "+ errMsg);
        return false;
    }else {

        //sino entonces se mandan escribir la información
        if(this.escribir_Datos(datos)==1)
            return true;
        }
        return false;
    }
}

```

A continuación se muestra el código de la función *“writePage”*:

```

public String writePage(String pageNo, String page) {
    // El número de la página necesita estar en representación
    // hexadecimal. Además, se construye el comando cumpliendo con el
    // protocolo antes mencionado
    sc.write(CMD_WRITE_PAGE + pageNo + page);

    // Se lee la línea de información que entra por el puerto.
    String respuesta = sc.readLine();

    // Si la página es nulo pues se lee la siguiente línea
    if (respuesta == null)
        respuesta = sc.readLine();
}

```



```
//si la página es nulo también se lee la siguiente línea
else if (respuesta.equals(""))
    respuesta = sc.readLine();

return respuesta;
}
```

III.3.3 Comunicación entre Lector y Etiqueta

La comunicación entre el lector y cualquier tipo de etiqueta es en base a comandos. (Ver Tabla 3 Estructura del Marco de la Petición del Transponder) El lector emite una petición al *transponder* en la forma de instrucciones y parámetros cifrados [Web 7]. (Ver Tabla 4) El *transponder* responde con un bloque que contiene el estatus de la información y los datos como respuesta. A continuación se muestran las estructuras de los comandos.

Request Frame	Bit Length	Bit Position
Request Code	2	0-1
Command Code	8	2-9
Format Type	1	10
Address Flag	1	11
Reserved	1	12
SID Address (optional)	32	13-44
Parameters	256 bits max	45-...
CRC	16	depends on previous fields

Tabla 3 Estructura del Marco de la Petición del Transponder

Response Frame	Bit Length	Bit Position
Response Code	2	0-1
Command Code	8	2-9
Format Type	1	10
Address Flag	1	11
Reserved	1	12
Error Flag	1	13
SID Address (optional)	32	14-45
Error Code (optional)	8	46-53
Data	256 bits max	54
CRC	16	depends on previous fields

Tabla 4 Estructura del Marco de la Respuesta del Transponder

III.4 Análisis de la Herramienta RFDUMP

Esta aplicación utiliza un Lector RFID ACG con Compact Flash Tipo 2 y ciertos tipos de *transponders* para su funcionamiento. Esta aplicación puede ser ejecutada en distintos sistemas operativos; por ejemplo: Windows y Linux [Lukas, Boris; 2002].

A pesar de esto, quedan dos dudas de su portabilidad en Windows CE, ya que esta aplicación no está probada en este sistema operativo. Una de ellas tiene que ver con la siguiente pregunta: ¿Existirá una máquina virtual de Java para PDA's? y la otra con la siguiente interrogante: ¿Existirá un paquete de javax.comm portable para PDA's? La primera se resuelve con la búsqueda e instalación de varias máquinas virtuales para PDA's, estas acciones dieron como resultado Jeode EVM y la segunda se resolvió con el hallazgo de los paquetes: *CESerial.jar*, *comm.jar* y *commapi.dll*. Tras las pruebas realizadas a la máquina virtual Jeode EVM con los paquetes *.jar* – previa configuración e instalación en una PDA (IPAQ H3900)- que se mencionaron con anterioridad se observó que esta máquina sería la adecuada para nuestros propósitos. A pesar de que este tipo de PDA no se encontraba dentro la lista de los modelos de PDA's en los que el desarrollador probó dichos paquetes, estos funcionaron perfectamente.

Al analizar las características que tenía y a las ventajas que ofrecía esta aplicación, se realizó un análisis exhaustivo del código para observar si sería útil el lenguaje de programación Java que se utiliza. Este análisis determinó que el lector

podía ser programado directamente por medio del puerto serie o COM, esta característica del lector abrió un amplio panorama de como manipular la comunicación computadora-lector con gran facilidad. Dentro de las características del lenguaje de programación Java, se puede enfatizar una muy fuerte e importante, que servirá como base para poder realizar los objetivos del presente trabajo de tesis, y esta es, que provee una interfaz llamada *comm.jar* para realizar llamadas al puerto serie. Esta interfaz es importante ya que la mayoría los lectores RFID usan la interfaz RS232 o puerto serie para comunicarse con la computadora (ver Apéndice A). Por lo tanto, en base a las características del lenguaje y a la facilidad de manipular la comunicación –previamente se probó la programación- del puerto serie se optó por utilizar el lenguaje de programación con el cual fue desarrollada la herramienta RFDUMP.

IV Desarrollo de Clases Base de Java para RFID

El propósito del análisis es definir los requerimientos necesarios para la realización de la herramienta a desarrollar. Tomando como punto principal la abstracción que debe concebir para facilitar a los desarrolladores una programación de alto nivel y despreocuparlos de una programación de bajo nivel.

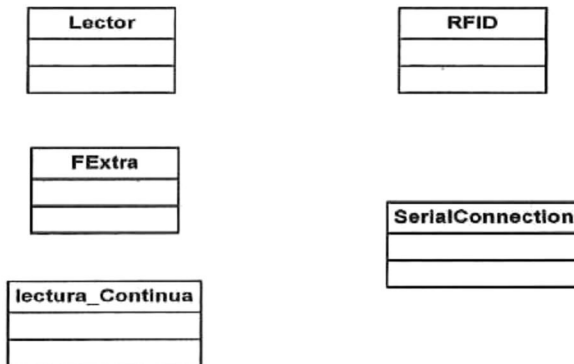
Esta herramienta no abarcará la implementación de todos los comandos (ver Criterios de Diseño) para poder manipular tanto a los lectores RFID como a las etiquetas RFID, ya que el fin es proveer una amplia generalización con respecto a la utilización de los comandos mínimos que tiene un lector RFID con una etiqueta RFID, y a la comunicación entre la computadora y el lector RFID.

Por ello, tiene que poseer un conjunto de clases o procedimientos bases, los cuales sean suficientes para poder manipular con gran facilidad los dispositivos (lectores y etiquetas RFID) de esta tecnología. Además, no debe estar limitada por una plataforma determinada, dando así, portabilidad a las aplicaciones que se realicen con esta misma.

Este conjunto de clases bases RFID servirán como interfaz entre la aplicación del usuario y el lector RFID (hardware) ya que con este conjunto de clases el usuario podrá manipular al lector RFID para efectuar operaciones sobre la etiqueta RFID.

IV.1 Análisis de la Arquitectura del sistema

IV.1.1 Abstracciones Clave



Definiciones de las Abstracciones Clave

Lector: Contiene los métodos del lector RFID en una abstracción alta de programación.

RFID: Contiene los comandos que para manipular al lector RFID en una abstracción baja de programación.

FExtra: Contiene los métodos para la conversión de datos tanto de carácter - decimal, decimal - hexadecimal, hexadecimal- decimal y decimal - carácter.

SerialConnection: Contiene los métodos para abrir y cerrar el puerto serie de la computadora.

Lectura_Continua: Es la ejecución de un ciclo infinito para leer continuamente el identificador de la etiqueta.

IV.2 Análisis

IV.2.1 Requerimientos

REQ1. Apagar el Lector RFID

REQ2. Prender el Lector RFID

REQ3. Leer el identificador de la etiqueta

REQ4. Leer de forma continua los identificadores de las etiquetas presentes en el área de lectura de la antena

REQ5. Leer los datos de la memoria de la etiqueta

REQ6. Escribir datos en la memoria de la etiqueta

REQ7. Obtener la versión del Lector RFID

REQ8. Detener la lectura continua de los identificadores

REQ9. Configurar el puerto serie donde esta conectado el lector RFID

IV.2.2 Casos de Uso Generales

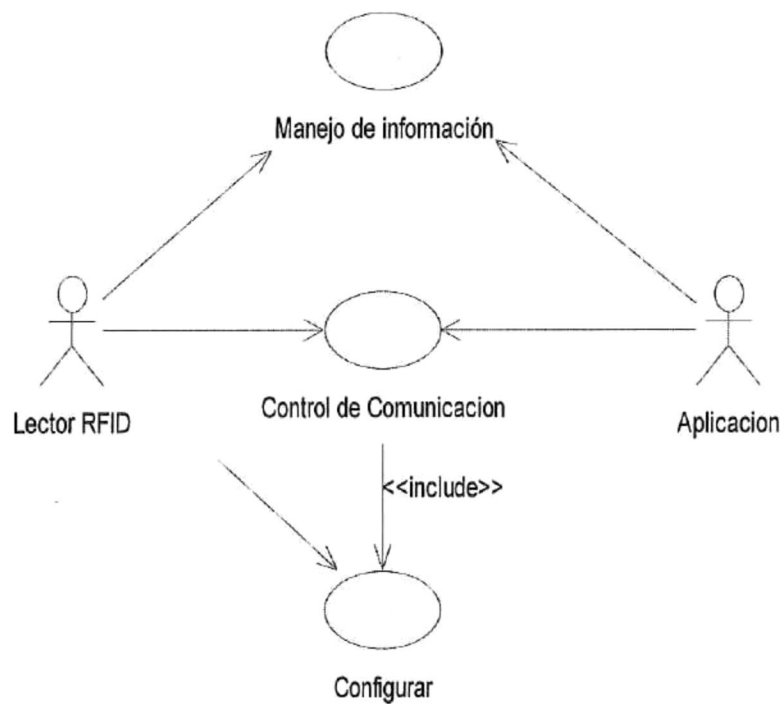


Figura 7 Casos de Uso Generales

IV.2.1.1 Caso de Uso Manejo de Información

Caso de uso: Manejo de información

Actores: Aplicación, Lector

Propósito: Envío y recepción de la información

Resumen: Es el control del envío (escritura) y recepción (lectura) de la información entre la aplicación y el lector RFID. La lectura de información puede darse de manera simple o continua.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. La aplicación envía la petición de	El sistema realiza las actividades

lectura (continua o simple) al lector RFID.	correspondientes para enviar el mensaje al lector RFID y regresa los datos pedidos.
2. La aplicación envía la petición de escritura datos a alguna etiqueta.	El sistema realiza las actividades correspondientes para enviar el mensaje al lector RFID y este último a su vez efectúa las operaciones de escritura pedidas por el sistema.
3. La aplicación envía la petición de detener la lectura.	El sistema realiza las actividades correspondientes para enviar el mensaje al lector RFID y este último a su vez efectúa la operación de apagar la antena lectora pedida por el sistema.
4. La aplicación envía la petición de obtener la versión del lector RFID	El sistema realiza las actividades correspondientes para enviar el mensaje al lector RFID y este último responde con su versión.

Cursos alternativos

1.

- El lector RFID no puede leer las etiquetas o transponders presentes porque no son compatibles con el lector RFID.
- El lector RFID no puede leer las etiquetas o transponders porque no están dentro del área de lectura del lector o antena lectora.

2.

- El lector RFID no puede escribir los datos en la etiqueta o transponder presente porque no son compatibles con el modo de escritura.
- El lector RFID no puede escribir los datos en la etiqueta o transponder porque no esta presente la etiqueta a la que se desea escribir los datos.

1, 2, 3 y 4. Que el lector RFID no esté presente y por tal motivo no se pueda acceder a él.

IV.2.1.2 Caso de Uso Control de Comunicación

Caso de uso: Control de Comunicación

Actores: Aplicación, Lector

Propósito: Controlar la comunicación entre el lector y la aplicación

Resumen: Establecer la comunicación entre la aplicación y el lector RFID, es decir, prender y apagar la comunicación con el dispositivo lector.

Dependencia: Para poder realizar sus actividades, el control de comunicación, depende del caso de uso “Configurar”, ya que en dicho caso de uso se especifican los parámetros que se utilizan para abrir la comunicación con el lector RFID

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. La aplicación envía la orden Encender a el lector RFID para comenzar a realizar las actividades de lectura o escritura.	El sistema realiza las actividades correspondientes desde la detección del puerto que se utilizará para trabajar con el lector hasta abrir la comunicación con el lector RFID y dejarlo listo para utilizarse.
2 La aplicación envía la orden Apagar el lector RFID.	El sistema realiza las actividades correspondientes para enviar el mensaje al lector RFID. Cerrar la comunicación con el dispositivo lector.

Cursos alternativos

1. No se puede encender el lector porque:
 - No está conectado en el puerto serie de la computadora o PDA
 - En caso de estar usando una Pocket PC. No está configurado el lector RFID en el puerto especificado.
 - El puerto configurado no está disponible en ese momento

- El puerto especificado en el archivo de propiedades no existe
- No existen los paquetes que facilitan la comunicación por el puerto serie

2. No se puede apagar el lector porque no está conectado al puerto serie de la computadora, laptop o PDA o simplemente nunca se encendió.

IV.2.1.3 Caso de Uso Configurar

Caso de uso: Configurar

Actores: Usuario final, Lector

Propósito: Configurar la tasa de bits por segundo, bits de datos, paridad, bits de parada y puerto.

Resumen: Configurar el puerto serie de comunicación del lector-computadora, configurar la tasa de transmisión de datos entre computadora-lector, esta tasa también es denominada la tasa de bits por segundo o baudios. Esta tasa es diferente en cada lector. Además, configurar los bits de datos, la paridad y los bits de parada.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
1. El usuario final abre el archivo de propiedades y modifica los parámetros del lector, después de esta acción guarda los cambios.	
2. La aplicación lee los parámetros de configuración de un archivo de propiedades y la establece como la nueva configuración del lector RFID para comenzar a realizar las actividades de lectura o escritura.	El sistema realiza las actividades correspondientes estableciendo los nuevos parámetros con los que se trabajará con el lector RFID y dejarlo listo para utilizarse.

Cursos alternativos

1.

- El usuario no puede modificar los parámetros porque el archivo está en modo de solo-lectura.
- No se puede abrir el archivo para modificar los parámetros porque está corrupto o dañado

2.

- No se puede establecer la nueva configuración porque el lector no soporta algunos de los nuevos parámetros o el lector no está presente.
- No se puede establecer la nueva configuración porque no se puede leer el archivo de configuración.
- No se puede establecer la nueva configuración porque existen problemas con el puerto serie.

IV.2.3 Casos de Uso Específicos

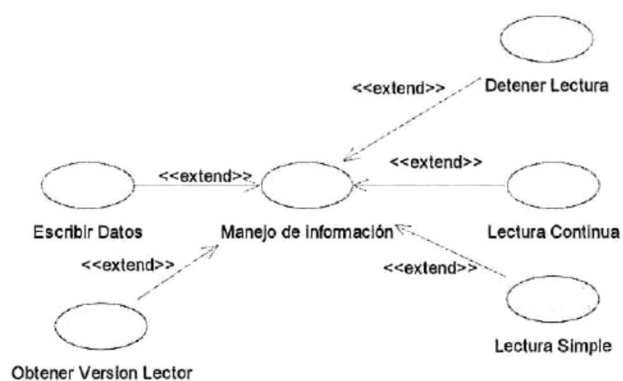


Figura 8 Caso de uso: Manejo de Información

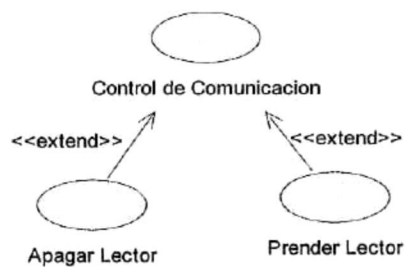


Figura 9 Caso de uso: Control de Comunicación

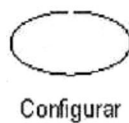


Figura 10 Caso de uso: Configurar

IV.2.2.1 Caso de uso Apagar Lector

Caso de uso: Apagar Lector

Actores: Aplicación, Lector

Propósito: Apagar lector RFID

Resumen: Terminar toda actividad y comunicación con el lector RFID, además de cerrar el puerto serie de la computadora el cual se usaba para comunicarse con el lector.

Precondición: El lector RFID tiene que estar conectado en el puerto serie

Curso normal de los eventos

1. La aplicación le envía al lector RFID la instrucción de apagarse
2. El lector recibe y procesa la instrucción, si la instrucción es válida el lector la ejecuta
3. La aplicación cierra el puerto serie de comunicación con el lector
4. Si se pudo cerrar el puerto serie, la aplicación recibirá un valor "true" como éxito

Alternativas:

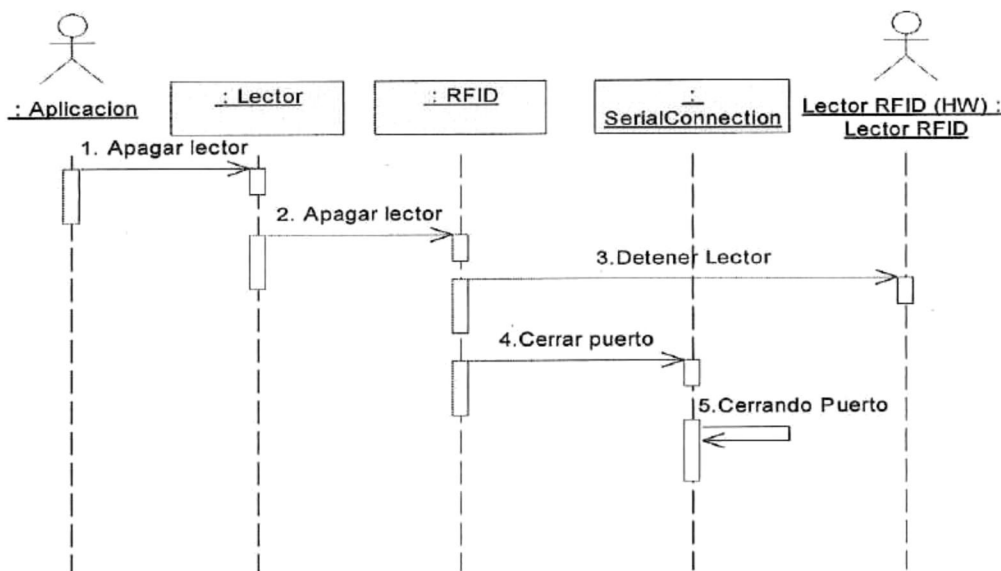
1. La aplicación no puede enviar la instrucción al lector porque no está abierto el puerto serie aún.
2. El lector no recibe la instrucción porque no se ha establecido la comunicación computadora-lector, si la instrucción es inválida el lector devuelve un mensaje de error
3. La aplicación no puede cerrar el puerto serie de comunicación porque la computadora tiene problemas con el puerto.

4. Si no se pudo cerrar el puerto serie, la aplicación recibirá un valor "*false*" como fracaso.

Postcondición: El lector RFID está apagado y el puerto serie está cerrado

Diagrama de Secuencia

Nombre: Apagar Lector



1. La aplicación le envía el mensaje "Apagar lector" a la clase Lector
2. La clase Lector le comunica a la clase RFID que apague el hardware lector RFID
3. La clase RFID manipula al lector ordenándole que se detenga.
4. Después la clase Lector le dice a la clase RFID que cierre la comunicación con el lector.
5. La clase RFID le avisa a la clase SerialConnection que cierre el puerto serie que se está usando en la comunicación computadora-lector.

Diagrama de Colaboración

Nombre: Apagar Lector

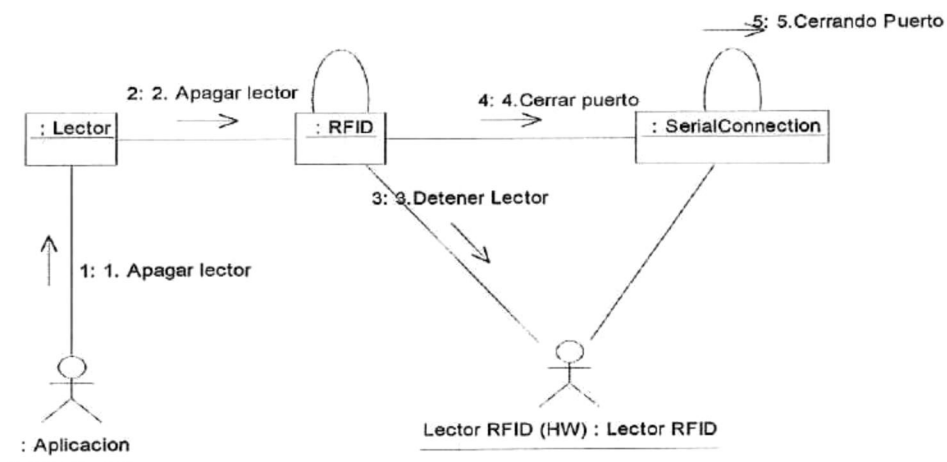
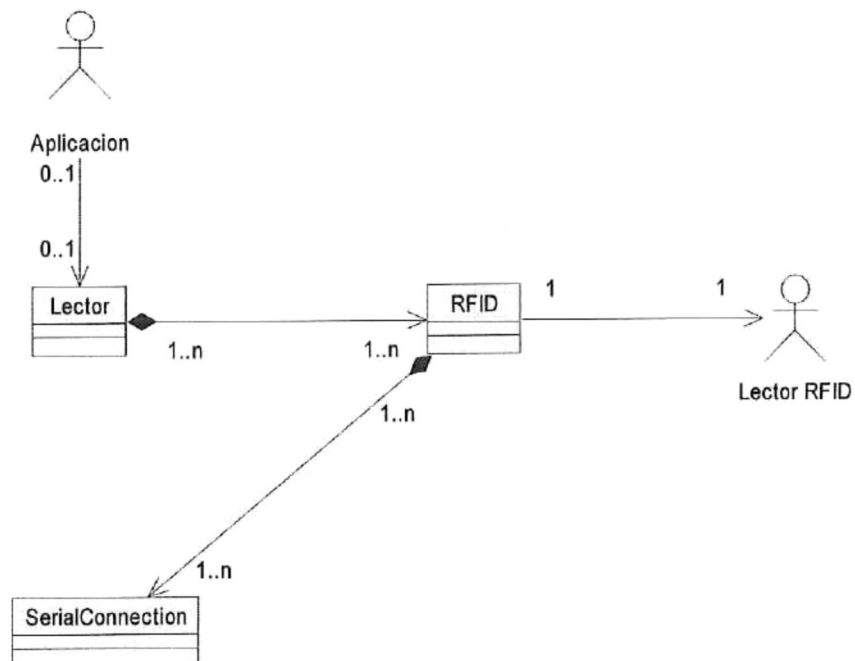


Diagrama de Clases

Nombre: Apagar Lector



IV.2.2.2 Caso de uso Prender Lector

Caso de uso: Prender Lector

Actores: Aplicación, Lector

Propósito: Prender le lector RFID

Resumen: Realizar las operaciones necesarias para abrir la comunicación con el lector RFID, es decir, abrir y configurar el puerto serie de la computadora.

Precondición: El lector RFID tiene que estar conectado en el puerto serie de la computadora o PDA

Curso normal de los eventos

1. Cargar el número y configuración del puerto serie de la computadora que se encuentra en el archivo de propiedades
2. Abrir el puerto serie y establecer la configuración
3. Establecer la comunicación con el lector, si se pudo establecer la comunicación se regresa "*true*" como éxito sino "*false*" como fracaso.

Alternativas:

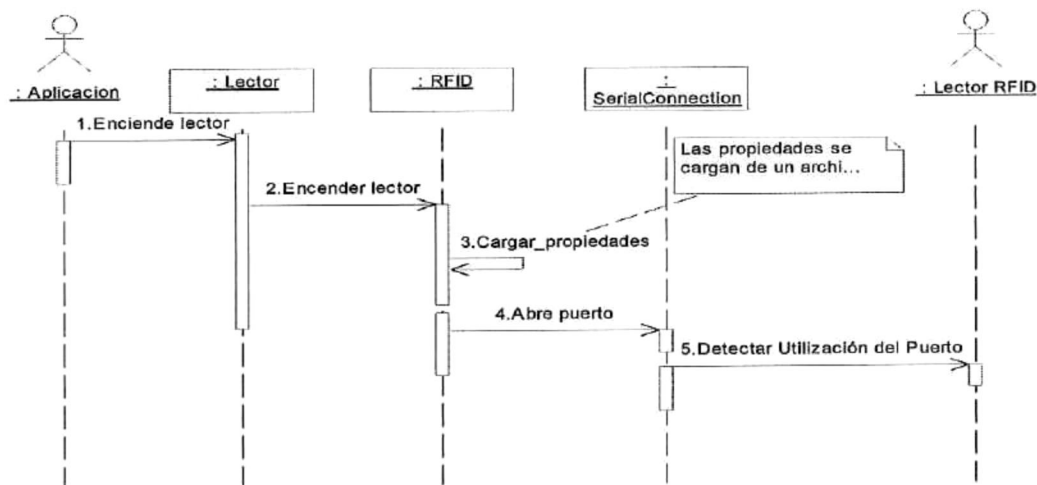
1. No se puede leer el archivo de propiedades por muchas razones como: no está en la ruta especificada, está corrupto, etc. y por ende no se puede cargar el número y configuración del puerto serie
2. No se puede abrir el puerto serie por razones como: no existe el número del puerto, está ocupada por otra aplicación, etc. y por consiguiente no se puede aplicar su configuración

3. Si no se puede establecer la comunicación con el lector entonces se devuelve *false* como fracaso, las razones pueden ser: porque no se pudo abrir el puerto o el lector no está conectado en ese puerto.

Postcondición: El lector RFID esta prendido y listo para usarse

Diagrama de secuencia

Nombre: Prender Lector



1. La aplicación le envía el mensaje “Enciende lector” a la clase Lector para empezar a realizar las actividades con el lector RFID.
2. La clase Lector le comunica a la clase RFID que encienda el hardware lector RFID.
3. La clase RFID carga las propiedades de configuración del lector, y las establece de inmediato.
4. La clase RFID le indica a la clase SerialConnection que abra el puerto serie.
5. Después la clase SerialConnection trata de detectar el puerto serie y lo abre y

configura con los parámetros que le fueron dados.

Diagrama de Colaboración

Nombre: Prender Lector

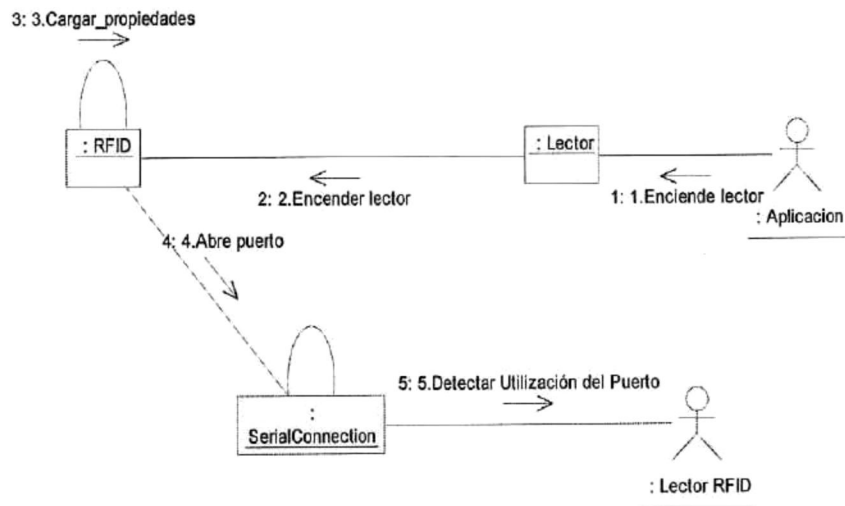
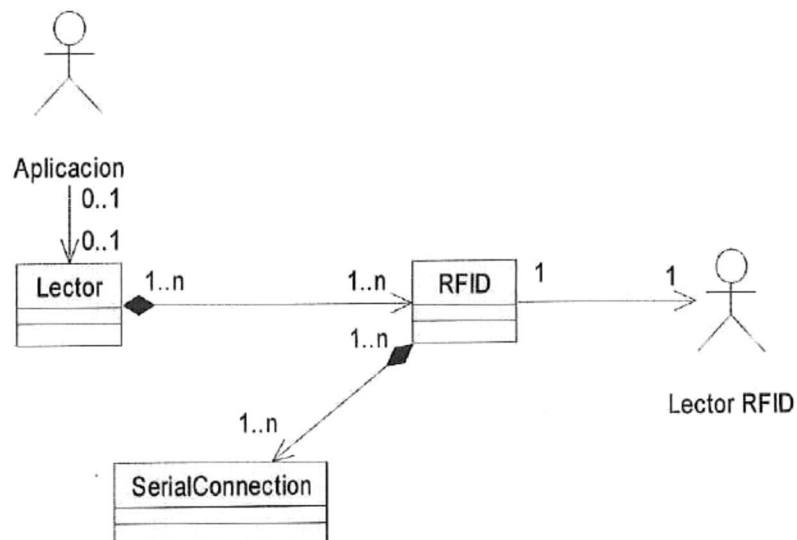


Diagrama de Clases

Nombre: Prender Lector



IV.2.2.3 Caso de uso Detener lectura

Caso de uso: Detener Lectura

Actores: Aplicación, Lector

Propósito: Detener la lectura de datos de las etiquetas

Resumen: El propósito es de apagar la antena lectora del lector RFID para detener la lectura continua de datos de las etiquetas que estén presentes en ese momento. Esta acción solo se usa cuando el lector está en modo de lectura continua.

Precondición: El lector RFID tiene que estar conectado en el puerto serie y prendido

Curso normal de los eventos

1. La aplicación le indica al lector RFID que ya no desea que lea más datos y por ende le ordena que apague su antena lectora.
2. El lector RFID recibe la instrucción
3. Si la instrucción es válida, el lector RFID la ejecuta y responde
4. La aplicación recibe el dato de éxito

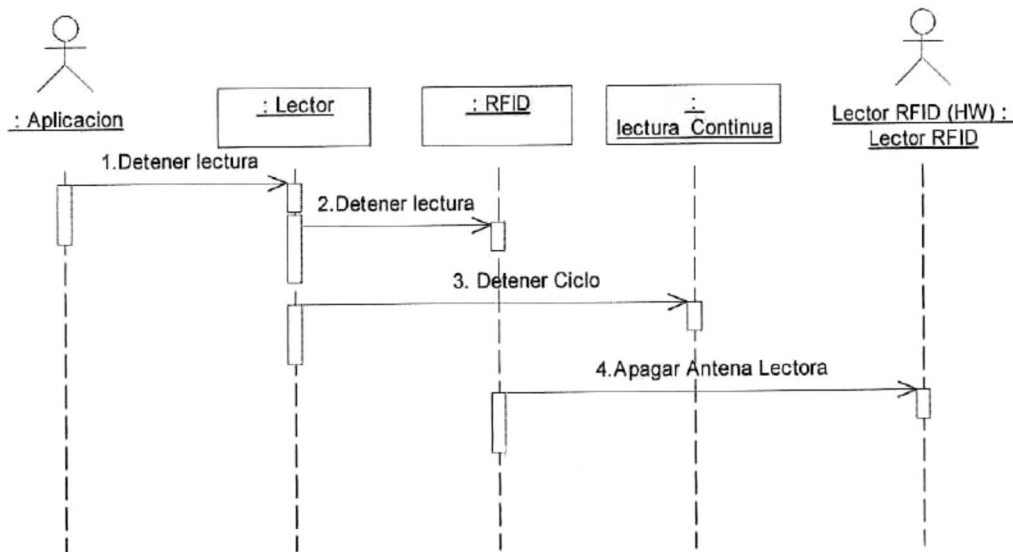
Alternativas:

1. La aplicación recibe un error diciendo que el lector no está encendido
2. El lector RFID no recibe la instrucción porque no está conectado en el puerto serie
3. La instrucción contiene un comando equivoco y por ello es inválida
4. La aplicación no recibe el dato porque existe un error ó mala lectura del puerto

Poscondición: El lector ha sido apagado

Diagrama de secuencia

Nombre: Detener Lectura



1. La aplicación le envía el mensaje de “Detener lectura” a la clase Lector para que detenga la lectura continua que se este realizando el lector RFID –hablando del hardware- en ese momento.
2. La clase Lector le comunica a la clase RFID que le ordene al lector RFID – hablando del hardware- que detenga la lectura continua.
3. La clase Lector le avisa a la clase lectura _ continua que detenga su ciclo continuo de petición de datos hacia el lector RFID –hablando del hardware-.
4. Después la clase RFID le ordena al lector RFID –hablando del hardware- que apague su antena lectora para que ya no reciba más datos de las etiquetas que están presentes en su área de lectura.

Diagrama de Colaboración

Nombre: Detener Lectura

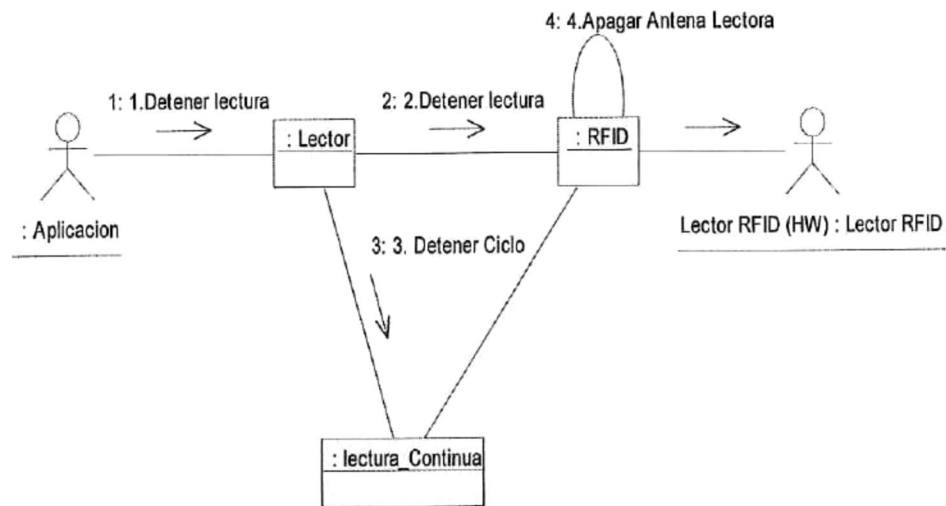
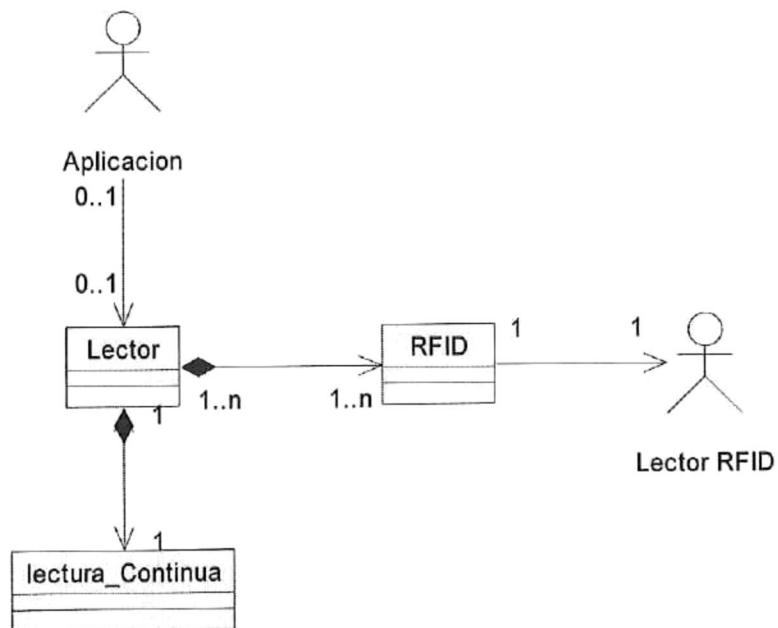


Diagrama de Clases

Nombre: Detener Lectura



IV.2.2.4 Caso de uso: Lectura Continua

Caso de uso: Lectura Continua

Actores: Aplicación, Lector

Propósito: Leer datos continuamente de las etiquetas RFID

Resumen: El propósito es de establecer al lector RFID en modo de lectura continua y así leer el identificador de todas las etiquetas que se estén presentando por el área de lectura de la antena lectora del dispositivo RFID.

Precondición: El lector RFID tiene que estar conectado en el puerto serie y prendido

Curso normal de los eventos

1. La aplicación le ordena al lector RFID que se ponga en modo de lectura continua
2. El lector RFID recibe la instrucción
3. Si la instrucción es válida, el lector RFID la ejecuta y devuelve el dato pedido
4. La aplicación recibe el dato y se queda en modo de espera por el siguiente dato

Alternativas:

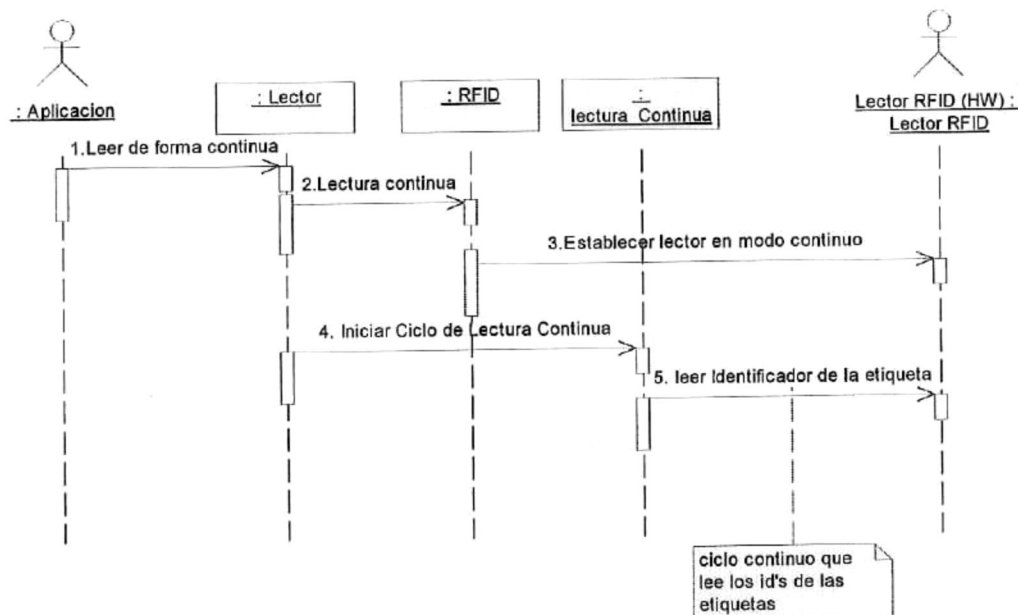
1. La aplicación recibe un error diciendo que no se puede comunicar con el lector
2. El lector RFID no recibe la instrucción porque no está conectado o hubo un problema en el puerto serie

3. La instrucción contiene un comando equívoco y por ende la instrucción es inválida
4. La aplicación recibe un dato erróneo debido a una mala lectura hecha por el lector

Poscondición: El lector ha sido establecido en modo de lectura continua

Diagrama de secuencia

Nombre: Lectura Continua



1. La aplicación le envía el mensaje de “leer de forma continua” a la clase Lector.
2. La clase Lector le comunica a la clase RFID que le ordene al lector RFID –hablando del hardware- que empiece la lectura continua.
3. La clase RFID le ordena al lector RFID –hablando del hardware- que se ponga en modo de lectura continua.
4. La clase Lector le avisa a la clase lectura _ Continua que inicie su ciclo continuo de petición de datos hacia el lector RFID –hablando del hardware-.
5. La clase lectura _ Continua en un ciclo infinito le pide al lector RFID que le dé el identificador de cada transponder que se encuentra en su área de lectura.

Diagrama de Colaboración

Nombre: Lectura Continua

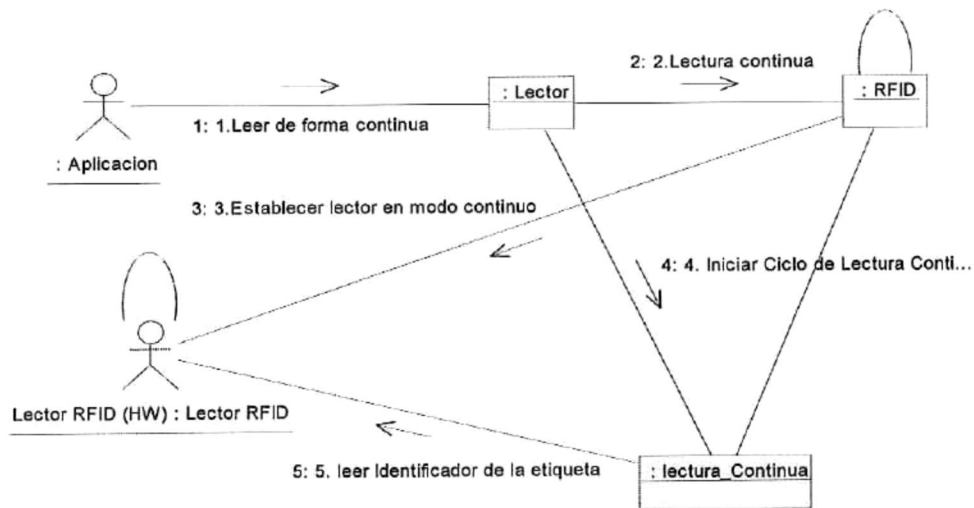
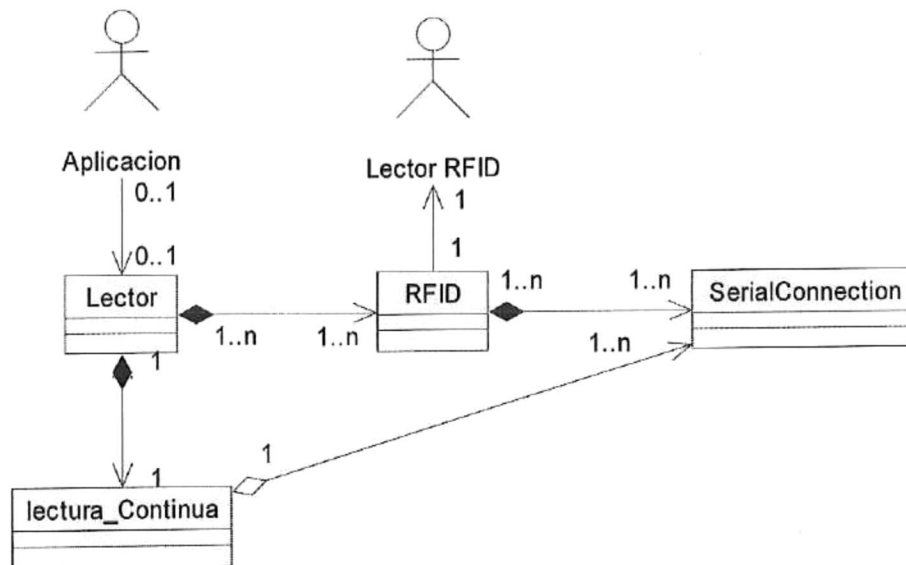


Diagrama de Clases

Nombre: Lectura Continua



IV.2.2.5 Caso de uso Lectura Simple

Caso de uso: Lectura Simple

Actores: Aplicación, Lector

Propósito: Leer datos de la etiqueta RFID

Resumen: El lector RFID tiene que leer los datos que están dentro de la memoria de la etiqueta RFID, así como su identificador, su tipo y el nombre de su manufacturadora.

Precondición: El lector RFID tiene que estar conectado en el puerto serie y prendido

Curso normal de los eventos

1. La aplicación le ordena al lector RFID que lea los datos de la memoria de la etiqueta RFID, su identificador, su tipo y el nombre de la manufacturadora.
2. El lector RFID recibe las instrucciones
3. Si la instrucciones son válidas, el lector RFID las ejecuta y devuelve los datos pedidos por la aplicación
4. La aplicación recibe los datos y los procesa

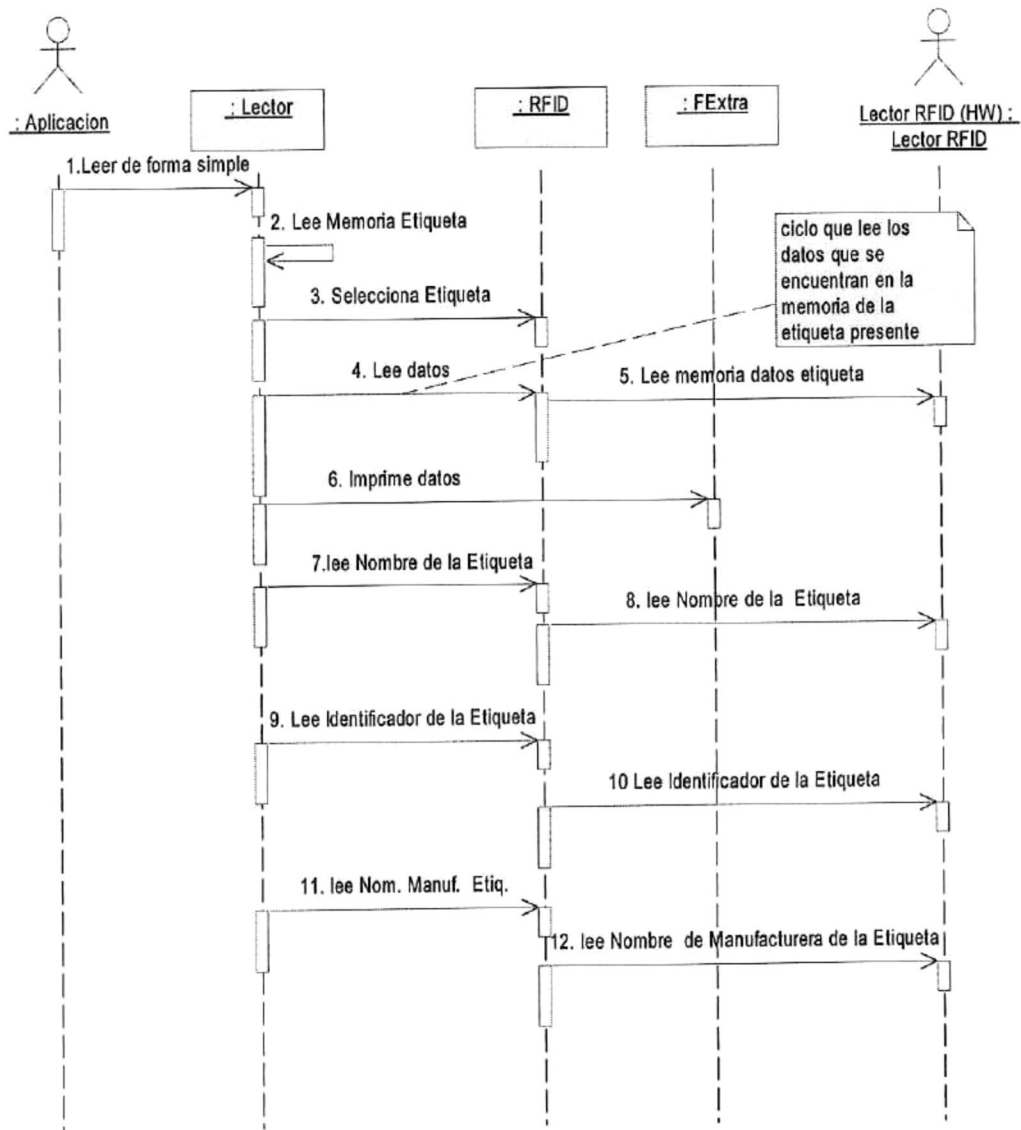
Alternativas:

1. La aplicación recibe un error, que no puede comunicar con el lector
2. El lector RFID no recibe la instrucción porque no está conectado o hubo un problema en el puerto serie
3. Si alguna instrucción es inválida el lector RFID arroja un error y se interrumpe la ejecución de las demás instrucciones
4. La aplicación recibe algún dato erróneo debido a una mala lectura hecha por el lector

Postcondición: El lector se queda esperando otra instrucción

Diagrama de secuencia

Nombre: Lectura Simple



1. La aplicación le envía el mensaje de “Leer de forma simple” a la clase Lector
2. La clase Lector llama a su función de lectura_simple, la cual con un ciclo lee la memoria de la etiqueta RFID que esté presente en el área de lectura del lector RFID.
3. La clase Lector le avisa a la clase RFID que le seleccione la etiqueta RFID que esté presente en el área de lectura.
4. La clase Lector le ordena a la clase RFID que lea los datos de la memoria de la etiqueta RFID.
5. La clase RFID le ordena al lector RFID que le envíe los datos que se encuentran en ciertas direcciones de memoria de usuario de la etiqueta.
6. La clase Lector le ordena a la clase FExtra que imprima los datos que está devolviendo la clase RFID.
7. La clase Lector le ordena a la clase RFID dar el nombre de la etiqueta
8. La clase RFID le ordena al lector RFID que le envíe el nombre de la etiqueta
9. La clase Lector le ordena a la clase RFID dar el identificador de la etiqueta RFID.
10. La clase RFID le ordena al lector RFID que le envíe el identificador de la etiqueta
11. La clase Lector le ordena a la clase RFID dar el nombre de la manufacturadora de la etiqueta
12. La clase RFID le ordena al lector RFID que le envíe el nombre de la manufacturadora de la etiqueta

Diagrama de Colaboración

Nombre: Lectura Simple

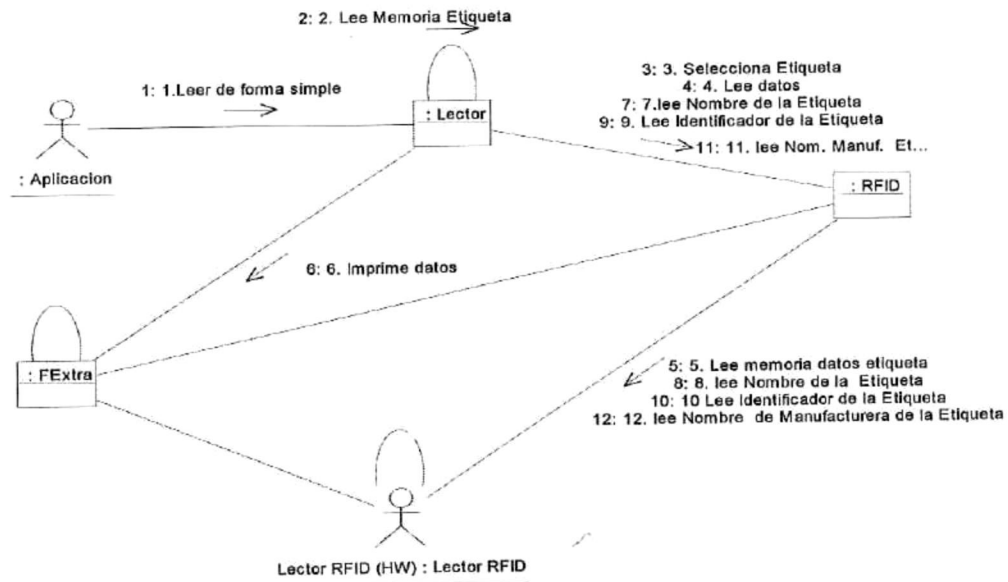
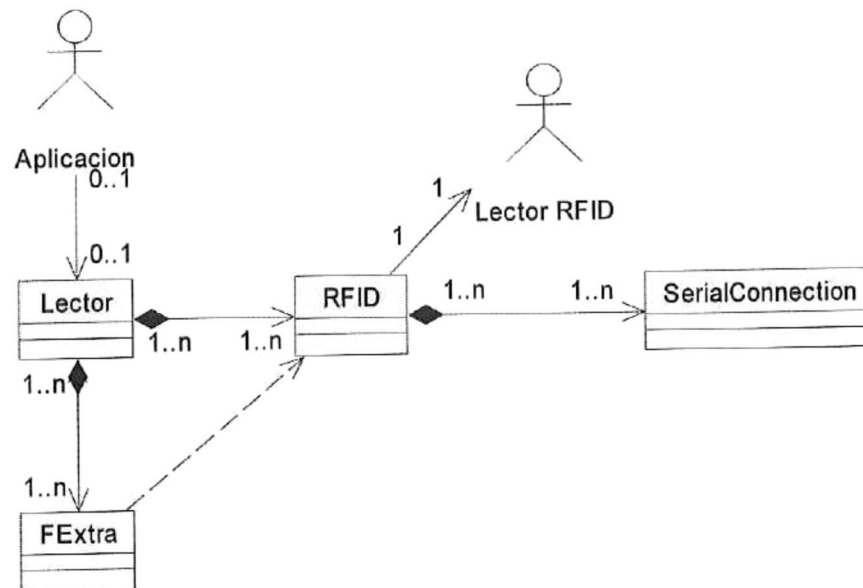


Diagrama de Clases

Nombre: Lectura Simple



IV.2.2.6 Caso de uso Escribir datos

Caso de uso: Escribir datos

Actores: Aplicación, Lector

Propósito: Escribir datos en la etiqueta RFID

Resumen: Escribir datos en la memoria de usuario de la etiqueta RFID, así como traducir los caracteres de los datos a el formato que maneja el lector y la etiqueta RFID

Precondición: El lector RFID tiene que estar conectado en el puerto serie y prendido.

La etiqueta tiene que estar en el área de lectura del lector RFID.

Curso normal de los eventos

1. La aplicación le ordena al lector RFID que escriba los datos en la memoria de la etiqueta RFID
2. El lector RFID recibe la instrucción y los datos
3. Si la instrucción es válida, el lector RFID la ejecuta y devuelve un mensaje de éxito.
4. La aplicación recibe los datos y los procesa

Alternativas:

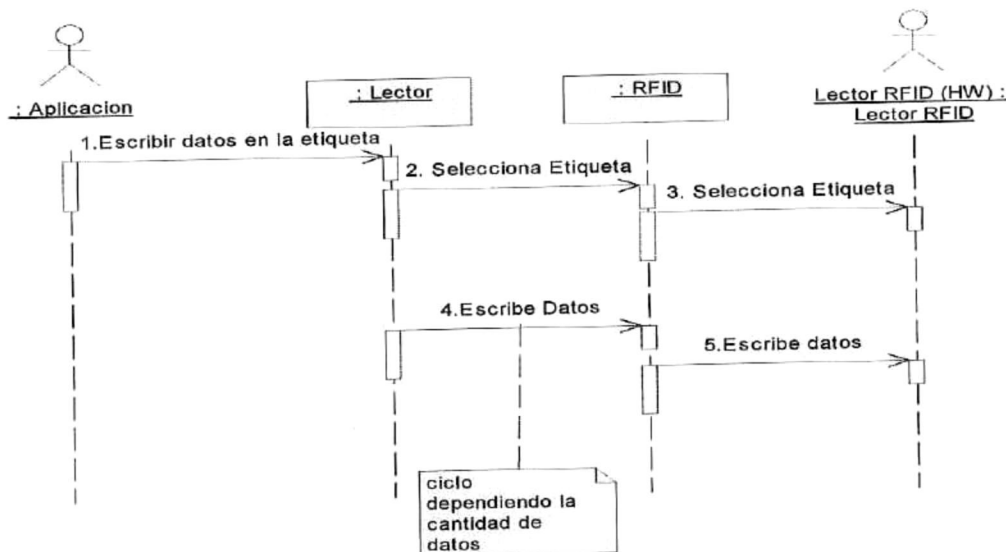
1. La aplicación recibe un error diciendo que no se puede comunicar con el lector
2. El lector RFID no recibe la instrucción porque no está conectado o hubo un problema en el puerto serie
3. Si alguna instrucción es inválida el lector RFID arroja un error y se interrumpe la ejecución de las demás instrucciones

4. La aplicación recibe algún dato erróneo debido a una mala lectura hecha por el lector

Postcondición: El lector se queda esperando otra instrucción

Diagrama de secuencia

Nombre: Escritura de Datos



2. La aplicación envía el mensaje de “Escribir datos en etiqueta” a la clase Lector.
3. La clase Lector le dice a la clase RFID que seleccione una etiqueta RFID para escribir en ella.
4. La clase RFID le ordena al lector RFID que seleccione la etiqueta RFID que está presente en el área de lectura.
5. La clase Lector le ordena a la clase RFID que escriba los datos que le envía en una etiqueta RFID.
6. La clase RFID le ordena al lector RFID que escriba los datos en la memoria de la etiqueta RFID

Diagrama de Colaboración

Nombre: Escritura de Datos

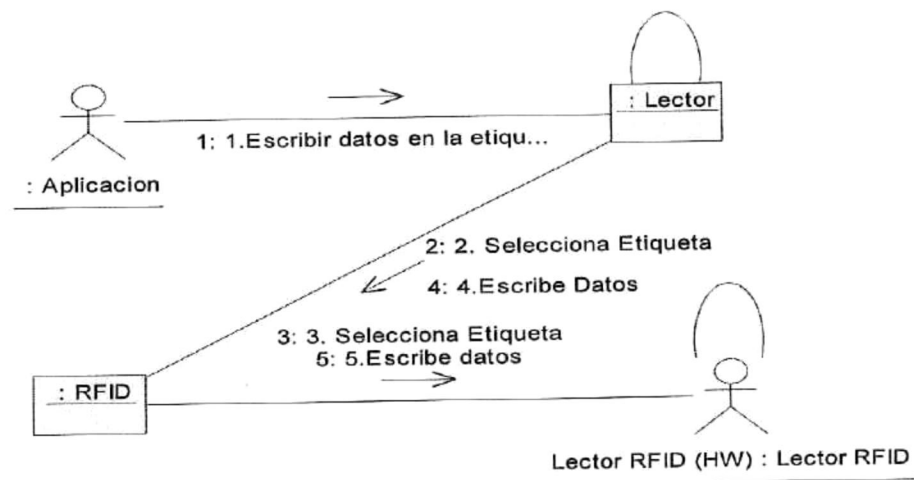
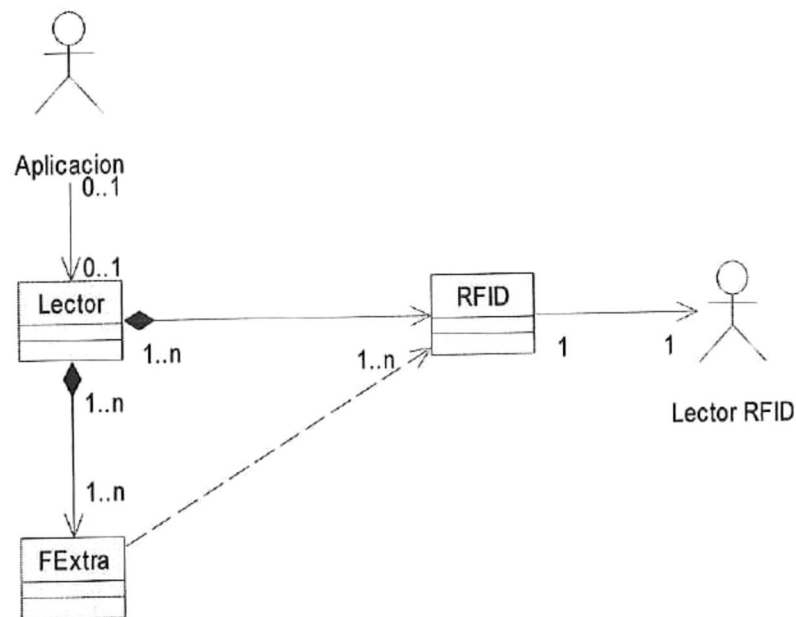


Diagrama de Clases

Nombre: Escritura de Datos



IV.2.2.7 Caso de uso Obtener Versión Lector

Caso de uso: Obtener Versión Lector

Actores: Aplicación, Lector

Propósito: Obtener la versión del lector RFID

Resumen: Obtener la versión del lector RFID para verificar si está prendido o funcionando

Precondición: El lector RFID tiene que estar conectado en el puerto serie y prendido

Curso normal de los eventos

1. La aplicación le ordena al lector RFID que verifique su versión y la regrese a la aplicación
2. El lector RFID recibe la instrucción
3. Si la instrucción es válida, el lector RFID la ejecuta y devuelve el dato pedido por la aplicación
4. La aplicación recibe los datos y los procesa

Alternativas:

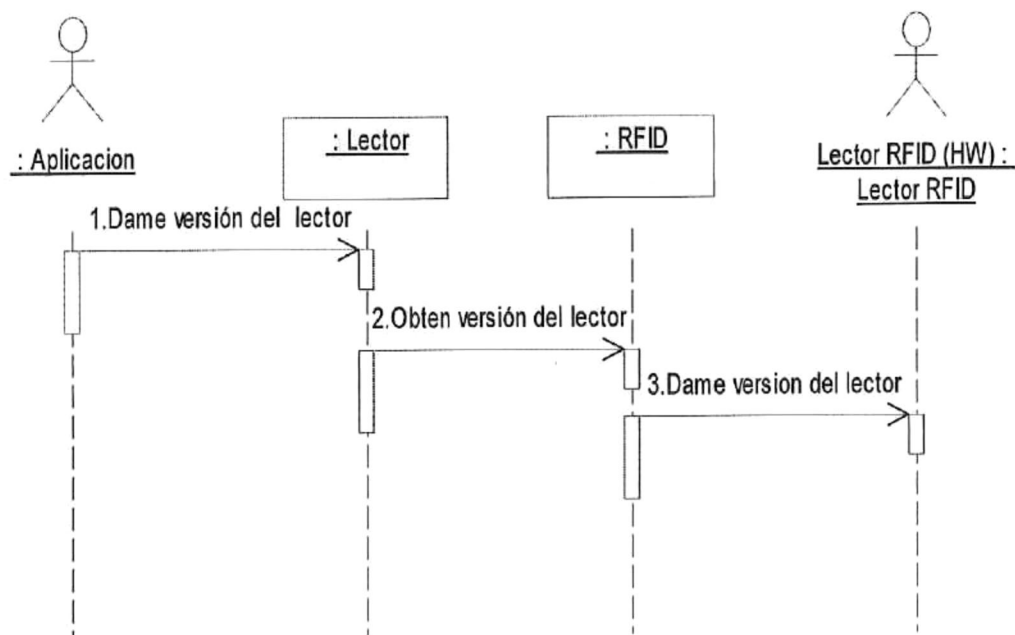
1. La aplicación recibe un error diciendo que no se puede comunicar con el lector
2. El lector RFID no recibe la instrucción porque no está conectado o hubo un problema en el puerto serie
3. Si alguna instrucción es inválida el lector RFID arroja un error y se interrumpe la ejecución de las demás instrucciones

4. La aplicación recibe algún dato erróneo debido a una mala lectura hecha por el lector

Postcondición: Ninguna

Diagrama de secuencia

Nombre: Obtener Versión del Lector



1. La aplicación le envía el mensaje de “Dame versión del lector” a la clase Lector.
2. La clase Lector le comunica a la clase RFID que le obtenga la versión del lector RFID –hablando del hardware- que responda su versión.
3. La clase RFID le ordena al lector RFID –hablando del hardware- que responda su versión.

Diagrama de Colaboración

Nombre: Obtener Versión del Lector

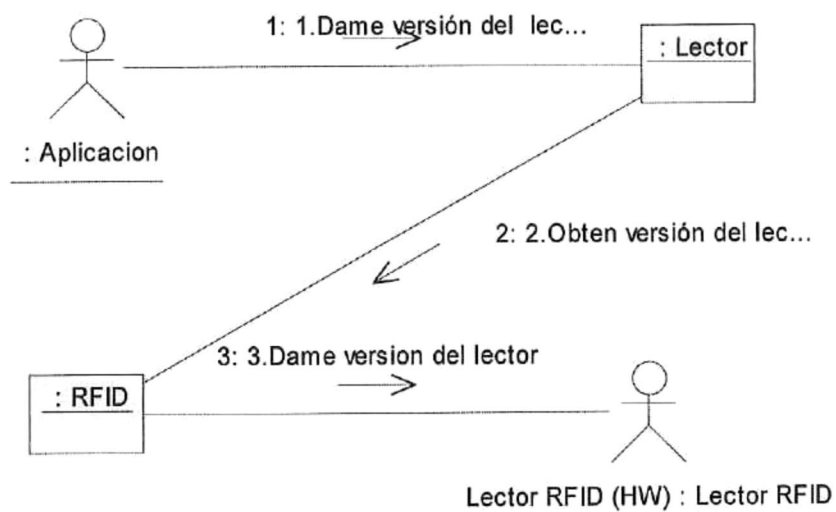
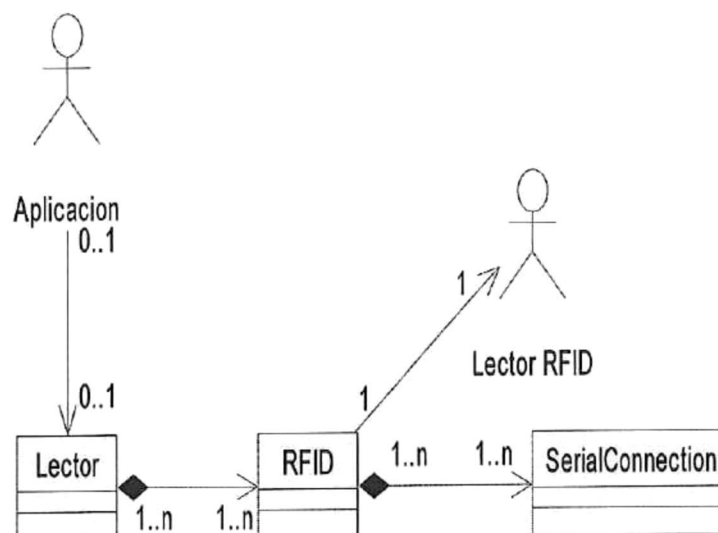


Diagrama de Clases

Nombre: Obtener Versión del Lector



IV.2.2.8 Caso de uso Configurar

Caso de uso: Configurar

Actores: Usuario final

Propósito: Configurar la tasa de bits por segundo, bits de datos, paridad, bits de parada y puerto.

Resumen: Configurar el puerto serie de comunicación del lector-computadora, configurar la tasa de transmisión de datos entre computadora-lector, esta tasa también es denominada la tasa de bits por segundo o baudios. Esta tasa es diferente en cada lector. Además, configurar los bits de datos, la paridad y los bits de parada.

Precondición: El lector RFID tiene que estar conectado en el puerto serie

Curso normal de los eventos

1. El usuario final abre el archivo `rfid.properties` para establecer los parámetros que necesite modificar.
2. El usuario final realiza la modificación de los parámetros del archivo `rfid.properties`.
3. El usuario final guarda los datos los cambios hechos a el archivo `rfid.properties`.

Alternativas:

1. No se puede establecer la nueva configuración porque el archivo `rfid.properties` no se puede abrir.
2. No se puede modificar la nueva configuración en el archivo `rfid.properties` porque el archivo no tiene propiedades de escritura.

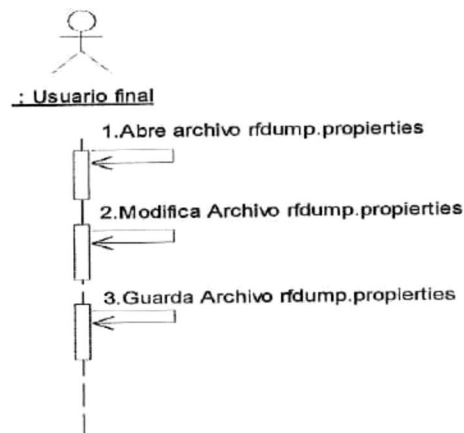
3. No se puede guardar la nueva configuración porque el usuario no tiene privilegios de escritura sobre ese archivo.

Postcondición: Ninguna

Diagrama de secuencia

Nombre: **Configurar**

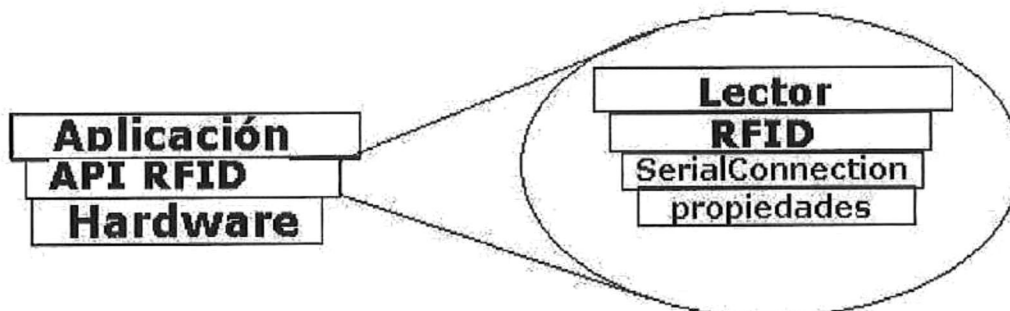
Nota: Aunque el usuario final no es un actor del sistema, en este diagrama, se representará como un actor sólo con el objetivo de mostrar sus actividades.



1. El usuario abre el archivo rfdump.propiedades para establecer los parámetros que necesite modificar.
2. El usuario modifica realiza la modificación de los parámetros del archivo rfid.propiedades.
3. El usuario guarda los datos los cambios hechos a el archivo rfid.propiedades

IV.3 Diseño

IV.3.1 Arquitectura de la Herramienta



La arquitectura del sistema consta de cuatro capas, la cuales son:

- La capa Lector
- La capa RFID
- La capa SerialConnection
- La capa propiedades

La capa Lector

La capa lector es donde se encuentra la clase que contiene todos los métodos del lector los cuales utilizará tanto para establecer la comunicación entre computadora-lector como la lectura ó escritura de datos de las etiquetas y hasta del mismo lector así como la configuración del lector. Además de otras clases que complementan la funcionalidad de esta clase.

La capa RFID

En esta capa se encuentra la clase que contiene los métodos del lector RFID, hablando del hardware, estos son los comandos que se utilizan para manipular por una parte al lector RFID y por otra parte a la información que contienen las etiquetas.

La capa SerialConnection

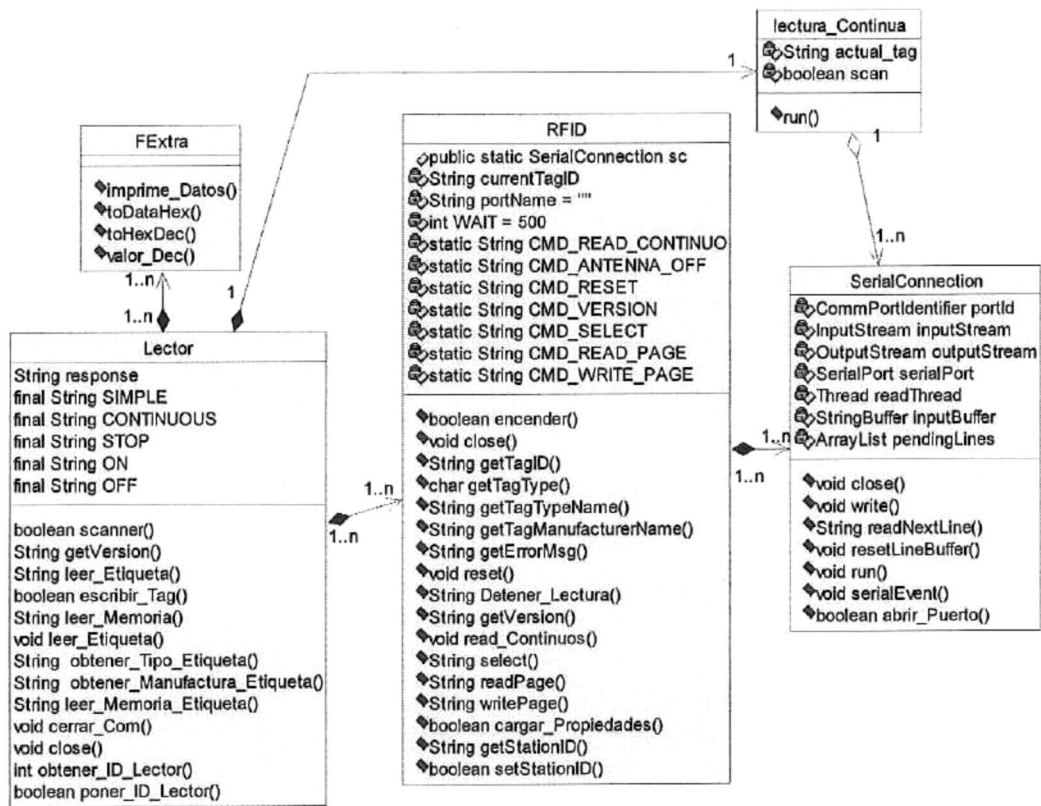
La capa SerialConnection es una clase que para lograr su cometido -el cual es la manipulación y administración de los comandos que se envían por el puerto serie- se basa en la implementación de otra interfaz llamada *javax.comm*. Esta clase está facultada para poder establecer la configuración del tanto del lector como del puerto serie.

La capa Propiedades

En la capa propiedades solo se encuentra el archivo que contiene todas las propiedades del lector desde el número del puerto serie por el cual se comunicará el lector con la computadora, pasando por el nombre de los comandos que se utilizan en cada lector hasta las propiedades de configuración del puerto del lector.

En cada cambio de lector si su configuración es diferente a la establecida por el archivo de propiedades, sólo se cambia los datos que se requieran.

IV.3.2 Diagrama General de Clases



IV.3.3 Criterios de Diseño

Respecto al lector y las etiquetas, ambos RFID, sólo se implementaron los comandos básicos (leer, escribir, encender lector, abrir puerto, etc.) de un lector RFID ya son los comandos mínimos semejantes que todos los lectores tienen incluidos en su funcionalidad, además la estructura de estos comandos es muy parecida. No se implementaron comandos más avanzados (setBroadcast, setTimeout, etc) porque se desconoce la estructura de manejo de dichos comandos y si es que todos los lectores RFID incluyen estas funcionalidades.

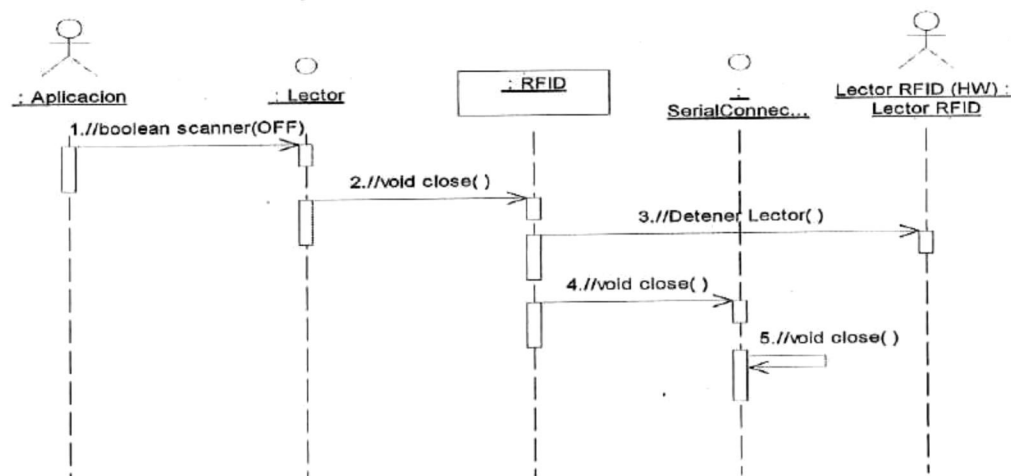
La Clase lectura_Continua es un hilo, ya que cuando se pone a el lector RFID en modo de lectura continua se necesita establecer un ciclo infinito de lectura para no perder la comunicación con el lector RFID, sino se hiciera de este modo, al ejecutar el ciclo de lectura continua sin un hilo, la aplicación se quedará congelada hasta que muera dicho hilo.

La clase lectura_Continua tiene un atributo booleano con nombre "scan" cuyo objetivo es indicarle al ciclo infinito de lectura que se inicie y por ende que empiece a ejecutar el hilo ó se detenga el ciclo y por consiguiente se termine de ejecutar el hilo.

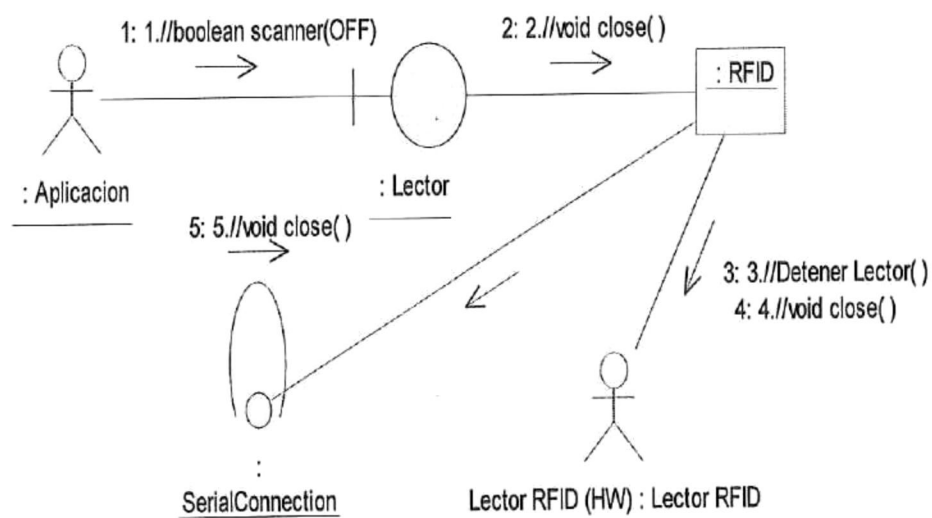
IV.3.4 Diagramas de Interacción en la Realización del Caso de Uso

IV.3.4.1 Caso de Uso: Apagar Lector

- Apagar Lector - Flujo Básico

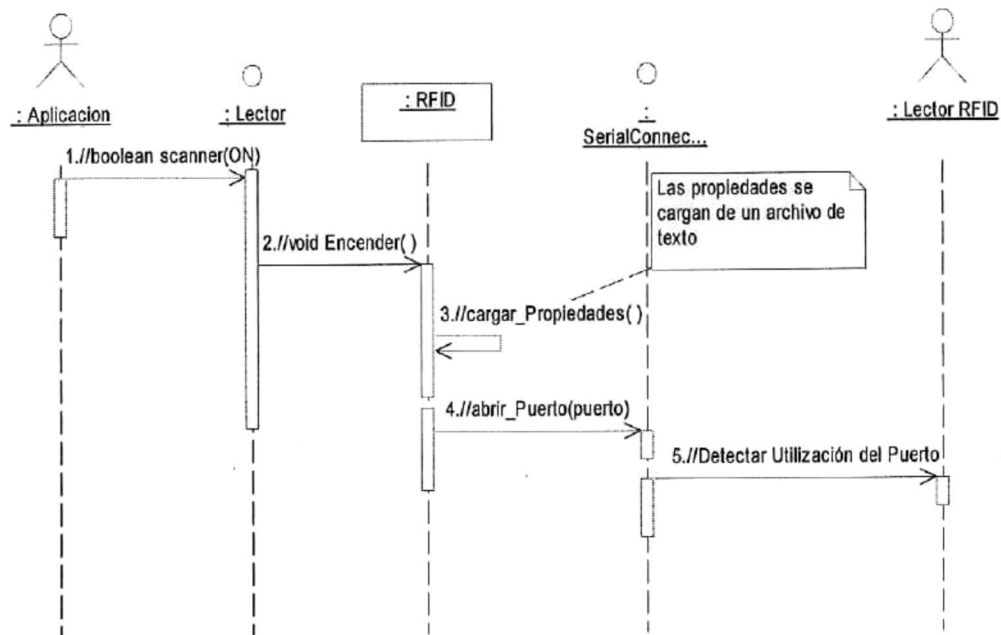


- Apagar Lector – Flujo Básico

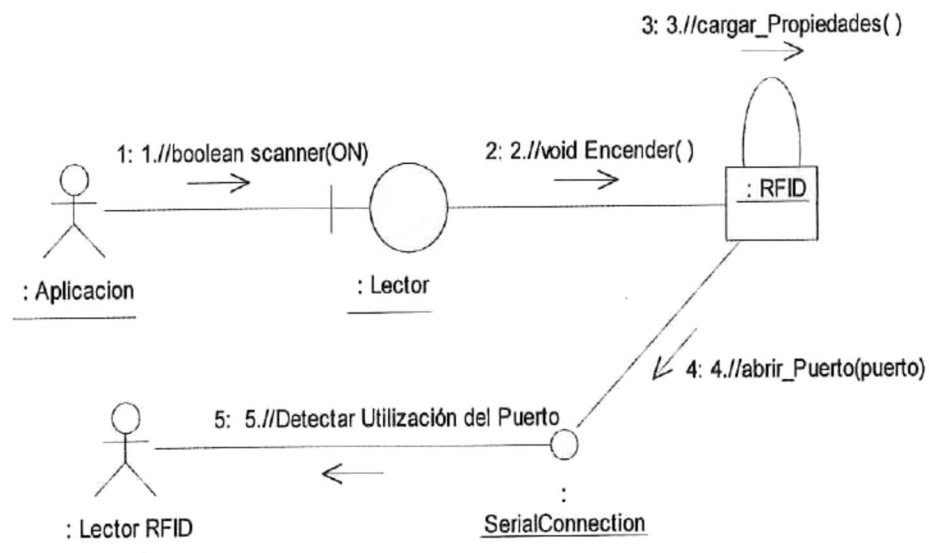


IV.3.4.2 Caso de Uso: Prender Lector

- Prender Lector - Flujo Básico

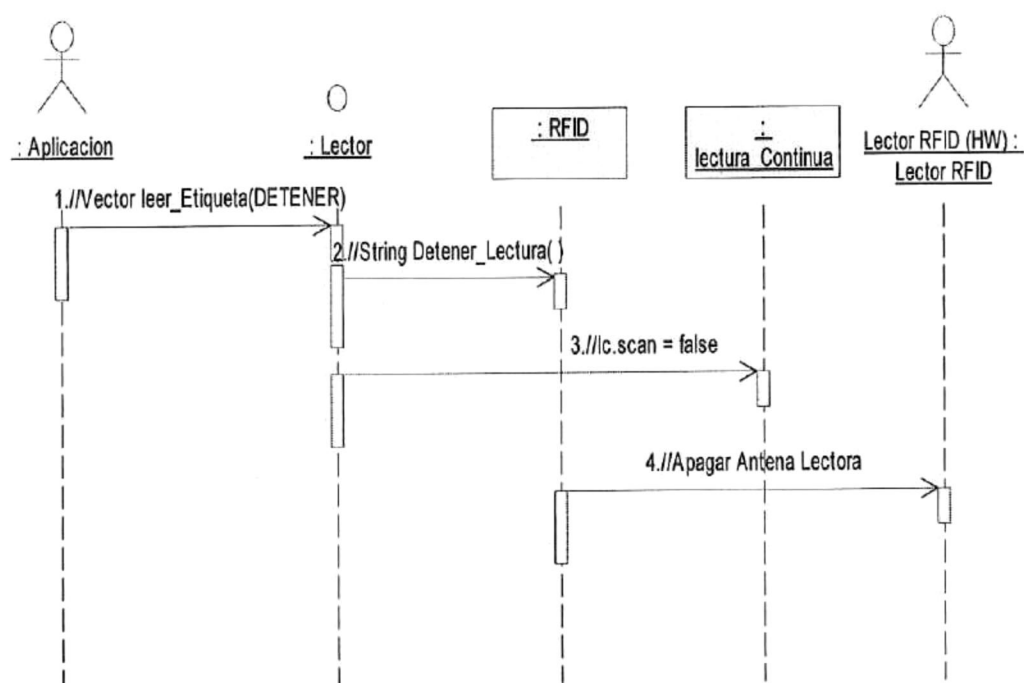


- Prender Lector – Flujo Básico

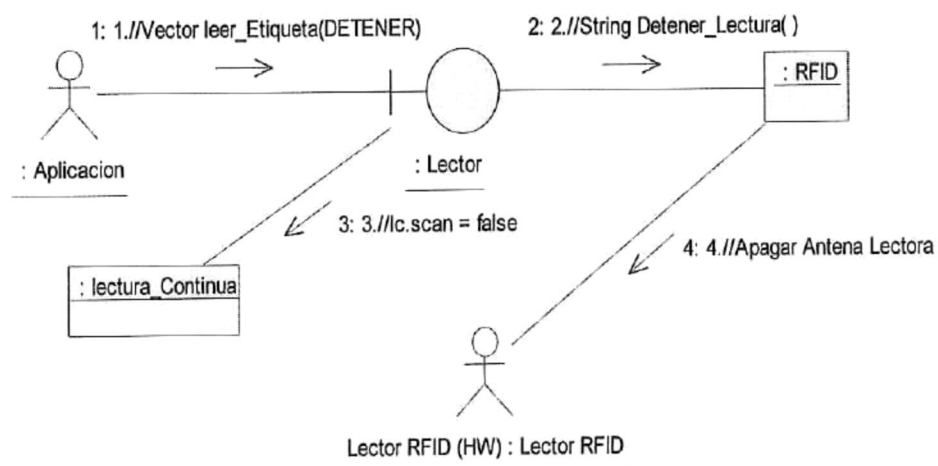


IV.3.4.3 Caso de Uso: Detener Lectura

- Detener Lectura - Flujo Básico

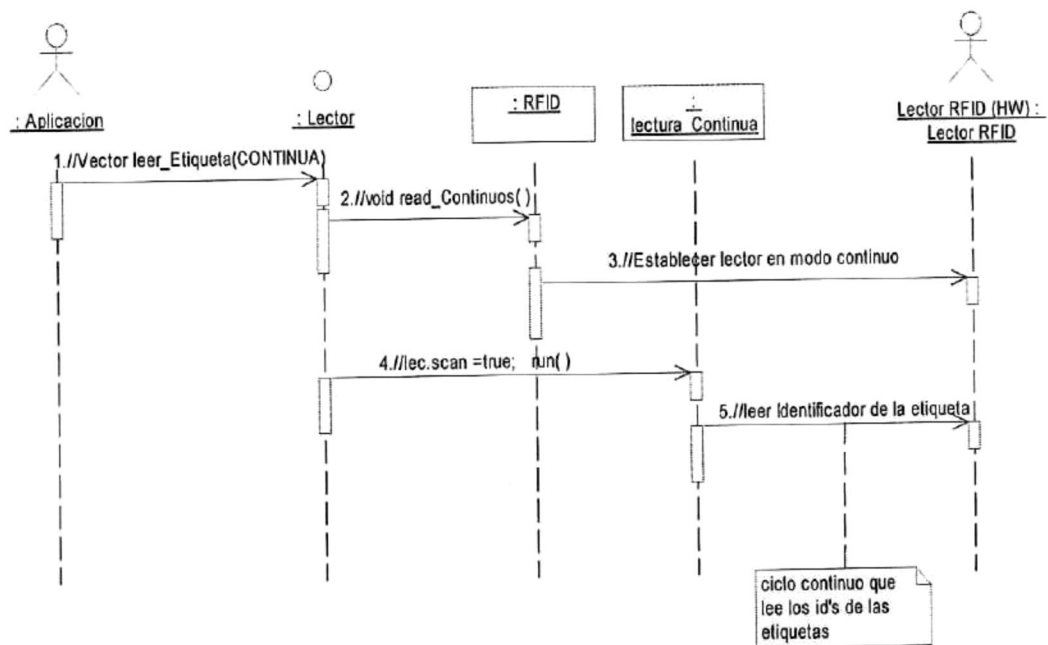


- Detener Lectura - Flujo Básico

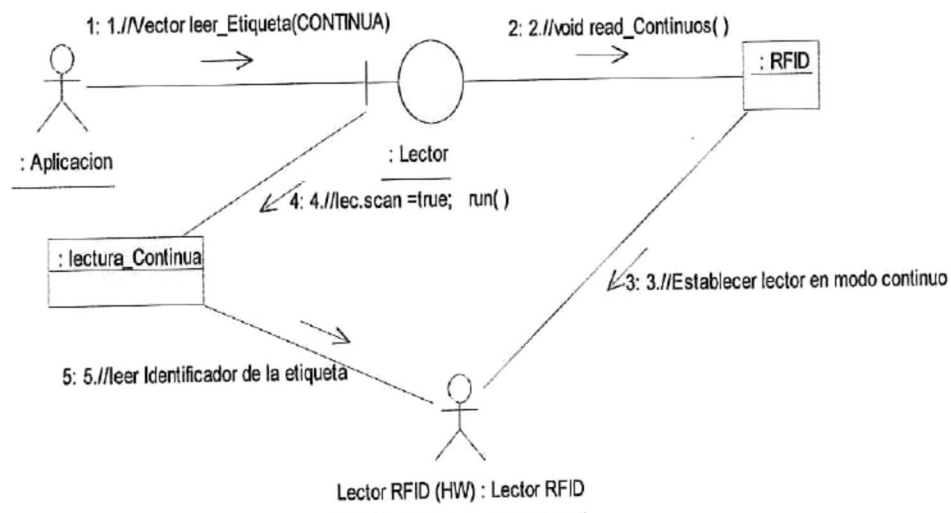


IV.3.4.4 Caso de Uso: Lectura Continua

- Lectura Continua - Flujo Básico

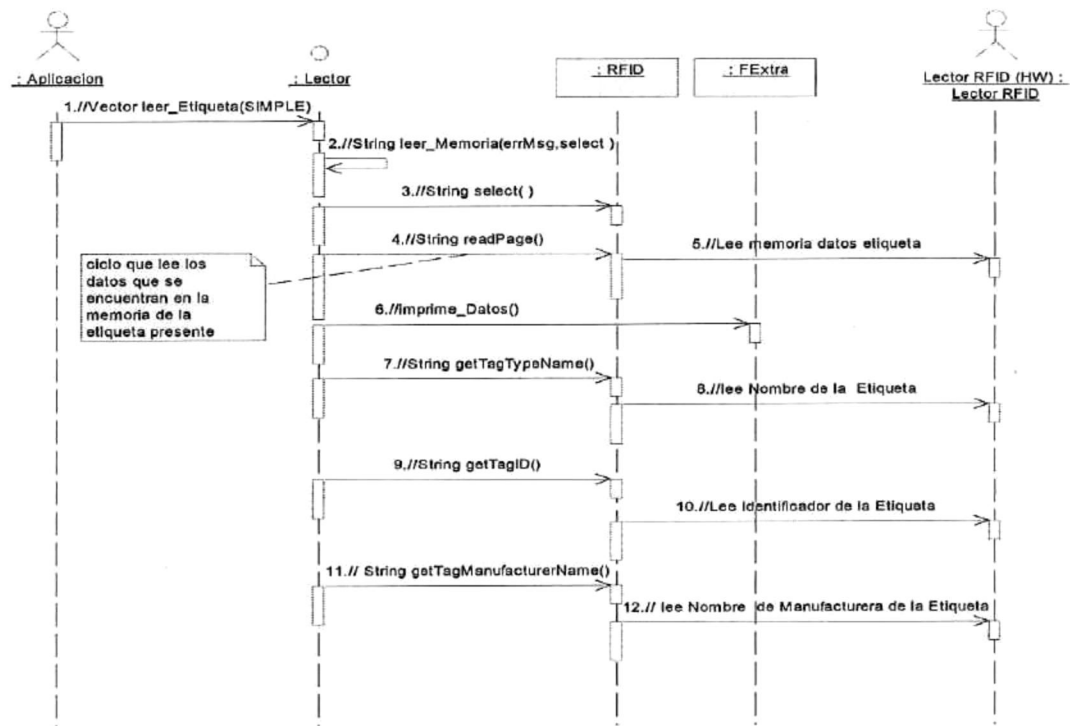


- Lectura Continua - Flujo Básico

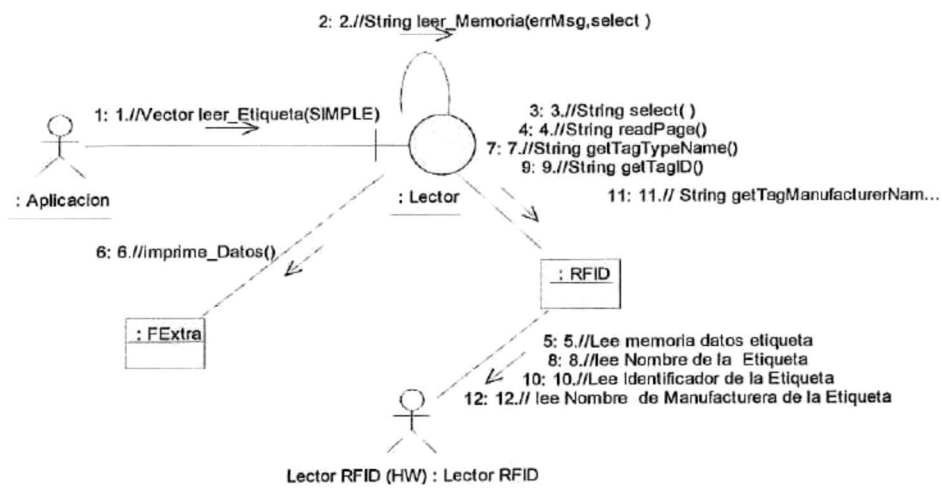


IV.3.4.5 Caso de Uso: Lectura Simple

- Lectura Simple - Flujo Básico

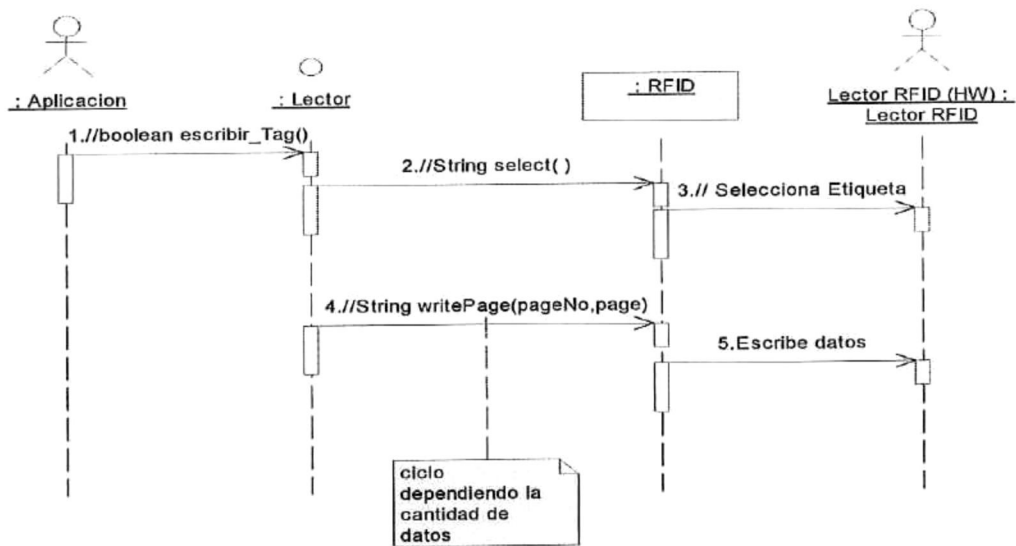


- Lectura Simple - Flujo Básico

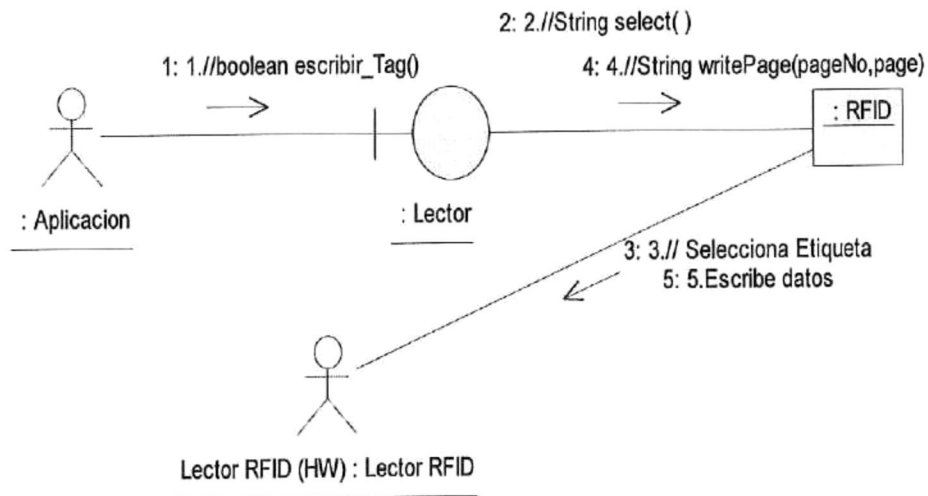


IV.3.4.6 Caso de Uso: Escribir Datos

- Escribir Datos - Flujo Básico

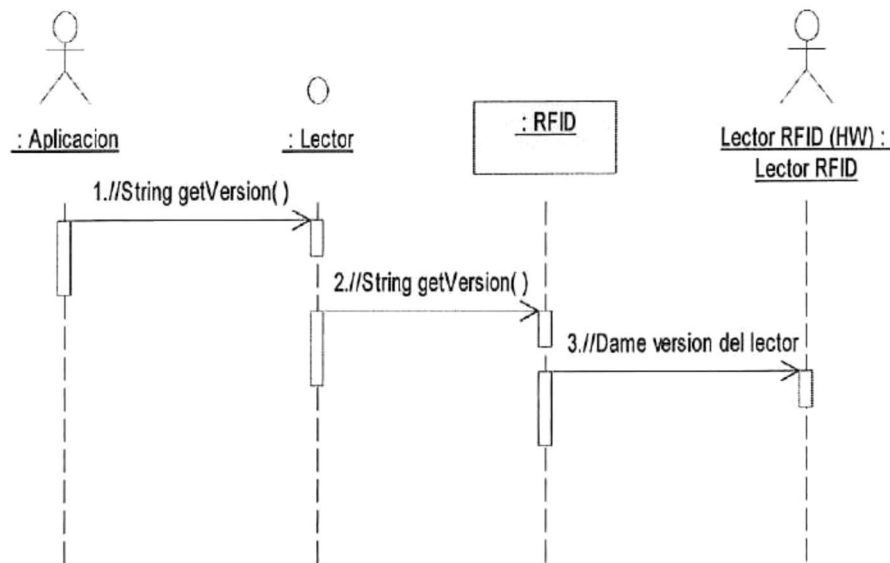


- Escribir Datos - Flujo Básico

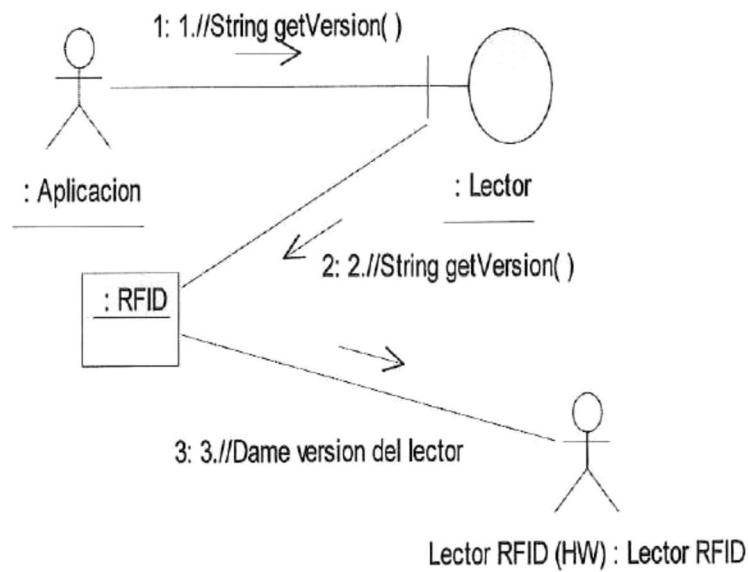


IV.3.4.7 Caso de Uso: Obtener Versión

- Obtener Versión - Flujo Básico



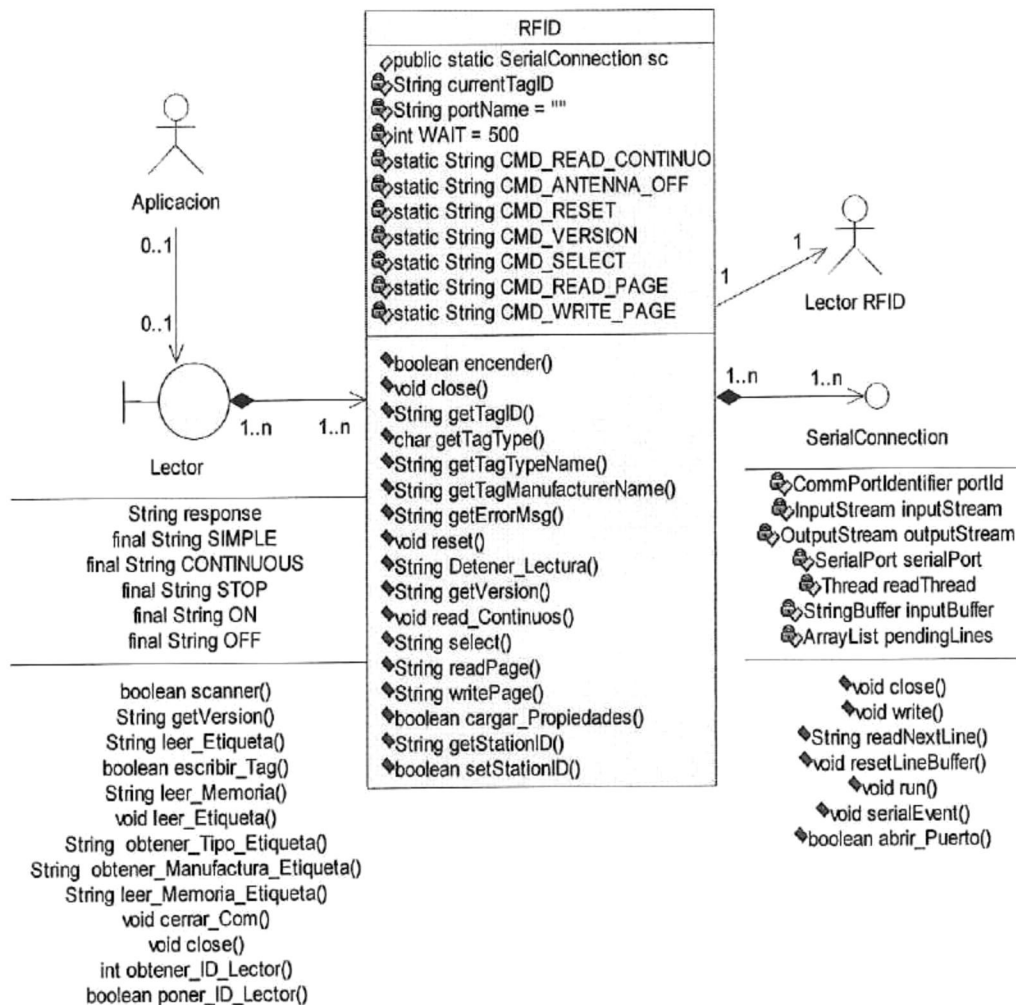
- Obtener Versión - Flujo Básico



IV.3.5 Vista de Clases Participantes en la Realización del Caso de Uso

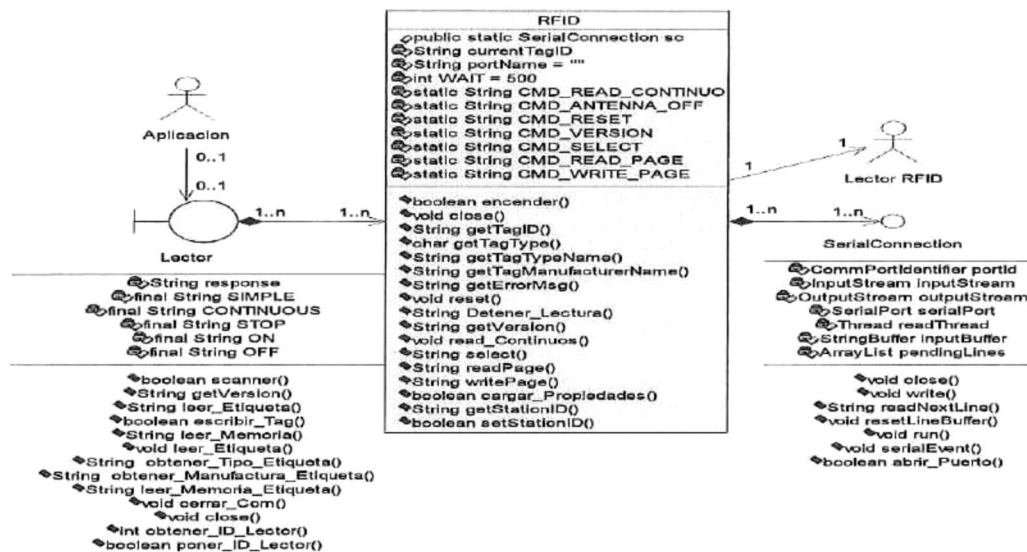
IV.3.5.1 Caso de Uso: Apagar Lector

- Apagar Lector - Flujo Básico



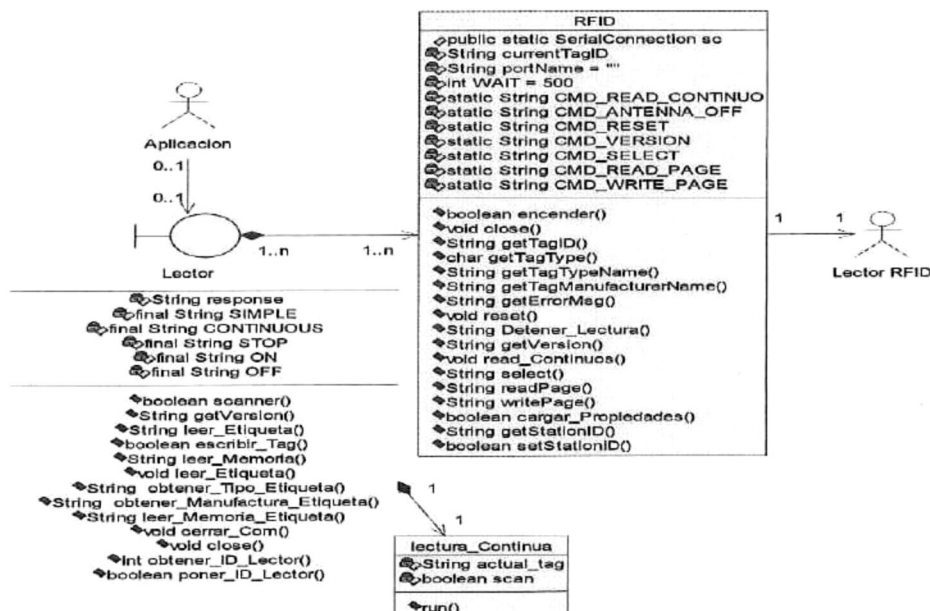
IV.3.5.2 Caso de Uso: Prender Lector

- Prender Lector - Flujo Básico



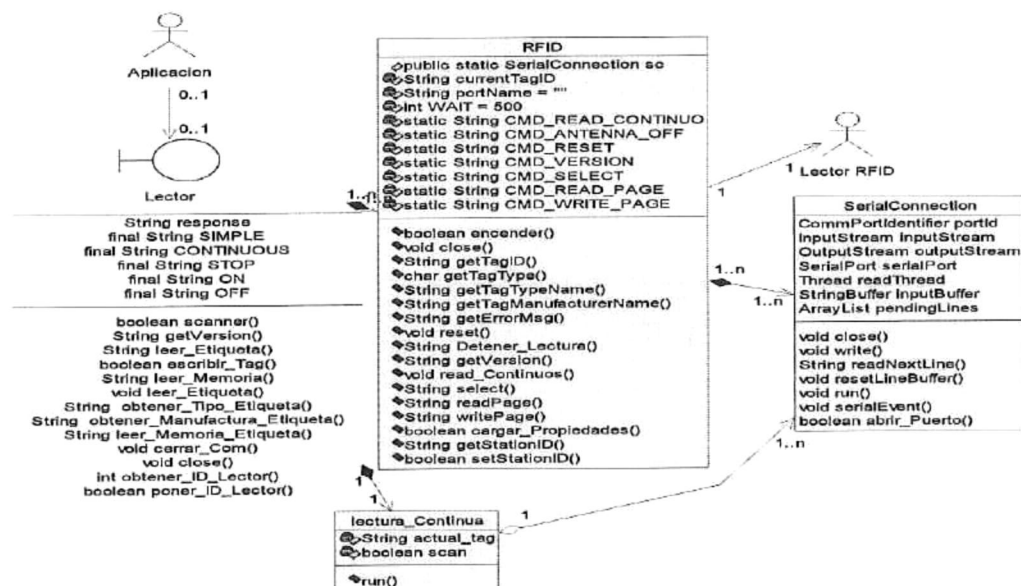
IV.3.5.3 Caso de Uso: Detener Lectura

- Detener Lectura - Flujo Básico



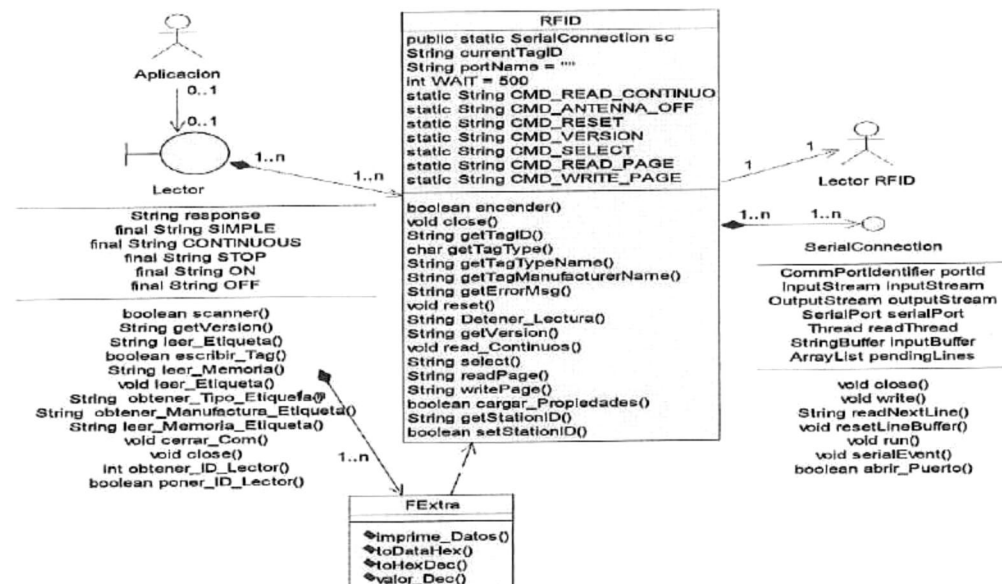
IV.3.5.4 Caso de Uso: Lectura Continua

- Lectura Continua - Flujo Básico



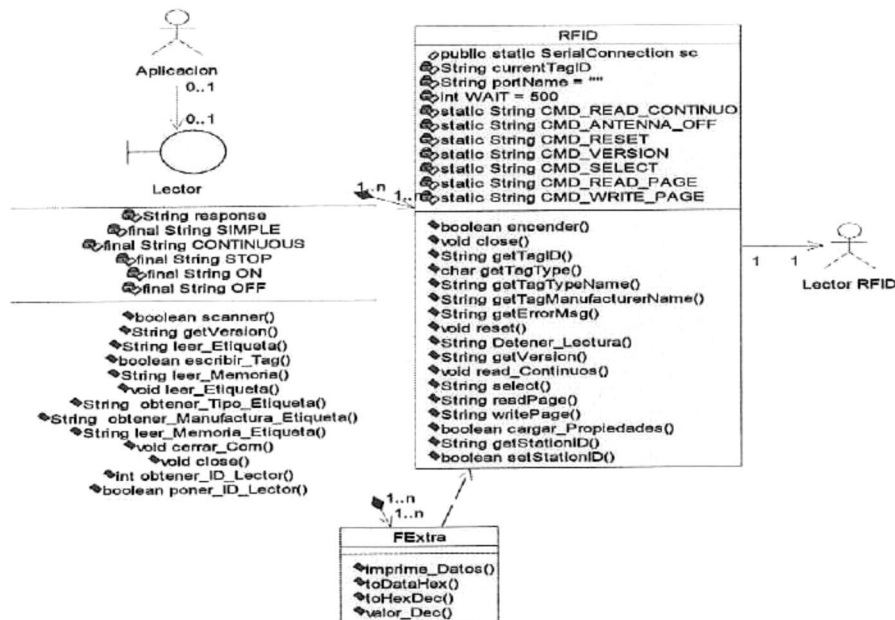
IV.3.5.5 Caso de Uso: Lectura Simple

- Lectura Simple - Flujo Básico



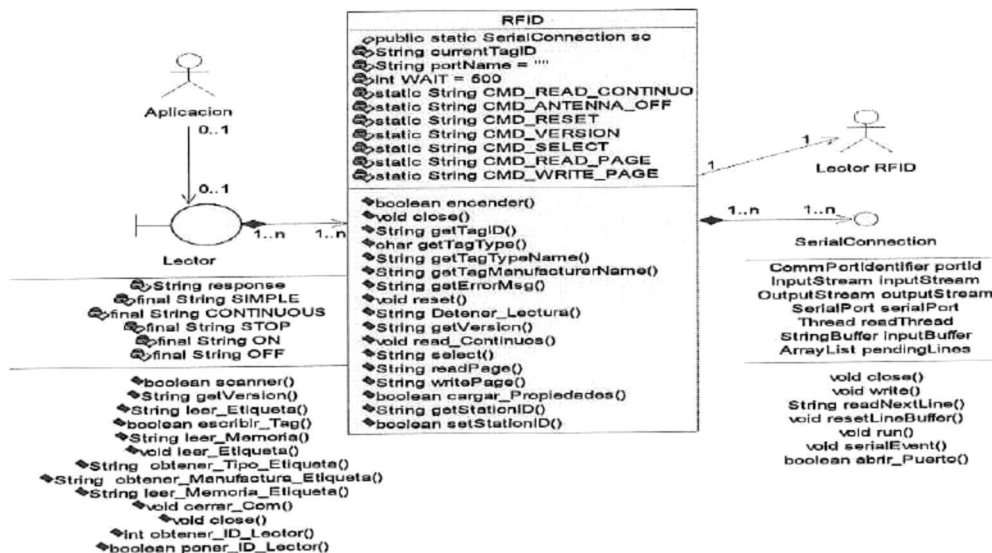
IV.3.5.6 Caso de Uso: Escribir datos

- Escribir datos - Flujo Básico



IV.3.5.7 Caso de Uso: Obtener Versión del Lector

- Obtener Versión del Lector - Flujo Básico



V Implementación

V.1 Implementación para PDA

Respecto a las limitaciones que tienen las máquinas virtuales tanto en paqueterías como en soporte de operaciones aritméticas y de concurrencia (ver Máquinas Virtuales para PDA's) se muestra un diagrama de componentes (ver Figura 11) donde se especifica que para abrir una comunicación a través del puerto serie de una PDA se necesita un paquete denominado *javax.comm* para Windows CE, el cual está constituido de forma diferente a el paquete *javax.comm* que se utiliza para las distintas versiones de Windows 32. Por ello, en este momento la discusión se centrará en la limitante de paqueterías para el manejo de llamadas al puerto serie, donde ésta paquetería es la base esencial de la herramienta y sin ella no se podría extender su portabilidad a la plataforma Windows CE.

Habiendo mencionado lo anterior, y dejando claro algunos aspectos en la diferencia existente entre los paquetes *javax.comm* para las dos plataformas, procederemos a explicar la implementación de esta herramienta en la plataforma de Windows CE.

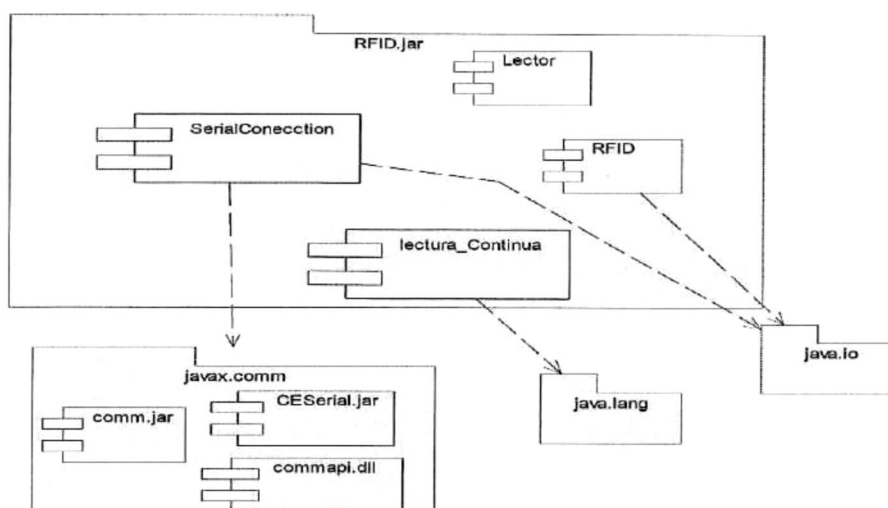


Figura 11 Diagrama de componentes: Entornos Windows CE para Pocket PC

Como se mencionó anteriormente para que la herramienta pudiera funcionar en el ambiente Windows CE ocupó un paquete `javax.comm` especial, este paquete está disponible en el siguiente hipervínculo: <http://www.teilo.net>, aquí pueden encontrar tanto el paquete `javax.comm` como su API de desarrollo. El como instalar y configurar este paquete se puede leer en el apartado de “Instalación y Ejecución en PDA” del Apéndice C: Paquetes para Puertos Serie.

Después de realizar lo anterior se verá como instalar el paquete que contiene las clases RFID el cual es “*jaih.ldml.rfid*”, la instalación y utilización de este paquete es similar al `javax.comm` para PDA’s. Respecto a la utilización, sólo basta con importar el paquete en la clase en donde se vaya a utilizar, y agregar la ubicación de este paquete en la variable de ambiente ***classpath*** de la máquina virtual que estemos utilizando (ver Apéndice C: Paquetes para Puertos Serie). Recuerde que la aplicación se desarrolla en la computadora de escritorio (ver Instalación en Computadoras de

Escritorio). Y posteriormente se copian los archivos “.class” a la PDA para su ejecución.

Ejemplo: Importar el paquete **jaih.ldml.rfid** en una clase determinada

1). Importar todas la clases del paquete

```
import jaih.ldml.rfid.*;  
  
public class “nombre de la clase” {  
  
    //cuerpo de la clase  
  
}
```

2). Importar una clase especifica del paquete

```
import jaih.ldml.rfid. “Nombre de la clase”;  
  
public class “nombre de la clase” {  
  
    //cuerpo de la clase  
  
}
```

V.1.1 Protocolo del Puerto Serie

Otro detalle importante que no se debe pasar por alto es que, para poder abrir un puerto en una PDA, el nombre del puerto para una PDA tiene un protocolo diferente con respecto al protocolo del puerto de una computadora de escritorio. El **protocolo** es el siguiente “nombre del puerto” seguido del número del puerto “número” y por último seguido de los dos puntos “:”.

Por ejemplo: Si se quisiera abrir el puerto COM6 sería así: **“COM6:”**

Si el nombre del puerto no cumple con este protocolo, el puerto serie de la PDA no se abrirá y marcará una excepción de puerto irreconocible.

V.2 Implementación para entornos Win32

Como se mencionó en el apartado de “Implementación para PDA”, los paquetes javax.comm son diferentes, en la Figura 12 seguida se muestra un diagrama de componentes, en el cual se hace notar las relaciones entre paquetes y se aclara el contenido del paquete javax.comm para sistemas operativos win32.

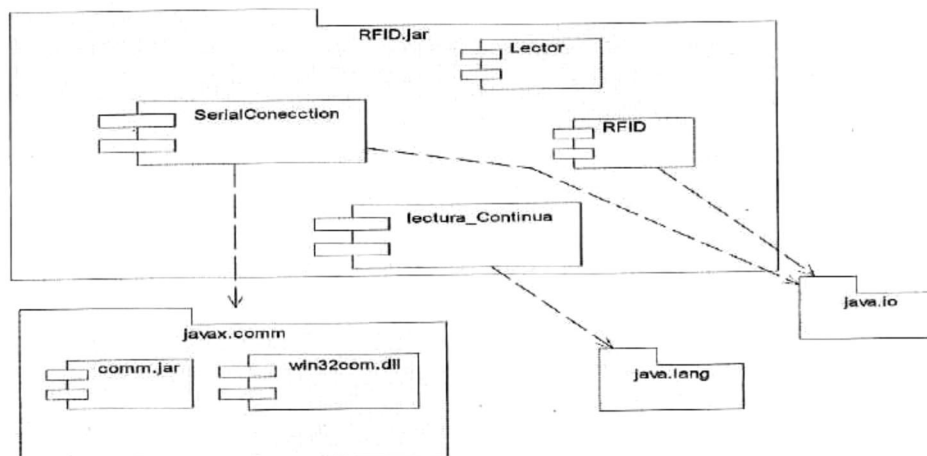


Figura 12 Diagrama de componentes para la aplicación en entornos Windows 32

El paquete **javax.comm** se puede descargar de los siguientes hipervínculos
<http://java.sun.com/products/javacomm/downloads/index.html> y
<http://llk.media.mit.edu/projects/picdev/software/javacomm.zip>

Con respecto a la instalación y utilización del paquete javax.comm, lea el apartado “**Instalación en Computadoras de Escritorio**” del Apéndice C: Paquetes para Puertos Serie.

V.2.1 Protocolo del Puerto Serie

El **protocolo** es el siguiente “nombre del puerto” seguido del número del puerto “número”. Por ejemplo: Si se quisiera abrir el puerto COM6 sería así: “**COM6**”

VI Conclusiones y Perspectivas

- Con éxito se logró el desarrollo de un conjunto de clases bases en lenguaje Java para poder desarrollar aplicaciones con tecnología RFID. El desarrollo de éste conjunto de clases bases liberará al usuario de la implementación de rutinas de bajo nivel para poder acceder a los dispositivos RFID y así poder desarrollar éste tipo de aplicaciones. Además se logra que el programador no sea dependiente del archivo.dll que algunos lectores RFID incluyen ó del lenguaje de programación que la empresa desarrolladora del lector RFID recomienda.
- Esta API no implementa los comandos particulares de cada lector RFID, pero si implementa los comandos básicos de los lectores RFID, siendo estos los necesarios para realizar aplicaciones para dispositivos RFID. Este conjunto de clases bases no se logró aplicar sobre un escenario real, pero se pudo realizar una aplicación desarrollada en Java, y se ejecutó con éxito tanto en Laptop como en PDA's (IPAQ H3900).
- Se pudo observar que aunque existen demasiadas propuestas de aplicaciones a gran escala en áreas determinadas tanto para el sector público como privado, hoy se encuentran algunas en fase de planeación y muy pocas han sido implementadas y liberadas, hoy podemos decir que el API desarrollado ya se encuentra listo para ser utilizado por programadores que ofrecen soluciones con tecnología RFID lo que sitúa a esta investigación como una aportación importante en esta área.

- Mediante UML (*Unified Modeling Language*) fue posible establecer los requerimientos y estructuras necesarias de la herramienta previo al proceso de escribir código. Lo anterior permitió realizar los diagramas de la herramienta correspondientes tanto del análisis como del diseño, sin tener que ser dependientes del lenguaje en el que finalmente ha de ser implementando la herramienta. Como la base de UML radica en un paradigma orientado a objetos, está hizo fácil la construcción de los diagramas y además ofrece la característica de que el análisis y diseño pueden ser cambiados con gran facilidad y adaptados al lenguaje de programación orientado a objetos que se tenga disponible.
- El haber realizado las fases de Análisis y Diseño de este conjunto de clases RFID fue con el objetivo de tener las bases para poder agregar funcionalidades e ir incrementando su compatibilidad con los diferentes lectores y transponders RFID, todo lo anterior con el objetivo futuro de poder construir un middleware.
- Con este conjunto de clases bases RFID se podrá manipular a todos los lectores RFID que cumplan con la versión ISO 0.9v. Además, este conjunto de clases nos independiza de los altos costos de las soluciones RFID que ofrecen las compañías y proveedores de dispositivos RFID.
- Esta herramienta será *Open Source* (Código Abierto) para los usuarios que deseen utilizarla y agregarle nuevas funcionalidades. Además el lenguaje seleccionado “Java” amplía la portabilidad de la aplicación en diferentes sistemas operativos.

VII Apéndice A. Tipos de Lectores

VII.1 Lector RFID para Alta Frecuencia

Esta es la antena Pasiva de Largo Alcance LRP2000 y combinado con una etiqueta (*tag*) de dimensión 14*9 cm. tiene un alcance de rango de lectura de hasta 2.44 metros pero se garantiza un rango de lectura optimo de 2 m [Web 16].

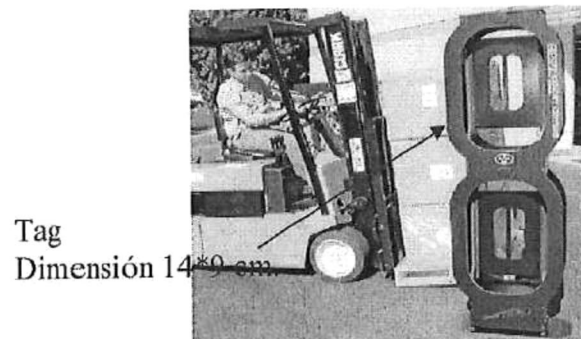


Fig. # Antena LPR2000

VII.2 Lector RFID para Baja Frecuencia

Utilizando un lector y combinado con un transponder cilíndrico de 120mm se puede lograr un área de lectura de hasta 2 m [Web 17].



Fig. # Transponder Cilíndrico de 120mm

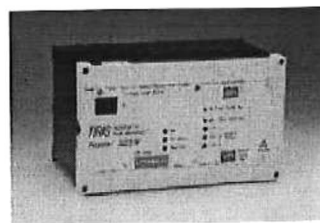
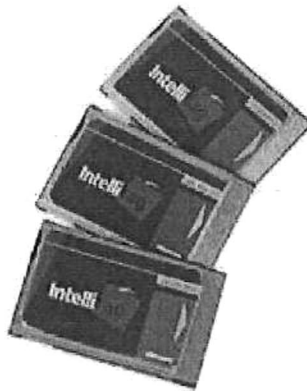


Fig. # Lector S251b serie 2000

VII.3 Lectores RFID para Ultra Alta Frecuencia

A continuación se muestran algunos lectores RFID en frecuencia UHF, así como sus características, tanto las imágenes como la información fueron extraídas directamente de la página del proveedor [Web 3]:

Lector Intellitag® UHF PC Card



Características

- Tarjeta PCMCIA Type II
- Protocolo Anticolisión
- Integración para aplicaciones con intervalo de 2 m.
- Adaptable a Laptop's, PC's, terminales o a cualquier otro

dispositivo con adaptador PCMCIA

Lector Intellitag® UHF P13**Características**

- Interface de comunicación: *Infrared data connection to 700 Color*
- Antena Interna, polarizado circular
- Identifica arriba de 6 *tags* por segundo
- Puede retener arriba de 100 ID's de *tag* en memoria volátil
- Leer un *tag* (8 bytes de datos) dentro de 12 metros.
- Rango de lectura 1.5 metros con antena simple y tag de 105 mm

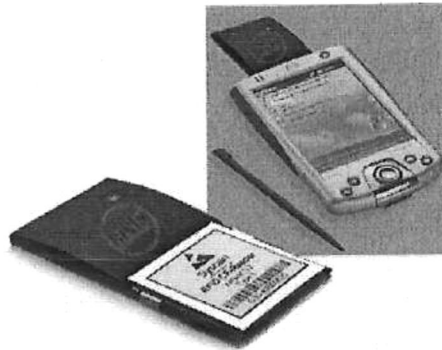
Lector Intellitag® UHF OEM**Características**

- Opciones del sistema flexible
- Rango de lectura arriba de 7 metros
- Protocolo Anticolisión

VII.4 Lector para PDA

ACG RF PC Handheld Reader

Es un lector con interfaz compact flash tipo 2. El lector puede ser conectado a los dispositivos de mano (*handheld*) más típicos y computadoras personales a través de una ranura compact flash (CF) tipo 2 o adaptador PCMCIA. Las *PDA*s, *PC*s, *laptops* y computadoras personales de escritorio pueden ser actualizadas con el lector RF PC *Handheld* para convertirlos en lectores RFID sin la necesidad de integrar hardware adicional, reemplazando el lector RFID móvil típico.



Características

- **Tamaño:** 43,0 x 84,0 x 7,0 mm
- **Sistema Operativo Soportada:** Windows CE 3.0 Pocket PC 2002
Windows CE 4.2 Pocket PC 2003
- **Distancia de lectura:** arriba de 90 mm. (dependiendo del transponder)
- **Interfaz:** Compact Flash Tipo 2

VII.5 Lectores para Identificadores de Animales

Tanto la información como las imágenes de los tipos de lectores que a continuación se presenta fueron extraídas directamente de la página del proveedor [Web 14].

Lectores ISO: compatibles con HDX y FDX-B

Gesreader GES2S

El Gesreader GES2S es un lector de mano ISO 11784/85 compatible, preparado para trabajar en condiciones de campo, permitiendo la recogida de datos y su posterior volcado. Equipo versátil y manejable, con teclado de membrana alfanumérico de gran durabilidad.



Gesreader SMART

Lector de mano alimentado por batería y dotado de antena, teclado y visualizador para leer *transponders* y almacenar información relativa a estos para su posterior descarga en PC. El almacenamiento de datos se realiza en una *FlashCard* extraíble. Mediante un lector/grabador de estas tarjetas se pueden descargar la información a un PC y volcar información al aparato. La interfaz con el usuario consta de un teclado de 49 teclas y un visualizador LCD gráfico monocolor de 128x64 píxeles con retroiluminación y contraste graduable. Las comunicaciones se

realizan vía RF, permitiendo configurar, descargar información del aparato a distancias de hasta 30 metros.



Lector F-110

Lector fijo especialmente diseñado para un uso en mataderos o ambientes con alto nivel de humedad.



VIII Apéndice B. Tipos de *Transponders*

Las “**etiquetas**” (RFID *tags*) consisten de una antena, un microchip o un circuito con memoria de datos capaz de ser leído y escrito sin contacto, vía ondas de radio, usando antenas. Las etiquetas de RFID pueden tener información tan simple como el nombre y la dirección del dueño de una mascota, o suficiente información para formar parte de un proceso de manufactura complejo [José Camilo Daccach T, 2005].

VIII.1 Tipos de Etiquetas

Existen 3 tipos de etiquetas, estas pueden ser pasivas, activas o semi-pasivas [Web 13].

Las **etiquetas pasivas** no usan batería ya que emplea la energía recibida de la antena lectora para energizarse y poder transmitir sus datos. La consecuencia es que, generalmente hablando, las transponder pasivos son de un costo mínimo y son más pequeñas; podrán contar con un área de lectura más corta, pero también cuentan con una vida teóricamente indefinida. Este tipo de transponder tiene diferentes formas pero únicamente en alta frecuencia (HF) tiene la forma de etiqueta electrónica o tags.

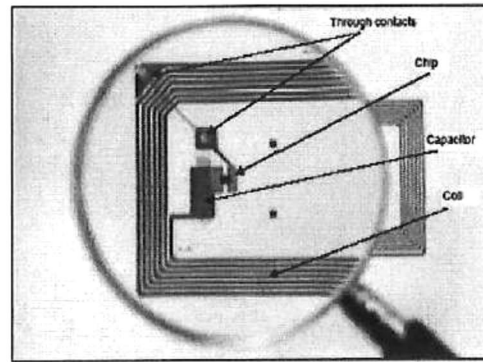


Fig. # Estructura de una etiqueta o tag RFID pasiva en HF.

Las **transponder activos** son energizadas por medio de una pila para transmitir sus datos. Por lo tanto, tienen una vida limitada porque depende de la vida de la pila la cual puede durar en promedio de 3 a 5 años [Web 4]. En consecuencia es más cara pero con la habilidad de proporcionar un rango de lectura mayor al de una etiqueta pasiva, el cual va desde los centímetros hasta los kilómetros, no las hay en formato de tag o etiqueta. La razón es obvia, ya que como podría alojarse una pila en un tag.

Los **transponder semi-pasivos**, estas son las etiquetas con una fuente de poder (generalmente un laminar, es flexible, tiene una batería de costo bajo) las cuales pueden ser usadas por una etiqueta que detecta (e.g, temperatura), pero no aumenta el área de lectura [Web 8 op. Cit.].

VIII.2 Modos de Operación

Las etiquetas o tags pueden clasificarse en Solo-Lectura, Lectura-Escritura, o Combinadas:

Una **etiqueta de solo lectura** consiste en un código único EPC de 64 0 96 bits almacenado en su memoria

Una vez que una etiqueta de Solo-lectura ha sido programada, ya sea durante su manufactura, o previamente a su primer uso con un código de identificación único, este no puede ser cambiado. Tales etiquetas son comúnmente usadas en placas automovilísticas, enlazando la etiqueta de identificación con un registro en una base de datos alojada en una computadora describiendo el objeto "etiquetado" [Web 4 op. Cit.].

Las **etiquetas de Lectura-escritura**, ofrecen la habilidad de contar con información actualizada o agregada, ya que se puede escribir y/o borrar información cada vez que se requiera hacerlo, así la información se va actualizando. Por lo tanto son aplicables para requerimientos de información variable. Una gran aplicación de estas etiquetas es en los inmovilizadores automovilísticos, donde la etiqueta cuenta con un código rodante actualizado cada vez que se utiliza como medida de seguridad. En estas etiquetas el área de escritura es menor al de lectura.

Las **etiquetas combinadas** o WORM (Write Once Read Many – Escribe una vez, lee muchas veces) poseen la habilidad de que cierta información (tal como el número de serie) es permitida ser codificada por el usuario después el usuario bloquea el código almacenándolo permanentemente, mientras que otra parte de la información se deja en memoria para una codificación posterior, o para ser actualizada durante su uso [Web 13 op. Cit.].

VIII.3 Formas de las Etiquetas

Las etiquetas RFID vienen en un gama de formas y tamaños diferentes, esto para que tengan la capacidad de ser aplicadas a cualquier tipo de aplicación y ambientes. El objetivo es que no haya obstáculos que impidan su utilización. Las siguientes formas son las más comunes en los sistemas RFID:

- **Etiqueta:** La etiqueta es plana, delgada y de forma flexible
- **Ticket:** Una etiqueta plana, delgada y flexible sobre papel
- **Tarjeta:** una etiqueta delgada y plana incrustada en un plástico duro para una vida larga
- **Glass bead:** Una etiqueta pequeño dentro de una cuenta cilíndrica de vidrio, usado para aplicaciones tales como etiquetado de animales (ejemplo: bajo la piel)
- **Banda para la Muñeca:** Una tag insertada dentro de una banda de plástico para muñeca
- **Botón:** Una pequeña etiqueta encapsulada en empaque rugadizo rígido

VIII.4 Transponders Pasivos TIRIS HDX 134,2 Khz. (LF)

La información y las imágenes que comprenden a los tipos de transponder pasivos de esta sección fueron extraídas directamente de la página del proveedor [Web 15].

Los diferentes formatos disponibles, se adaptan a cualquier necesidad de identificación de los sectores industriales. Dependiendo de la necesidad de su aplicación, usted podrá optar por utilizar modelos de sólo lectura con código único mundial e inviolable, o por modelos de lectura/escritura con distintas capacidades de memoria de datos.



VIII.4.1 Ventajas clave

- Máximas distancias de lectura/escritura en tecnología de baja frecuencia (hasta 240 cm.)
- Distintas capacidades de memoria de datos.
- Modelos herméticos, para ambientes agresivos.
- Transponder para entornos de temperaturas extremas.
- Tiempo de funcionamiento ilimitado (sin baterías).

VIII.4.2 Capacidades de Memoria de Datos.

- Modelos de sólo lectura (R/O) 64 bits programados en fábrica, con garantía de código único mundial.

- Modelos de lectura/escritura de 64 bits de datos (R/W).
- Modelos multipágina de lectura escritura (MPT) con memoria de 1360 bits estructurados en 17 páginas.
- Modelo con respuesta por direccionamiento de memoria.

VIII.4.3 Modelos de Transponder

Cápsula de cristal



Diseñados específicamente para trabajar en ambientes con condiciones extremas y para contacto permanente con líquidos. La protección del transponder es hermética y el rango de lectura es de 60-90 cm. dependiendo del modelo y la configuración de lector/antena utilizado. Para instalación del transponder sobre partes o componentes metálicos existen cápsulas termoplásticas que a la vez le protegen de posibles golpes. Por sus prestaciones técnicas puede ser utilizado en cualquier tipo de aplicación industrial, ganadera, accesos de personal, identificación de productos, etc.

Tarjeta



Las prestaciones de distancia de lectura de hasta 120 cm., hacen de la tarjeta el mejor transponder pasivo Manos libres del mercado. También es utilizada con lectores Microreader (muy económico) en aplicaciones de lectura por proximidad (lectura hasta 40 cm.). Se suministra en blanco para poder ser personalizada por el cliente.

Disco 30mm



Está diseñado con un orificio central para poder ser fijado mediante un sistema de remache rápido en el elemento a identificar. Por sus prestaciones de lectura (100cm) es ideal para la identificación de elementos que se desplazan en las líneas de producción como bandejas, palets o el producto directamente y también para la identificación de contenedores de basura. La cápsula termoplástica de protección es resistente a temperaturas altas y golpes.

Disco 85 mm.



Está diseñado para ser utilizado en aplicaciones de control de acceso de vehículos. Se instala en el interior del vehículo, en la zona del parabrisas. La distancia

de lectura 150 cm. hace posible la automatización de controles de acceso de vehículos con lectura sin detención de la marcha (Velocidad máxima aconsejable 30 Km. /h.)

Cuña 12mm



Es el modelo más pequeño de la familia. Está diseñado para optimizar la distancia de lectura en cualquier orientación respecto a la antena. Este transponder es utilizado por los principales fabricantes del sector de automoción en los sistemas de inmovilización. Por sus características puede ser encapsulado en otros elementos e incluso durante la inyección de determinados plásticos.

Cilindro



Es el modelo de mayores prestaciones de distancia de lectura (hasta 240 cm.) ideal en aplicaciones de identificación de vehículos (flotas de transporte) y contenedores.

M-o-M para metales



Es el transponder ideal para la identificación de elementos metálicos siendo el rango de lectura de 120 cm.

Llavero

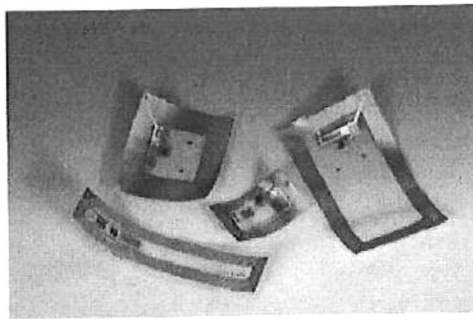


Basado en el transponder de vidrio de 23mm, está diseñado para incorporarlo en un llavero. Es utilizado principalmente en identificación de personal y como doble sistema de seguridad en los sistemas de inmovilización de vehículos.

VIII.5 Transponders Pasivos Tag-it HF 13,56 MHz

La información que comprender a los tipos de transponder pasivos Tag-it de esta sección fue extraída directamente de la página del proveedor [Web 15].

Tag-it™ es una nueva generación de Transponder pasivos de Texas Instruments, desarrollados como base para la creación de etiquetas electrónicas destinadas a todos los sectores productivos, ofreciendo soluciones para cualquier control automatizado y haciendo realidad la trazabilidad de un producto desde la fabricación hasta el consumo.



VIII.5.1 Ventajas clave

- Producto disponible conforme a norma ISO de interoperatividad de protocolos de sistemas (ISO15693).
- Distancias de lectura/escritura eficientes para la automatización de procesos
- Memoria de datos de 2 kbits.
- Diferentes modelos de transponder, adaptados a cualquier tipo de aplicación.
- Transponder de bajo coste para aplicaciones de gran consumo.
- Tiempo de funcionamiento ilimitado (sin baterías).
- Sistema anticolidión. Lectura de hasta 30 Transponder por ciclo de lectura.

VIII.5.2 Capacidades de memoria de datos

Modelos conforme a normativa ISO 15693

- 2Kbits de memoria para datos de usuario estructurados en 64 páginas de 32 bits cada una.
- 64 Bits de código serializado programado en fábrica, de solo lectura

Modelos no ISO

- 256 bits de memoria para datos de usuario estructurados en 8 páginas de 32 bits

cada una.

- 32 Bits de código serializado programado en fábrica, de solo lectura

VIII.5.3 Modelos de Transponder

Cuadrado 45x45mm



Disponible en versiones ISO/IEC 15693 apart. 1-2-3 y con resonancias adaptadas para utilización en tarjetas de proximidad y etiquetas adhesivas de identificación de productos también en formato no ISO de 256 Kbits.

Mini 22,5x38mm



Disponible en versiones ISO/IEC 15693 apart. 1-2-3 y con resonancias adaptadas para la utilización en tarjetas y etiquetas adhesivas de identificación de productos. También en formato no ISO de 256 Kbit.

Rectangular de 45x76mm.



Disponible en versiones ISO/IEC 15693 apart. 1-2-3 y con resonancias adaptadas para utilización en tarjetas de proximidad y etiquetas adhesivas de identificación de productos también en formato no ISO de 256 Kbits.

Cinta 17x93 mm.



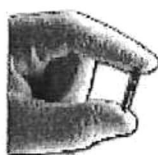
Modelo disponible únicamente en versión no ISO. Este identificador está diseñado especialmente para la identificación de libros en la gestión de bibliotecas.

VIII.6 Tipos de Identificadores para animales

La información y las imágenes que comprende a los tipos de transponder identificadores para animales de esta sección fueron extraídas directamente de la página del proveedor [Web 14].

INJECTABLE MICROCHIP

Microchip de 32 o 23 mm sistema HDX de aplicación mediante inyección subcutánea.



VIII.7 Tipos de Transponder Pasivos en Frecuencia Ultra Alta (UHF)

La información que comprende a los tipos de transponder pasivos en frecuencia ultra alta así como las gráficas de funcionalidad que son presentadas en esta sección fueron extraídas directamente de la página del proveedor [Web 2].

915 MHz Container Tag

La etiqueta inteligente para contenedor INTELLITAG® CONTAINER TAG es un producto durable con alta funcionalidad originalmente diseñado para uso con los palets de plástico. Esta etiqueta tiene uno de los diseños más versátiles dentro del portafolio de INTELLITAG debido a su consistente funcionalidad con un rango ancho de materiales.

Aplicaciones típicas: seguimiento de palets de cartón y contenedores.



915 MHz Reusable Plastic Container (RPC) Tag

La etiqueta inteligente INTELLITAG® RPC TAG fue originalmente diseñada para una función optima sobre plástico. La durabilidad de esta etiqueta la ha hecho atractiva para muchas aplicaciones, incluyendo el seguimiento de bloques de máquinas de camiones durante el proceso de manufactura.

Aplicaciones típicas: Palets, seguimiento de contenedores y cajas de cartón.



915 MHz Reusable Container (RPC) Insert

La etiqueta inteligente INTELLITAG® RPC INSERT es el centro de la etiqueta RPC. La RPC INSERT esta diseñada para un funcionamiento óptimo sobre plástico, es adecuada para encapsulación y ha sido moldeada en numerosos productos para aplicaciones de contenedores inteligentes.

Aplicaciones típicas: Palets, seguimiento de contenedores y cajas de cartón.



915 MHz Encapsulated Stick Tag

La etiqueta inteligente INTELLITAG® ENCAPSULATED STICK TAG tiene un diseño rugadizo que se utiliza en el plano trasero del metal para proveer un funcionamiento consistente sobre una variedad de materiales. El diseño largo del dipolo provee un funcionamiento para un rango superior y es ideal para rastreo de partes de metal y contenedores en ambientes de manufactura.

Aplicaciones típicas: Trabajo en proceso (WIP), Palets, seguimiento de contenedores y cajas de cartón.



IX Apéndice C: Paquetes para Puertos Serie

IX.1 Instalación y Ejecución en PDA

Como se mencionó en el apartado de “Implementación en PDA” se tiene que descargar el paquete javax.comm. Una vez descargado el paquete, su instalación es muy fácil. A continuación se muestran los pasos para dicha instalación:

1. Se descomprime el archivo que contiene a éste paquete, como descomprimir un archivo es cosa fácil, en este apéndice no se abarcará el explicar este proceso.
2. Después de descomprimirlo, aparece una carpeta con nombre de “commapi” adentro de esta carpeta están los archivos requeridos, los cuales son: comm.jar, CESSerial.jar, commapi.dll y javax.comm.properties.
3. Tomamos el archivo commapi.dll y lo copiamos dentro de la carpeta de “Windows” de la PDA, su ruta es “\windows”.
4. Tomamos el archivo javax.comm.properties y lo copiamos en la carpeta de nombre “lib” la cual se encuentra dentro de la carpeta de Windows, su ruta es “\windows\lib”.
5. Los archivos comm.jar y CESSerial.jar se copian dentro del directorio de trabajo, es decir, si tengo los archivos de mi aplicación dentro de una carpeta con nombre “\mi _ aplicación” entonces estos archivos .jar van dentro de esta carpeta. Para poder utilizar estos archivos con la aplicación se tienen que poner en el “classpath” (*ver Ejecutar una aplicación Java en Jeode utilizando el classpath*) de la máquina virtual que se esté utilizando al momento de ejecutar la aplicación.
6. Listo, ya están instalados los paquetes necesarios para poder realizar llamadas al puerto serie de la PDA.

IX.1.1 Formas de Ejecutar una Aplicación Java en Jeode

Ejecutar una aplicación Java en Jeode utilizando el classpath

Antes de seguir con la explicación se hace caso omiso de que el interesado ya tiene instalada la máquina virtual de Java, en este caso Jeode, ya que está es la máquina virtual que se uso en este trabajo. Y además, que tiene conectado y sincronizado el dispositivo (PDA) a su computadora.

Existen dos formas de ejecutar una aplicación Java en la máquina virtual de Java denominada “Joede”. A continuación se muestran estas formas:

La primera, es desde la línea de comandos o consola como se suele llamar.

Primero se ejecuta la máquina virtual para que muestre la consola

Ya estando abierta, procedemos a escribir lo siguiente:

```
-cp \ruta\CESerial.jar;\ruta\comm.jar;\windows\lib\javax.comm.propierties;\ruta\rfid.jar;
\ruta_aplicación “nombre de la clase que contiene el main”
```

Por ejemplo; si se tuviera la aplicación, el CESerial.jar y comm.jar dentro de una carpeta llamada “*mi_aplicacion*”, la línea del classpath quedaría como sigue:

```
-cp
\mi_aplicacion\CESerial.jar;\mi_aplicacion\comm.jar;\windows\lib\javax.comm.propierties;\ruta\rfid.jar;\mi_aplicación “nombre de la clase que contiene el main”
```

Donde “cp” es la abreviación de classpath

Con esta línea de comando le estamos diciendo a la máquina virtual Jeode que ejecute la aplicación que está dentro de la carpeta \mi_aplicación. Sin embargo, al

ejecutar la aplicación la consola se cierra, y no podremos ver los mensajes que nuestra aplicación llegase a mostrar en la consola. Para que la consola se mantenga abierta existe el siguiente comando:

```
-Djeode.evm.console.local.keep=true
```

Este comando se agrega al principio de la línea, quedando esta última como sigue:

```
-Djeode.evm.console.local.keep=true          -cp    \ruta_aplicación\CESerial.jar;\
ruta_aplicación \comm.jar;\windows\lib\javax.comm.properties;\ruta\rfid.jar;
\ruta_aplicación "nombre de la clase que contiene el main"
```

Como nota preventiva, dentro del classpath no deben existir espacios en blanco.

Y la **segunda y más fácil**, es la de crear un acceso directo a la aplicación, esta forma de ejecución es la más cómoda de hacer ya que no hay que estar abriendo y cerrando la consola de Jeode cada vez que se tenga un error. El proceso es el siguiente:

1. Para crear un acceso directo, tenemos que usar el Windows Explorer de nuestra computadora, esta es la forma más fácil de hacerlo.
2. Después de esto dirigirse a la siguiente ruta **Mobile Devices > My Pocket PC > Windows** estando ya ubicados dentro de esta carpeta seleccionamos **"evm.exe"**
3. después seleccionamos **"editar" (edit)** y escogemos **"copiar" (copy)** e ir a la carpeta donde queramos colocar el acceso directo de nuestra aplicación y damos un clic con el lápiz y seleccionamos **"copiar como acceso directo" (Paste Shortcut)**
4. Listo, ya está creado el acceso directo a nuestra aplicación

Realmente esto crea un “**archivo de acceso directo**” a **evm.exe.lnk** dentro de tu PDA, este archivo es sólo un enlace entre la aplicación y la máquina virtual.

Para modificar este archivo lo tienes que copiar a tu computadora y después abrirlo con el “**editor de notas**” (NotePad) o cualquier editor de texto. El contenido mínimo de este archivo es el siguiente:

```
18#"\\Windows\\evm.exe"
```

Ahora puedes editar este archivo para agregarle los parámetros de tu línea de comando. Tomando el ejemplo anterior de nuestra línea de comando ésta quedaría así:

```
18#"\\Windows\\evm.exe" -Djcode.evm.console.local.keep=true -cp \\ruta\\CESerial.jar;
\\ruta\\comm.jar;\\windows\\lib\\javax.comm.propiedades;\\ruta\\rfid.jar;\\ruta_aplicación
“nombre de la clase que contiene el main”
```

Después de esto, se tiene que copiar el archivo de nuevo a la PDA. Y la forma de ejecución es sólo darle un clic sobre el acceso directo.

IX.2 Instalación en Computadoras de Escritorio

Para instalar el paquete javax.comm en una computadora de escritorio o laptop, es de la siguiente manera:

1. Después de que se descomprime el archivo, este muestra una carpeta con nombre de commapi.
2. Dentro de ésta carpeta se encuentran demasiados archivos incluyendo la documentación de ayuda, pero los más importantes son el comm.jar y el win32com.dll.
3. Entonces copiamos el archivo win32com.dll a la carpeta de \Windows\System32\
4. Y listo ya está instalado el paquete en la máquina

Ahora, ¿Cómo se utiliza?, pues bien sólo hay que poner la ruta del archivo en la variable de ambiente classpath de la computadora. En este apéndice no explicaremos como configurar la variable de ambiente classpath de la computadora, sólo se explicará como llamar el paquete comm.jar desde la línea de comandos de la consola.

A continuación se explica como:

La línea de comando para compilar quedaría así:

```
javac -classpath directorio de trabajo;\ruta\comm.jar;\ruta\rfid.jar archivos.java
```

La línea de comando para ejecutar la aplicación quedaría así:

```
java -classpath directorio de trabajo;\ruta\comm.jar;\ruta\rfid.jar "clase con main"
```

Por ejemplo; si yo tengo una aplicación hecha en java en donde la clase que contiene el main se llama **"Interfaz"**, mi directorio de trabajo se llama **"aplicación"** y el archivo **"comm.jar"** se encuentra en una carpeta con nombre "commapi" la cual esta en la raíz (C:\) y además mi paquete **"rfid.jar"** se encuentra en una carpeta con nombre **"ipaq"**, la línea de comando quedaría como sigue:

Para compilar:

```
javac -classpath aplicación;C:\commapi\comm.jar;C:\ipaq\rfid.jar *.java
```

Con el “*” digo que compile todos los archivos .java

Y para ejecutar:

```
java -classpath aplicación;C:\commapi\comm.jar;C:\ipaq\rfid.jar Interfaz
```

Referencias:

Web 1 SAMSYS, The power to choose (sitio Web. Octubre 29, 2004) Technology RFID. <http://www.samsys.com/default.php?alpha=rfid>

Web 2 Intermec Expect More. Líder Mundial en Captura de Datos (sitio Web. noviembre 11, 2004), Guide to RFID Tag Election.
http://epsfiles.intermec.com/eps_files/eps_spec/RFIDTagGuide_spec_WEB.pdf

Web 3 Intermec expect More. Líder Mundial en Captura de Datos (sitio Web. Agosto 23, 2004) Especificaciones de Productos RFID.
http://www.intermec.com/eprise/main/Intermec/Content/Products/Products_ListProducts?Category=RFID&Family=RFID2

Web 4 Robotiker Tecnalía. Centro Especializado en Tecnologías de la Información y Telecomunicaciones (sitio Web. Octubre 13 2004).Presentación de la Tecnología RFID.
http://www.robotiker.com/castellano/noticias/eventos_pdf/33/RBTK_Inigo.pdf

Web 5 Revista RED. La Comunidad de Expertos en Redes (sitio Web. Noviembre 10, 2004) RFID y el Retail.
<http://www.red.com.mx/scripts/redArtBusqueda.php3>

Web 6 RF-ID.com System Solution Company (sitio Web. Octubre 18, 2004) Tags and Reader RFID UHF.
<http://www.rf-id.com/rfidit.html>

Web 7 Texas Instruments (sitio Web. Octubre 7, 2004) Applications Wireless, RF identification. Reference Manual Tag-it Transponder Protocol. Especificaciones Marzo

2000 y Octubre 2001.

http://www.ti.com/tiris/docs/manuals/refmanuals/tag_it_transponder_protocol.pdf

Web 8 Capta. Soluciones Automatizadas para la seguridad de personas, activos y propiedad intelectual (sitio Web. Agosto 20, 2004). División RFID. Catalogo de Productos de Escort Memory System (EMS).

http://www.capta.com.mx/solucion/ems_rf_id_tags.htm

Web 9 Data Collection (sitio Web. Noviembre 07, 2004) Tags RFID para localizar personas en parques temáticos

Artículo hecho por la redacción.

http://dc.editricetemi.com/ed_temi/dcSp/articulo.jsp?idArt=19

Web 10 NEDAP Automatic Vehicle Identification (sitio Web. Noviembre 7, 2004)

Acceso de Vehículos.

http://www.google.com.mx/search?q=cache:eWzrY36fz5YJ:www.nedapavi.com/pdf/AVI_brochure_S.pdf+acceso+de+vehiculos+RFID+&hl=es

Web 11 SEEBURGER, Bussines Integration Solutions (sitio Web. Septiembre 01, 2004)

Qué es RFID?

http://www.seeburger.info/international/spain/estandards_rfid_i.htm

Web 12 Information and Privacy Commissioner (sitio Web. Octubre 29, 2004) Artículo, Tag, You're It: Privacy Implications of Radio Frequency Identification (RFID)

Technology. <http://www.ipc.on.ca/docs/rfid.pdf>

Web 13 IDTechEX Article (sitio Web. Noviembre 16, 2004) RFID Tag Type Options Tag choices.

<http://www.idtechex.com/products/en/article.asp?articleid=41>

Web 14 Gesimpex Comercial S.L. (sitio Web. Octubre 14, 2004) Sistemas de identificación Pasivos por Radio Frecuencia, División Ganadera, Transponder Pasivos RFID para Animales. <http://www.gesimpex.com/gesimpex/public/home.htm>

Web 15 Gesimpex Comercial S.L. (sitio Web. Octubre 14, 2004) Sistemas de identificación Pasivos por Radio Frecuencia, División Industrial, Tipos de Transponder Pasivos RFID. <http://www.gesimpex.com/gesimpex/public/home.htm>

Web 16 Escort Memory System, A Datalogic Group Company (sitio Web. Agosto 20, 2004), División RFID, What is RFID?. <http://www.ems-rfid.com/>

Web 17 Texas Instruments (sitio Web. Octubre 8, 2004), Division Wireless, Tag-it RFID Document Center. <http://www.ti.com/tiris/>

Web 18 UPM Rafsec (sitio Web. Octubre 18, 2004) Advantages of the UHF 862-928 MHz Frequencies.

http://www.rafsec.com/uhf_advantages.htm

Web 19 Syxpert, Ingeniería en Sistemas Expertos (sitio Web. Noviembre 1, 2004) Diagramas Técnicos de Monitoreo por RFID. <http://www.bugless-arg.com.ar/diagramas1.htm>

Web 20 Syscan Intenational Inc.(sitio Web. Nov 24, 2004) Lector RFID para iPAQ, Syscan RFID Reader with Compact Flash Type II. http://www.syscan.com/w4/Main_Readers_e.htm

Web 21 Freshmeat, (sitio Web. Abril 19, 2005), Aplicaciones RFID de Libre Distribución. <http://www.freshmeat.net>

Web 22 SourceForge (sitio Web. Abril 19,2005) Aplicaciones RFID de Libre Distribución. <http://www.SourceForge.net>

Art 1 Maino M., Mario, Dr., (M.V.; D. Ec. Agr.). Una nueva forma de identificación animal. Revista de extensión TECNOVET. Diciembre de 1995.

http://bellota.sisib.uchile.cl/Tecnovet/CDA/tecnovet_articulo/0,1409,SCID%253D9719%2526ISID%253D429,00.html

Noviembre 12, 2004.

Art 2 Cavoukian, Ana (Ph.D. Commissioner). Tag, You're It: Privacy Implications of Radio Frequency Identification (RFID) Technology. February 2004.

<http://www.ipc.on.ca/docs/rfid.pdf>. Enero 01, 2005.

Art 3 Sandoval Reyes, Sergio. Identificación de Objetos por Radio Frecuencia. Centro de Investigación en Computación, Instituto Politécnico Nacional. Agosto 31, 2004.

Art 4 Reigelsperger, Dan; Harkins, James. RFID Basic, Keeping Pace with RFID.

Lowry, Data Flow at lightning Speed.

<http://www.lowrycomputer.com/webinars/files/RFID.pdf>. Noviembre 16, 2004.

Art 5 Leiva Aguilera, Javier. RFID. Catalogo Catorze. Julio 22, 2003.

http://www.catorze.com/blog/articulos/2003_07.php

Septiembre 01, 2004.

Art 6 Malatto, L.; Lozano, A.; Milano, O.; Fraigi, L. Trazabilidad de Ganado. Saber Como, Instituto Nacional de Tecnologia Industrial. Agosto 19, 2004.

<http://www.inti.gov.ar/sabercomo/sc19/inti6.php>. Marzo 31, 2005.

Art 7 Daccach T. José Camilo. Impacto de Tecnologías Informáticas III. DELTA con Profundidad. Noviembre 2004. <http://deltaasesores.com/prof/PRO308.html> . Marzo 31, 2005.

Art 8 Martínez, Claudia. ¿Qué es RFID?. NetMedia, B:Secure. Diciembre 9, 2004. http://www.netmedia.info/bsecure/articulos.php?id_sec=51&id_art=5021 . Abril 4, 2005