

**IMPLEMENTACIÓN DE LA TECNOLOGÍA DE
IDENTIFICACIÓN AUTOMÁTICA DE OBJETOS
AUTO ID EN EL CONTEXTO DE LA CONSTRUCCIÓN**

**FABER ALBERTO MUÑOZ
JAVIER HERNÁN BURBANO NARVÁEZ**

**Documento final de Grado para optar al título de
Ingeniero en Electrónica y Telecomunicaciones**

**Director
GUSTAVO RAMIREZ
Ingeniero en Electrónica y Telecomunicaciones**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES
DEPARTAMENTO DE TELEMATICA
POPAYÁN
2006**

*De todo corazón agradezco al Todopoderoso,
y dedico enteramente este logro a mi familia,
especialmente a mi madre que ha sido un gran apoyo,
sin el cual hubiese sido imposible llegar hasta aquí.*

*Agradezco a mis amigos, que se preocuparon
y apoyaron en este proceso.*

*Un agradecimiento muy especial a:
Carlos Alfredo Narváez
por ser más que un gran amigo y colega.*

*Atentamente:
Javier Hernán Burbano Narváez*

*Este peddãe lo debe a muchas personas que me tendieron su ayuda desinteresada,
y siempre fueron una voz de aliento.*

*Atentamente:
Faber Alberto Muñoz*

AGRADECIMIENTOS

Los autores expresan sus agradecimientos:

- A GUSTAVO RAMÍREZ, Profesor del Departamento de Telemática de la Facultad de Ingeniería Electrónica y Director del proyecto. A quien debemos en gran parte el norte que nuestra vida tomó, además de servir de apoyo y respaldo sin el cual hubiese sido más difícil nuestro camino. Gracias por ser un ejemplo y por brindarnos su conocimiento, sabemos que Dios se lo recompensará.
- A DEPARTAMENTO DE TELEMATICA, Facultad de Ingeniería Electrónica.
- A Todas las personas que de una u otra forma contribuyeron, para la cristalización de este proyecto.

CONTENIDO

	Pág.
INTRODUCCIÓN	11
1. IDENTIFICACIÓN POR RADIOFRECUENCIA RFID	14
1.1. LA TECNOLOGÍA	14
1.2. COMPONENTES RFID	16
1.2.1. El Lector	17
1.2.2. Las Etiquetas o TAG	17
1.2.3. El Host	18
1.3. EL ESTANDAR EPC	19
1.3.1. ¿Qué es El EPC?	19
1.4. VENTAJAS DE RFID	21
1.4.1. Lectura y Escritura	21
1.4.2. Ausencia de una Línea de Visión Directa	21
1.4.3. Capacidad de Datos	22
1.4.4. Reutilización y Durabilidad	22
1.4.5. Lectura Múltiple	22
1.4.6. Robustez	23
1.4.7. Seguridad	23
1.4.8. Proceso de Marcación	24
1.5. DESVENTAJAS DEL RFID	24
1.5.1. Ambientes Metálicos o Materiales Conductores	24
1.5.2. Etiquetas en Presencia de Líquidos	25
1.5.3. Susceptibilidad a Interferencia Electromagnética	25
1.5.4. Captura Múltiple de Datos	25
1.5.5. Exposición Humana	26
1.5.6. Prueba de Lectura	27
2. EL SISTEMA DE RED EPC	28
2.1. DESCRIPCIÓN Y COMPONENTES CLAVES DEL SISTEMA	29
2.1.1. El Servicio de Nombres de Objetos (ONS)	30
2.1.2. Lenguaje de Marcación Físico (PML)	31
2.2. ARQUITECTURA DEL SISTEMA EPC	31
2.2.1. La Arquitectura de Software de el Sistema EPC	33
2.2.2. El Savant o Software Mediador	35
2.2.3. Servicio de información PML y EPC	39
2.2.4. Servicio de Información EPC	41
2.3. ARQUITECTURA AUTO ID EN LA CONSTRUCCIÓN	42
2.4. EL ESPECTRO DE RADIO FRECUENCIA Y LA RED EPC	44
3. DESCRIPCIÓN DE LA SOLUCIÓN DESARROLLADA	47
3.1. DESCRIPCIÓN DE LA APLICACIÓN	48
3.1.1. Lectores	50
3.1.2. MEPCC	51

3.1.3.	ONS	52
3.1.4.	Servicio de Información EPC.....	52
3.1.5.	Aplicación Empresarial	53
3.2.	EJECUCIÓN DEL PROYECTO	53
3.2.1.	Componentes J2EE para la Aplicación	54
3.2.1.1.	Cliente J2EE MEPCC	55
3.2.1.2.	Servidor J2EE Componente Web	56
3.2.1.3.	Servidor J2EE Componente de Negocio	57
3.2.1.4.	Capa EIS (Sistema de Información Empresarial)	58
3.2.2.	Aplicación J2EE Proyecto ServidorPML	58
3.2.2.1.	Base de Datos EPC Sistema de Información Empresarial	60
3.2.2.2.	Bean de Sesión sin Estado ServidorPMLAccessBean.....	61
3.2.2.3.	La Interface DAO	67
3.2.2.4.	El Método de Negocio IDs	69
3.2.2.5.	Implementación DAO.....	70
3.2.2.6.	Métodos de Retrollamada.....	71
3.2.2.7.	Servicio Web.....	72
3.2.2.7.1.	El cliente Web.....	76
3.2.2.8.	Aplicación Cliente.....	78
3.2.2.8.1.	TEXAS INSTRUMENTS TECHNOLOGY.....	78
4.	RECOMENDACIONES Y CONCLUSIONES.....	82
	REFERENCIAS	85
	BIBLIOGRAFIA	86

LISTA DE FIGURAS

		Pág.
Figura 1.	Componentes RFID.....	16
Figura 2.	Etiquetas o TAGS.....	18
Figura 3.	EPC Clase1 versión de 96 bits.....	20
Figura 4.	Arquitectura de la red EPC.....	33
Figura 5.	Arquitectura AUTO ID en el contexto de la construcción.....	43
Figura 6.	Espectro de frecuencias alrededor del Mundo.....	45
Figura 7.	Ventajas y desventajas de la banda de frecuencias.....	46
Figura 8.	La Red EPC y el MEPCC.....	49
Figura 9.	Aplicación J2EE Arquitectura.....	55
Figura 10.	Aplicación J2EE ServidorPML.....	60
Figura 11.	Base de datos EPC Sistema de Información empresarial.....	61
Figura 12.	Archivos de configuración generados por Lombos.....	63
Figura 13.	Archivo EjbGenerate.xml.....	64
Figura 14.	Archivo EjbGenerate.xml etiquetas Jboss.....	65
Figura 15.	Archivo EjbGenerate.xml etiqueta Dao.....	66
Figura 16.	Archivo.ejbGenerate.xml etiquetas Data.....	67
Figura 17.	Interface DAO.....	68
Figura 18.	Clases Ejbs generadas en el paquete sesión.....	.69
Figura.19.	Archivo del paquete WebService generados.....76

Figura 20	La clase Cliente.....	77
Figura 21	El menú Comunicaciones.....	79
Figura 22	El menú Leer UID.....	80
Figura 23	El menú Programar UID.....	80

ANEXOS

- ANEXO A. CONFIGURACIONES
- ANEXO B. MANUAL DE USUARIO
- ANEXO C. MODELADO
- ANEXO D. CODIGO FUENTE (Formato Digital)

RESUMEN

La Identificación por radio frecuencia (RFID), es relativamente una tecnología antigua, alrededor de unos 60 años 1950, donde se realizaron las primeras aplicaciones orientadas a la identificación de vehículos, que esta ganando una enorme popularidad en últimos años, cada vez más en el área de los negocios como una posible tecnología que puede mejorar los procesos existentes. Se presenta a menudo hoy en día como un reemplazo del código de barras, pero esta tecnología tiene muchas más posibilidades, tal como un número serial individual para cada artículo, y la posibilidad de leer este número a varios metros de distancia, no hay necesidad de línea de vista ni contacto entre el lector y la tarjeta.

Esta tecnología tiene un campo de aplicación multisectorial entre el que se ha localizado el sector de la construcción. Una obra cambia todo el tiempo, y se desarrolla en varios sitios de manera muy diferentes. Las áreas de almacenamiento cambian durante el proceso de construcción, y los materiales tal vez no estén donde el trabajador los necesite. Implicando tanto perdida de tiempo como dinero al incurrir en la búsqueda. Adicionalmente hay toneladas de diferentes materiales llegando al sitio todo el tiempo. La presión ejercida por el tiempo a menudo significa que haya poco control o que este se realice informalmente. Los materiales son desplazados

exactamente al punto en donde se piensa serán utilizados. Con el tiempo se descubren errores, los materiales son a menudo desempacados y ubicados en sitios no adecuados de la construcción. Esto provoca una difícil reposición o devolución del material y pérdida de tiempo. Si los materiales estuvieran marcados con etiquetas RFID, estas fallas en la entrega quizás se descubrirían cuando los materiales arribaran al lugar de la construcción.

En este trabajo titulado “Implementación de la Tecnología de Identificación Automática de Objetos Auto ID en el Contexto de la Construcción”, para optar al título de Ingeniero en Electrónica y Telecomunicaciones de la Universidad del Cauca. Se consigna la investigación llevada a cabo sobre una posible implementación de la tecnología de identificación automática de objetos por radio frecuencia “RFID”, en el sector de la construcción, además de describir el diseño de una herramienta software de apoyo para su explicación basada en la red EPC/RFID. Iniciativa que busca la automatización de la cadena de suministro. El Software de Apoyo desarrollado bajo plataforma J2EE permite: leer y escribir información en una etiqueta o Tag de RFID, además de manejar en pequeña versión un sistema información del contexto en que trabaja. La acción de leer esta asociada a la consulta en línea de las características del artículo que tiene asociada la Tag en cuestión. La acción de escribir tiene la función de dejar una señal en el Tag para una posterior consulta que ayuda a una correcta logística al interior de grandes construcciones.

INTRODUCCIÓN

El objetivo fué realizar una investigación y desarrollar una aplicación software de apoyo para evaluar la posible implementación de identificación de activos en la construcción, aplicando la tecnología de identificación automática de objetos basada en la propuesta AUTO ID EPC/RFID. Esta propuesta y tecnología de identificación automática esta enfocada a proporcionar una posibilidad de seguimiento y rastreo de elementos dentro de una cadena de procesos, permitiendo verdaderas ventajas competitivas con respecto a otras tecnologías. Se presenta a continuación la información detallada sobre el desarrollo del proyecto “Implementación de la Tecnología de Identificación Automática de Objetos Auto ID en el Contexto de la Construcción”

En el trabajo de investigación se cumplieron con los objetivos al desarrollar una aplicación para la identificación de objetos en la construcción aplicando la tecnología de identificación automática de objetos basada en la propuesta de EPC/RFID, se generó un software de apoyo para dar solución al problema y cumplir con los requerimientos de la nueva red, que permite leer y escribir datos en las etiquetas mediante el uso del protocolo Adecuado (EPC).

El desarrollo de este proyecto se dividió en tres fases consignadas en tres capítulos de este escrito y un cuarto capítulo en el que se mencionan algunas conclusiones, recomendaciones, Se describe a continuación el desarrollo del proyecto capítulo a capítulo:

Inicialmente se desarrollo un trabajo de investigación y capacitación en los temas relacionados con la tecnología RFID, En el primer Capítulo, se da una descripción de la tecnología RFID, sus componentes, variaciones, estándares, ventajas, desventajas y su futuro, lo que permite ubicarnos dentro del entorno de esta tecnología.

En el segundo capítulo, se explica la tecnología AUTO-ID EPC/RFID, la arquitectura para habilitar la nueva red de EPC (código de producto electrónico), componentes del sistema, lenguaje PML, software del sistema, además una descripción del caso de estudio relacionado con la construcción y consideraciones generales de la aplicación.

En el tercer capítulo, se describe la solución al problema planteado, su funcionalidad y ubicación dentro de esta arquitectura, se presenta el desarrollo del software Mediador EPC para la construcción (MEPCC) que fue desarrollado con una herramienta de entorno visual de grandes prestaciones utilizando el lenguaje Java y la interfaz de desarrollo IDE Eclipse, logrando ofrecer una velocidad adecuada para interactuar con el sistema hardware (lector), una interfaz amigable e intuitiva para un fácil manejo, además de la interacción con la red EPC supuesta y con el sistemas de control de obra representado por una base de datos en el sistema cliente.

También contendrá las diferentes decisiones de diseño, implementación y desarrollo tomadas en la realización del proyecto. Esto involucra selección de plataformas, modelos descriptivos, diagramas correspondientes, tecnologías empleadas, etc.

El cuarto capítulo, presentará las Conclusiones y Recomendaciones respectivas, producto de la realización del proyecto, tanto a nivel del desarrollo como a nivel teórico. De igual manera contiene las recomendaciones producto de la experiencia adquirida, que puedan ser aplicadas en el desarrollo de trabajos posteriores.

Por último se hace referencia de los documentos que fueron consultados como fuente en el desarrollo del proyecto, así como una lista de términos a tener en cuenta para un mejor entendimiento de la documentación. Se incluyen los Anexos: Anexo de configuraciones, manual de usuario, modelado y código fuente, los cuales son la documentación generada para profundizar o esclarecer un tema específico.

Con la presente monografía, se pretende dejar una síntesis del trabajo desarrollado y las experiencias adquiridas durante su ejecución. Con el objeto de que las personas en un futuro, con proyectos relacionados con RFID cuente con una documentación relativamente amplia y de fácil comprensión.

1. IDENTIFICACIÓN POR RADIOFRECUENCIA RFID

La RFID utiliza etiquetas llamadas TAG que permiten reflejar ondas de radiofrecuencia de manera que las ondas devueltas cargan datos. RFID es clasificado como una tecnología de Captura Automática de Datos – ADC, al igual que el código de barras, el reconocimiento óptico de caracteres, el reconocimiento de formas y las tarjetas de cinta magnética, etc. [1]

1.1. LA TECNOLOGÍA

Un sistema de identificación automatizada que utiliza RFID tiene un lector, también llamado interrogador, el cual transmite una señal a cierta frecuencia a todas las etiquetas de RFID dentro de su rango, a su vez las etiquetas de RFID devuelven la señal al lector. Cuando las etiquetas son “activadas” por el lector, se establece un “diálogo”, de acuerdo con un protocolo de comunicación predefinido para el intercambio de datos.

El sistema de RFID puede operar en bajas frecuencias LF (9 KHz a 135 KHz) o altas frecuencias HF (13.56 Mhz), al igual que en UHF (300 a 1200 Mhz) y Microondas a 2.4 Ghz (Para Tags Activas). El lector y la etiqueta de RFID se comunican por medio

de campos electromagnéticos creados entre la antena del lector y la etiqueta de RFID.

El campo electromagnético, suministra el poder que activa la etiqueta. El circuito integrado que está inmerso en la etiqueta de RFID ejecuta los programas para el cual ha sido diseñado. El circuito integrado de la etiqueta RFID transmite información por amplitud, frecuencia o por modulación de fase de la onda portadora. Existen etiquetas pasivas y activas, la diferencia radica en que las primeras no tiene alimentación propia mientras que las segundas tienen una batería que les permite mayor alcance al generar un mayor campo magnético.

El lector recibe las señales y las transforma en datos binarios (0 o 1). De igual manera, el lector modula la señal portadora para enviar datos a la etiqueta RFID donde la señal portadora es reconvertida por el circuito integrado de la etiqueta de RFID en información digital, dicho en otras palabras, entre el lector y la etiqueta de RFID se lleva a cabo un proceso de modulación y demodulación de la señal portadora de la información. Una característica de este principio operativo es que a menor frecuencia operativa, mayor debe ser la antena requerida en la etiqueta de RFID. Como resultado de lo anterior, bajar frecuencias de operación aumenta la complejidad de la fabricación. La etiqueta de RFID puede adherirse o ser llevada en un objeto. La palabra "objeto" es usada en su más amplio sentido: Puede ser una tarjeta inteligente (tarjeta telefónica, bancaria), un paquete, un vehículo, etc. [1]

1.2. COMPONENTES RFID

Las aplicaciones de RFID necesitan ciertos componentes básicos, ver Figura 1.

- La etiqueta o TAG de RFID.
- El lector (estático o portátil).
- El Host .

Estos componentes se comunican por medio de dos interfaces:

- La interface aérea entre la etiqueta de RFID y el lector. Este es un enlace radial con sus propias características físicas, en especial la frecuencia y el protocolo de las comunicaciones.
- El enlace local entre el lector y los sistemas de computador que usan los datos de la etiqueta RFID como parte de un sistema centralizado de distribución.[1]

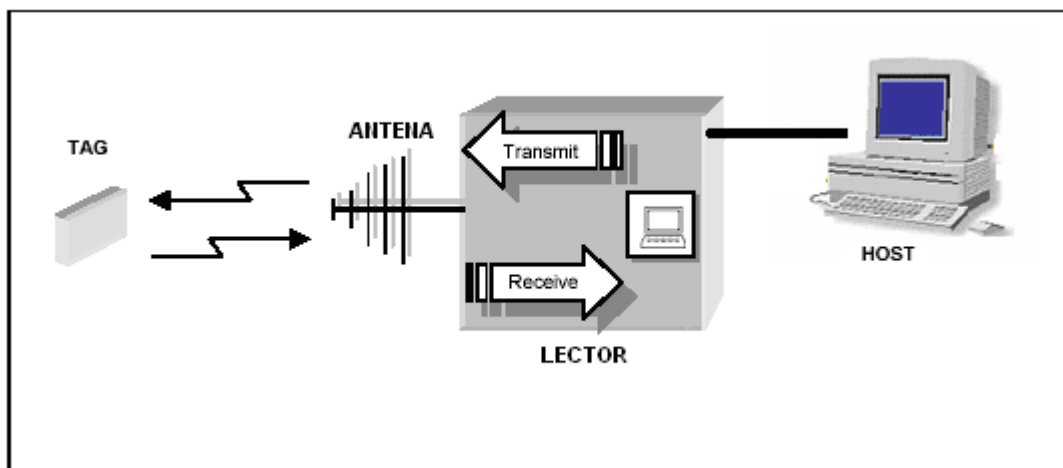


Figura 1 Componentes RFID

1.2.1. El Lector

El lector se compone de una antena y un controlador de comunicaciones. El controlador tiene tres funciones principales:

- Una función es la de codificar y descodificar los datos, verifica los datos, almacena los datos y se comunica con el servidor;
- La responsabilidad de activar la etiqueta de RFID, iniciar una sesión, leer y escribir los datos, autorización, etc.
- Comunicaciones físicas, la frecuencia, la tasa de datos, modulación.

La antena emite ondas de RF al medio después de ser generadas dentro del transmisor. Esta misma antena recibe respuesta de las etiquetas solo en el rango de acción de su campo magnético. Por lo general los lectores usan una o más antenas dependiendo de su función. De esta manera se puede lograr arreglos de antenas y aumentar la cobertura. [1]

1.2.2. La Etiqueta o TAG

La etiqueta de RFID usualmente tiene: un circuito integrado unido a una antena y una pila (para etiquetas activas). Las etiquetas de RFID sin pila son llamadas etiquetas pasivas. En ese caso, la energía requerida para el circuito integrado de la etiqueta RFID es generada por un campo electromagnético suministrado por el

lector. Las etiquetas de RFID con pilas (etiquetas RFID activas), pueden usarse en frecuencias de microondas para comunicaciones de larga distancia (Figura 2). [1]



Figura 2 Etiquetas o TAGS

1.2.3. El Host

El Host de un sistema de RFID es el centro de la aplicación. Éste es responsable de enviar y recibir mensajes desde y hacia el lector con el fin de: manipular, configurar y capturar los datos, además es el punto de conexión hacia cualquier tipo de red integrando de esta forma el lector de RFID a los diferentes sistemas de información. El sistema Host está conectado al lector por un enlace de comunicación local estándar este puede ser RS232, RS485 o RJ45.

1.3. EL ESTANDAR EPC

Siempre se ha visto la necesidad de estandarizar los diferentes tipos de codificación, sobre todo cuando se habla, de tecnologías de Captura Automática de Datos – ADC como el código de barras, la idea es que todas las compañías tengan un entendimiento común de su significado. Es fundamental que la tecnología RFID cuente también con una definición de un estándar que permita que su uso sea generalizado y global.

En el proceso de estandarización para RFID han participado múltiples entidades, jugando un papel especial los fundadores del AUTO ID Center en el MIT (Massachusetts Institute of Technology), organismo promovido por la UCC (Uniform Code Council), el cual cuenta con la participación y el apoyo de la EAN (European Article Number) internacional que es la organización mundial de estándares y cerca de 90 empresas entre las cuales se cuentan las más importantes compañías de productos de consumo masivo y de soluciones tecnológicas.

El AUTO ID ha desarrollado lo que se conoce como EPC (Código Electrónico de Producto) el cual se promueve como el estándar para RFID. En nuestro país lidera el tema de la estandarización la IAC miembro de la EAN Internacional. [2]

1.3.1. ¿Qué es EPC?

La tecnología EPC se desarrollo con base en los principios de la comunicación a través de frecuencia de radio, incorporando a dicha tecnología los principios de

identificación única para cada producto y la simplificación de la información contenida en la etiqueta (TAG). [3]

El EPC contendrá la misma información que esta en los códigos de barras, más otros datos adicionales que permiten identificar cada unidad de producto o cada ítem de manera única (Figura 3).

En esencia el TAG de radiofrecuencia contiene una identificación única de 64, 96 128 ó 256 bits, que al igual que en la tecnología de código de barras contiene en su estructura la identificación del fabricante, del producto y adicionalmente el número seriado del ítem, lo cual brinda una identificación única para este producto en todo el mundo. [2]



Figura 3 EPC Clase1 Versión de 96 bits.

Con el código ejemplo EPC Clase1 de la figura 3, 16 millones de compañías pueden categorizar 16 millones de productos c/u y cada categoría de producto puede contener 1099 billones de ítem individuales. [3]

1.4. VENTAJAS DE RFID

Siendo el código de barras la tecnología universal usada para la identificación y la captura automática de datos. Las ventajas y desventajas de las etiquetas de RFID están, por lo tanto, descritas en relación con los códigos de barras y son las siguientes:

1.4.1. Lectura y Escritura

A diferencia de los códigos de barras donde los datos se fijan una vez han sido impresos, los datos almacenados en una etiqueta de RFID pueden modificarse, extenderse o reducirse por usuarios autorizados (únicamente en etiquetas con la facilidad de lectura -escritura). Las etiquetas de RFID pueden incorporarse por ejemplo en las estibas o en el empaque original. [1]

1.4.2. Ausencia de una Línea de Visión Directa

Para permitir que las etiquetas de códigos de barras se lean automáticamente, los organismos de estandarización, tales como EAN y UCC, han definido unas reglas para la ubicación de etiquetas en los productos y las unidades logísticas. Sin embargo, las etiquetas de RFID no están sujetas a las mismas restricciones de visibilidad y ubicación.

Lo único requerido para detectar una etiqueta RFID es ésta debe estar dentro del campo de incidencia del lector y no estar bloqueada por metales o líquidos. [1]

1.4.3. Capacidad de Datos

Teóricamente no existe límite para la capacidad de datos almacenados en una etiqueta de RFID, considerando que en los códigos de barras lineales la cantidad de datos máxima es de 50 bytes, y en los códigos de barras en 2 dimensiones pueden tener hasta 3000 bytes. Una etiqueta de RFID potencialmente podría almacenar hasta varios megabytes de información. El incremento en la capacidad de datos aumentará el costo y reducirá el rango de la operación, manteniendo todo lo demás estático. [1]

1.4.4. Reutilización y Durabilidad

En teoría no existe un límite sobre cuantas veces se puede leer una etiqueta de RFID. Es de consenso común que para las aplicaciones donde el mismo artículo puede ser usado varias veces, por ejemplo para Identificar estibas o contenedores retornables, una etiqueta RFID puede durar hasta 10 años. [1]

1.4.5. Lectura Múltiple

Un lector, que usa un algoritmo anticolidión, puede leer varias etiquetas que están en el campo del lector al mismo tiempo. Los datos son recuperados de una etiqueta a la vez, por lo tanto, el tiempo para leer todas las etiquetas en el campo será mayor, mientras mayor sea el número de etiquetas que se van a leer en el campo. Los algoritmos anticolidión más efectivos tienen tiempos de lectura que aumentan

linealmente con el número de etiquetas. Los algoritmos anticolidión menos eficientes tienen tiempos de lectura que aumentan exponencialmente con el número de etiquetas. [1]

1.4.6. Robustez

Las etiquetas de RFID no necesitan estar en la parte exterior del artículo que se va a identificar, por lo tanto, pueden estar protegidas durante el almacenamiento, manejo o transporte. Los principios operativos las hacen inmunes al mugre o rasgaduras que si pueden afectar el uso de los códigos de barras.

En general, la robustez depende de la forma como el artículo es empacado. El empaque debe ser adaptado teniendo en cuenta su ambiente operativo. [1]

1.4.7. Seguridad

Como cualquier medio digital, una etiqueta de RFID puede estar protegida por medio de una clave tanto para la lectura como para la escritura. Los datos pueden estar encriptados. Una sola etiqueta de RFID puede contener datos protegidos así como datos desprotegidos. Estas características de seguridad pueden ser útiles en aplicaciones contra robo y la piratería. [1]

1.4.8. Proceso de Marcación

Los códigos de barras usados en la distribución generalmente requieren que éste sea impreso en una etiqueta de papel. El manejo y la aplicación de la etiqueta todavía son una operación manual o mecánica, y esto es lo mismo en las etiquetas de RFID. Sin embargo, la diferencia radica en que la información sobre la unidad de logística se puede añadir en cualquier punto una vez se conozca, quedando como un proceso completamente automatizado. La característica de escritura puede permitir que artículos retornables (estibas, canastillas, bidones, etc.) y empaques sean fabricados con etiquetas de RFID y por lo tanto se reducirá la necesidad de aplicaciones manuales. [1]

1.5. DESVENTAJAS DEL RFID

Las ventajas explicadas anteriormente tienen ciertas desventajas:

1.5.1. Ambientes Metálicos o Materiales Conductores

Las etiquetas de RFID pueden ser afectadas por la presencia cercana de metales, antes de que puedan ser usadas, generalmente se debe modificar el empaque de ciertos productos, especialmente aquellos que van en contenedores metálicos. Aun los empaques que contienen metal enterrado pueden afectar las etiquetas de RFID, esto se aplica a los empaques de líquidos en donde hay una capa delgada de película metálica para asegurar que el empaque esté sellado. Este problema ocurre

con todas las frecuencias, cuando se aplica una solución aumenta bastante el costo de la etiqueta, ya que la etiqueta de empaque debe ser construida para aislar de manera efectiva la etiqueta del metal. [1]

1.5.2. Etiquetas en Presencia de Líquidos

La lectura de las etiquetas de RFID también puede verse afectada por líquidos en el producto. Si existe algún líquido entre el lector y la etiqueta, la lectura se podría dificultar o ser imposible en algunos casos. Las características de la etiqueta pueden ser alteradas por el líquido y reducir el rango de lectura o eliminar la posibilidad de lectura. [1]

1.5.3. Susceptibilidad a Interferencia Electromagnética

Los sistemas de RFID pueden ser susceptibles a EMI (Interferencia Electromagnética) de los computadores (monitores), luces, otros sistemas que usan la misma frecuencia (RFID o Bluetooth a la misma frecuencia), y otro tipo de diferentes generadores EMI, lo anterior implica que se debe evaluar el ambiente en el cual se van a usar. [1]

1.5.4. Captura Múltiple de Datos

En muchas aplicaciones, diferentes etiquetas de RFID pueden estar presentes dentro del campo del lector al mismo tiempo. Los lectores diseñados para manejar varias etiquetas de RFID simultáneamente usan algoritmos o técnicas anticolidión.

La anticolisión se refiere al número de etiquetas que pueden ser leídas en un área radial o de difusión, al usar las técnicas de anticolisión es posible asegurar que un tipo de etiqueta es leído en determinado momento y no todas al tiempo. Por ejemplo, si existe una estiba con una etiqueta impresa, y las cajas en la estiba y los artículos dentro de las cajas también tienen etiquetas similares, es necesario asegurar que solamente se lee la etiqueta con la información requerida. [1]

Aun cuando estas técnicas han sido demostradas con éxito, en la práctica las técnicas anticolisión no son todavía totalmente confiables y pueden extender considerablemente el tiempo de lectura. [1]

1.5.5. Exposición Humana

El cuestionamiento acerca del efecto de las emisiones de radio frecuencia en la salud ha sido un tópico discutido por varios años, especialmente en relación con las puertas de detección de robo y teléfonos celulares. Las etiquetas de RFID en sí no presentan ningún riesgo por mas comunes que se vuelvan, ya que están inactivas por lo menos el 99.99% del tiempo. Los lectores, por otro lado, son el tema de estudio, los cuales tienden a definir el criterio para el control del nivel de las emisiones para evitar equipos poco tranquilizadores en cuanto a salud se refiere, tales como marca pasos así como para prevenir riesgos en la salud.

Los comités de estandarización para los sistemas de comunicaciones en Europa, los Estados Unidos y Japón están trabajando en la estandarización de los niveles de poder máximo mundialmente permitidos. [1]

1.5.6. Prueba de Lectura

Aún cuando la capacidad de leer varias etiquetas es una ventaja, también puede ser considerado como una desventaja. Es en realidad muy difícil tocar este tema de prueba de lectura, por ejemplo, ¿han sido leídas adecuadamente todas las etiquetas presentes en el campo del lector?. Ante el cuestionamiento anterior se puede responder que depende de muchos parámetros o factores como: la ubicación de la tag, la interferencia, la profundidad de penetración de la onda de radiofrecuencia asociada a la potencia de emisión de la misma etc.

2. EL SISTEMA DE RED EPC

El código de producto electrónico (EPC) y la red (o sistema) de EPC están creados para ayudar a los negocios para mejorar la visibilidad de los productos y ayudar a mantener la seguridad e integridad del producto a través de la cadena de suministro. Para las compañías no es necesario solamente saber donde están sus productos, también necesitan compartir esa información con sus socios para procesos de negocio más seguros y eficientes. [3]

La Red de EPC permite a socios de negocios seguir y rastrear artículos automáticamente a través de la cadena de suministro. Esto provee a los negocios de una visión en tiempo real sin precedente de sus activos e inventarios dondequiera que estén, trayendo consigo beneficios significativos en operaciones y protección de marcas.

La red EPC provee ventajas más allá de eficacias operacionales permitiendo cadenas de abastecimiento y suministro más seguros y confiables con aplicaciones contra falsificación, sabotaje, terrorismo, conformidad con regulaciones y otras.

Este capítulo describe la arquitectura para habilitar la red de EPC. La arquitectura esta específicamente diseñada para su puesta en práctica en grande en las

empresas que necesitan integrar datos en tiempo real que fluyen adentro de procesos existentes del negocio y de usar el back-end de los sistemas de la empresa usando Tecnologías automáticas de identificación (Auto - ID) tales como etiquetas de identificación por radiofrecuencia (RFID). [4]

El centro Auto-ID es un desarrollo del Instituto tecnológico de Massachussets como laboratorio patrocinador del desarrollo de tecnologías de identificación automática. Este ha desarrollado infraestructura, ha recomendado estándares y ha identificado diversas aplicaciones de identificación automática para formar una red de objetos físicos

En su visión del futuro, determina que prácticamente todos los dispositivos electrónicos están integrados en una sola red y están electrónicamente etiquetados con información pertinente sobre sí mismo. Al alcanzar esto, se obtienen grandes beneficios en diversas áreas incluyendo la cadena de abastecimiento y suministro y el inventario, control, seguimiento y localización de productos. [5]

2.1. DESCRIPCIÓN Y COMPONENTES CLAVES DEL SISTEMA

Una red de las proporciones propuesta por el AUTO ID center, requiere un sistema que sea capaz de manejar todos los objetos que se comercian en el mundo. Algunas de las características de este sistema deberían ser:

- Una arquitectura del sistema simple y abierta.
- La arquitectura del sistema debe de ser independiente de plataforma y altamente interoperable.

- Mantener independencia de la tecnología de etiquetado de producto.
- La arquitectura abierta debe de ser flexible y adaptable para cambiar y mejorar mucho tiempo después de ser instalada.

La arquitectura propuesta ostenta estas características y contiene los siguientes elementos:

- El código EPC (Dato de la Etiqueta)
- La Etiqueta física
- Tecnología De Red Local
- El servicio de nombre de objetos (ONS)
- Lenguaje de marcación Físico (PML)
- Servidores PML.

Del código EPC, la etiqueta y la tecnología de red ya se hablo en este documento.

Pasamos a definir el resto de componentes. [6]

2.1.1. El Servicio de Nombres de Objetos (ONS)

Teniendo en cuenta que el dato EPC es el único dato a ser almacenado en la etiqueta, este debe poder usarse para obtener mayor información sobre el producto. Esta información adicional puede ser almacenada en servidores que pueden ser conectados redes locales o a Internet. El servidor es identificado como un sistema conocido como ONS servidor de nombre de objetos. Este es análogo al DNS o servicio de nombrado de dominio, usado para localizar recursos de información en

Internet. El ONS es el directorio de servicios para conectar el objeto físico y la información sobre sí mismo. [7]

2.1.2. Lenguaje de Marcación Físico (PML)

Los archivos PML están almacenados en servidores especiales llamados. Servidores PML. El fabricante de un objeto mantendrá un servidor PML para todos sus objetos. El servidor PML almacenara una variedad de información como: datos de clases (como el nombre del producto), datos de instancia (como al fecha de fabricación), datos de distribución (como donde esta localizado); e instrucciones y datos entendibles por una máquina o algoritmos (como tiempo de cocción en un microondas), para este ultimo ejemplo no se conoce aplicaciones reales solo es un uso potencial que se le puede dar a la información que trae asociada la tag.

La razón de la clasificación de información es debido a que la implementación de cada una de estos almacenes de información puede ser diferente en eficiencia y comprensión. Una clase de información puede ser común y ser almacenada de manera estándar. Una instancia de información es específica y además muy numerosa. [6]

2.2. ARQUITECTURA DEL SISTEMA EPC

De la anterior versión genérica del sistema EPC se ha evolucionado hasta la propuesta actual que hace un refinamiento de los componentes genéricos.

La siguiente es la visión general de lo que es el sistema EPC. En esta se distinguen dos elementos claves:

- La arquitectura de software del sistema
- Los elementos de datos estándar.

La arquitectura de software del sistema o red EPC esta constituida por:

- Por los lectores.
- El savant (o software de mediación)
- El servicio de información EPC
- El servicio de nombrado de objetos (ONS).

A su vez los datos estándar del sistema EPC son:

- El código electrónico de producto EPC
- El lenguaje de marcación de objetos PML

La actual propuesta amplia los componentes haciendo más énfasis en lo que será la infraestructura lógica que hará posible la red EPC. Es así como se transforma el concepto de servidor PML por servicios más amplios en el servicio de información EPC, al igual que se le da importancia al elemento de mediación entre los lectores, el sistema EPC como tal y los sistemas empresariales actuales, el cual es el Savant actualmente llamado como software de mediación. [4]

2.2.1. La Arquitectura de Software de el Sistema EPC

La figura 4 muestra los componentes a alto nivel de la red EPC.

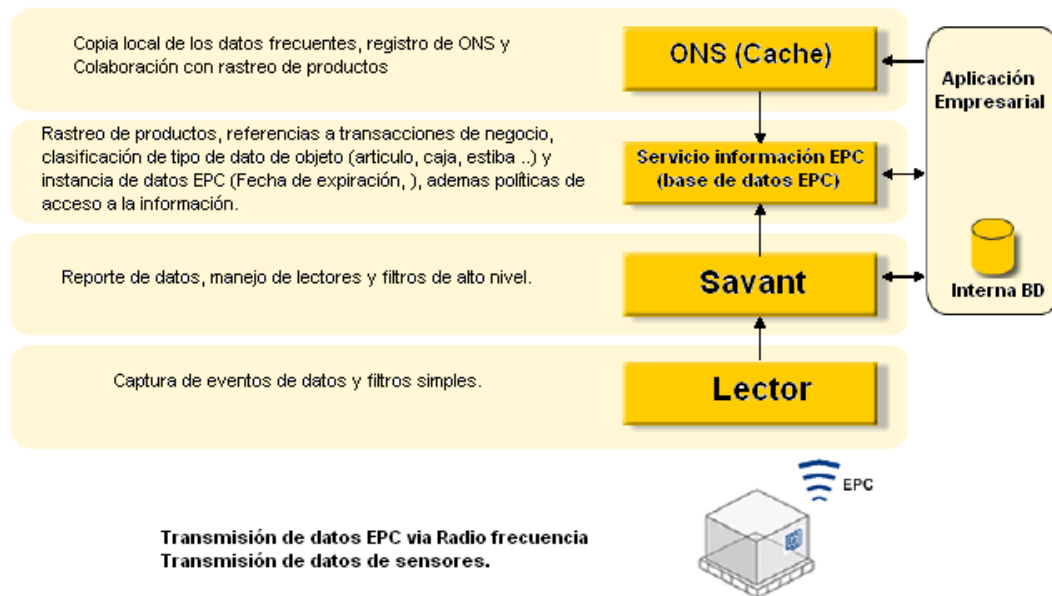


Figura 4 Arquitectura de la red EPC

En la parte baja están los lectores o los sensores de las etiquetas que son responsables de la lectura de etiquetas en los artículos, los cuales pueden estar en un estante o pueden moverse a través de un portal de lectores.

Cada lector lee muchos artículos marcados con la etiqueta y envía continuamente esos datos al software mediador (Savant) para ser procesados. Para cada lector, el rendimiento de procesamiento típico es aproximadamente 1000 lecturas por segundo. Esta diseñado para procesar flujos de datos de etiquetas o de datos de sensores (datos de eventos) que vienen de uno o más dispositivos del lector. El Savant tiene la capacidad de filtrar y/o priorizar los flujos de datos. Por ejemplo, un

objeto etiquetado delante de un lector sin moverse genera muchas lecturas redundantes.

El filtro del manejador de eventos se puede programar para hacer a un lado cualquier dato que demuestre que un mismo objeto está situado en un mismo lugar, y disparar una acción o evento solo cuando el objeto cambie de lugar. El manejador de eventos se puede configurar para estar acorde a las reglas del negocio en particular. Una vez logrado el respectivo filtraje los datos relevantes se trasladan a las otras capas.

Para organizar el tráfico del lector, una empresa puede tener instancias varias del *savant* para gestionar sitios geográficamente alejados, tal como un almacén, un centro de distribución, o bodega.

Un almacén típico es probable que tenga muchos lectores. Dado la cantidad de tráfico de la red de lectores, es importante localizar y filtrar los datos, esto lo realizan los servidores de manejo de eventos, en vez de enviar todo el tráfico sobre el Internet. Adicionalmente, es bueno aislar a los lectores de Internet por razones de seguridad.

La tercera capa en la arquitectura es el servicio de información de EPC. Se sugiere se utilicen esta capa para conectar la capa mediadora de eventos EPC con los sistemas de información Empresariales que ya existen, los sistemas de planeación de recursos (ERP), sistemas de gestión de almacenamiento (WMS), gestión de la cadena de suministro (SCM), y gestión de relación con clientes (CRM), así como

otros usos que se puedan dar a la información de la etiqueta. Como parte de la arquitectura de red EPC, está planeado que habrá servidores de información de EPC (IS) que permiten mantener y diseminar la información del producto en el formato de marcación físico (PML).

En la parte superior se encuentra el ONS local, el cual se conectará al ONS y contendrá una copia local de los datos ONS usados. Hará el registro de otros ONS y colaborara brindando información en tareas de trazabilidad.

De manera vertical encontramos los sistemas empresariales tales como ERP, WMS, sistemas heredados, y sistemas propietarios. Éstos deben aceptar e integrar datos y los eventos recibidos sobre objetos etiquetados. [5]

2.2.2. El Savant o Software Mediador

El SAVANT es el componente clave de de EPC. Actualmente se tiene la versión 1,0 de los estándares de Savant, que fueron desarrollados por parte del Grupo de acción de software del Auto ID center.

El manejador de eventos de EPC se ajusta a la especificación básica 1.0 y proporciona características y funcionalidad adicionales que se diseñan específicamente para implementación a gran escala.

El Savant es sobre todo responsable de procesar datos de etiquetas de RFID con un código EPC único, este número describe el fabricante de un objeto marcado con etiqueta, el tipo del producto, y el número de serie.

El Savant proporciona las ventajas siguientes:

- Proporciona un interfaz que permita a lectores de RFID y a otros dispositivos o sensores de la red ser conectado con la red de EPC.
- Ayuda a integrar datos y eventos RFID con los sistemas empresariales definiendo un sistema de interfaces que facilitan el envío y recepción de datos en tiempo real hacia y desde estos sistemas.
- Proporciona un sistema de enrutamiento de eventos genérico multipropósito.

Esencialmente, el Savant es colector y router de datos que realiza operaciones tales como captura, supervisión y transmisión de datos. Para cada lectura, el Savant recolecta un mínimo de información tal como los códigos EPC, la identificación del lector que leyó la etiqueta, y el instante de lectura. Los requisitos específicos para el proceso de EPC varían según cada caso o aplicación, por eso el Savant se define en términos de módulos de procesamiento o servicios, cada uno de los cuales proporcionan un sistema específico de características y puede ser combinado para tratar requisitos específicos de la aplicación. El énfasis en la versión 1,0 de la especificación del Savant está más en su capacidad de ampliación que en su procesamiento específico, este define solamente los módulos de proceso más básicos, y presenta a un marco dentro del cual módulos de procesamiento definidos por el usuario pueden funcionar.

Esta arquitectura modular facilita la innovación sin contrariar alguna especificación monolítica a la cual procura satisfacer requisitos generales.

El Savant esta diseñado para el desarrollo flexible de capacidades sin sacrificar disponibilidad, escalabilidad (tanto horizontal como vertical) o flexibilidad. Una de las características que hace que el SAVANT sea único es su arquitectura distribuida.

Una premisa fundamental de sistemas distribuidos es poder estar disponible a acomodarse a los cambios que puedan ocurrir en una red.

Un recurso computacional en una red, tal como un servidor u otro dispositivo, puede fallar o morir. Un nuevo recurso puede ser introducido a la red. Como resultado de esas acciones las aplicaciones ejecutándose en un recurso en especial puede trabajar desfavorablemente o fallar. Ese es el mismo caso en una red EPC. Por ejemplo, un lector del almacén puede fallar inesperadamente o un servidor puede salir de servicio. Como los sistemas distribuidos esta diseñados para responder a las variaciones de la red de igual manera lo hace una red EPC.

El Savant implementa lo que se conoce como una arquitectura de servicio federada, la cuál esencialmente proporciona capacidades de distribución y auto-organización. Estos bloques permiten una arquitectura dinámica y distribuida capaz de adaptarse a cambios imprevisto en la red. Esta arquitectura también hace al Savant altamente escalable.

Por ejemplo, servicios o componentes individuales como el servicio o el registro del lector, pueden ser distribuidos para funcionar como recursos que comparten múltiples redes.

Además, esta arquitectura es flexible. Si el lector o el otro recurso está físicamente inhabilitado o dañado en el piso de la tienda, el gestor de eventos continúa trabajando dinámicamente aprovisionando y relocalizando los servicios de un recurso a otro en la red.

Algunos de los componentes del gestor de eventos son:

- Adaptador de dispositivo: esta capa permite habilitar dispositivos de diferentes fabricantes, ya sea RFID o lectores de códigos de barras, comunicarse e interactuar con el gestor de eventos
- Filtros: Estos asisten en interpretar los datos útiles del ruido constante generando por la información de las etiquetas. Filtros pueden contener también pequeñas piezas de procesos o lógica de negocio. Filtros estándares son provistos para revisión, organización, cambio o bloqueo de los eventos.
- Notificadores: son similares a los dispositivos adaptadores, excepto que estos se usan para notificar a sistemas externos, los eventos de datos. El gestor de eventos de EPC provee notificadores estándar que registran información ya sea en un archivo, un mensaje, o a través de XML, http o mensajes SOAP.
- Gateway empresarial: Este componente es usado como la interfaz a aplicaciones empresariales que solicitan datos del administrador de eventos EPC.

El servicio de información EPC provee acceso a eventos significativos de negocio generados por el administrador de eventos EPC. Este también sirve como capa de integración que ofrece múltiples opciones para integración del Savant con las aplicaciones empresariales. Conexiones directas al Savant puede que resultaren en la creación de sitios de información.

Usando el servicio de información EPC entre el Savant y las aplicaciones de la empresa, se logra máximo de flexibilidad cuando los requisitos de negocio o aplicaciones empresariales cambian.

Datos del Savant alimentan el servidor de información EPC, donde es almacenado y queda disponible de manera consistente con cualquier aplicación que lo requiera. Esta aproximación incrementa la confiabilidad y flexibilidad, mientras reduce los costos de mantenimiento y soporte.

Esto también provee una adecuada localización y correlación de eventos en una lógica de negocios. Otro beneficio es que el servidor de información EPC puede ser usado para implementar una funcionalidad adicional, como cambio de formato de datos o almacenamiento de los mismos. [6]

2.2.3. Servicio de información PML y EPC

A continuación se describe los estándares de servicios de información PML (Lenguaje de descripción físico) y EPC. Estándares aun en evolución.

El desarrollo de PML es parte de los esfuerzos por estandarizar interfaces y protocolos para comunicación al interior de la infraestructura de EPC propuesta la actual versión 1.0 define la sintaxis y semántica básica.

PML provee un estándar basado en XML para el intercambio de datos EPC y consta de dos partes:

- Núcleo PML: describe los datos en bruto recibidos de las TAGs y sensores.
- Extensiones PML: pueden ser usadas para asociar información con objetos físicos.

PML provee un método común, ampliamente jerárquico para describir objetos físicos. Por ejemplo un productor de bebidas puede describir su producto, como una bebida carbonatada, la cual cae en la subcategoría de refresco o bebida sin alcohol. No todas las clasificaciones son tan simples, por lo que para asegurar que PML tenga una amplia aceptación se espera ampliar su definición y completar con otros estándares asociados. PML no pretende reemplazar los vocabularios actuales para transacciones de negocios u otras extensiones de XML, pero si complementa las definiciones relacionadas con datos en la red EPC.

Adicional a la información intrínseca de producto que no cambia, PML incluye datos que cambian constantemente (datos dinámicos) y datos que cambian con respecto al tiempo (datos temporales). Como dato dinámico en un archivo PML se puede incluir la temperatura de embarque de una fruta. Datos temporales, como la localización de un objeto cambian a través del ciclo de vida del producto. Convirtiendo toda esta información disponible en PML, permitiendo a las compañías,

emplear información en forma novedosa. Una compañía puede fijar datos para que el precio de un producto cambie cuando su fecha de expiración se aproxima. Operadores logísticos podrían expedir información en sus contratos sobre elementos como la temperatura de almacenamiento. [7]

2.2.4. Servicio de Información EPC

El servicio de información EPC esencialmente habilita los datos en formato PML para solicitudes de terceros, con su respectiva autorización. Los datos disponibles a través del servicio de información EPC pueden incluir:

- Datos colectados por el Savant a través de los lectores y sensores
- Especificar información de los objetos etiquetados como fecha de fabricación, peso, fecha de expiración y otros.
- Información de catálogos de productos

En respuesta a diversas peticiones, el servicio de información EPC obtiene información de una variedad de fuentes de datos que ya existen en una empresa trasladándolos a formato PML.

Es así como un servicio de información puede ser tan simple como un servicio Web como fachada a un EIS. [6]

2.3. ARQUITECTURA AUTO ID EN LA CONSTRUCCIÓN.

La tecnología RFID y basado en ella, el concepto de AUTO ID tiene un campo de aplicación multisectorial entre el que se ha localizado el sector de la construcción. En grandes construcciones se ve la dificultad de realizar de forma ágil el control de obra, esto se debe a la velocidad a la cual se trabaja y a la cantidad de materiales que intervienen en su ejecución.

La tecnología RFID y la propuesta AUTO ID integradas al sistema de presupuesto y control de obra, sugieren la solución para realizar de forma ágil el seguimiento y control de materiales en una construcción. Al no realizarse un control de obra adecuado no es posible cumplir con el presupuesto establecido, las consecuencias son evidentes en el aspecto económico.

Con la tecnología Auto-ID, los objetos físicos (Artículos que interviene en la construcción) llevan integrado un sistema que les permitirá comunicarse con los sistemas de información de las empresas constructoras y en especial con el sistema de control de obra, agilizando la captura de información. La tecnología Auto-ID proporciona un sistema numérico automatizado que revolucionará la forma como se controla los activos al interior de una construcción.

El esquema siguiente establece la relación que existe entre el control de obra en grandes construcciones y la propuesta AUTO ID. Figura 5 [7]

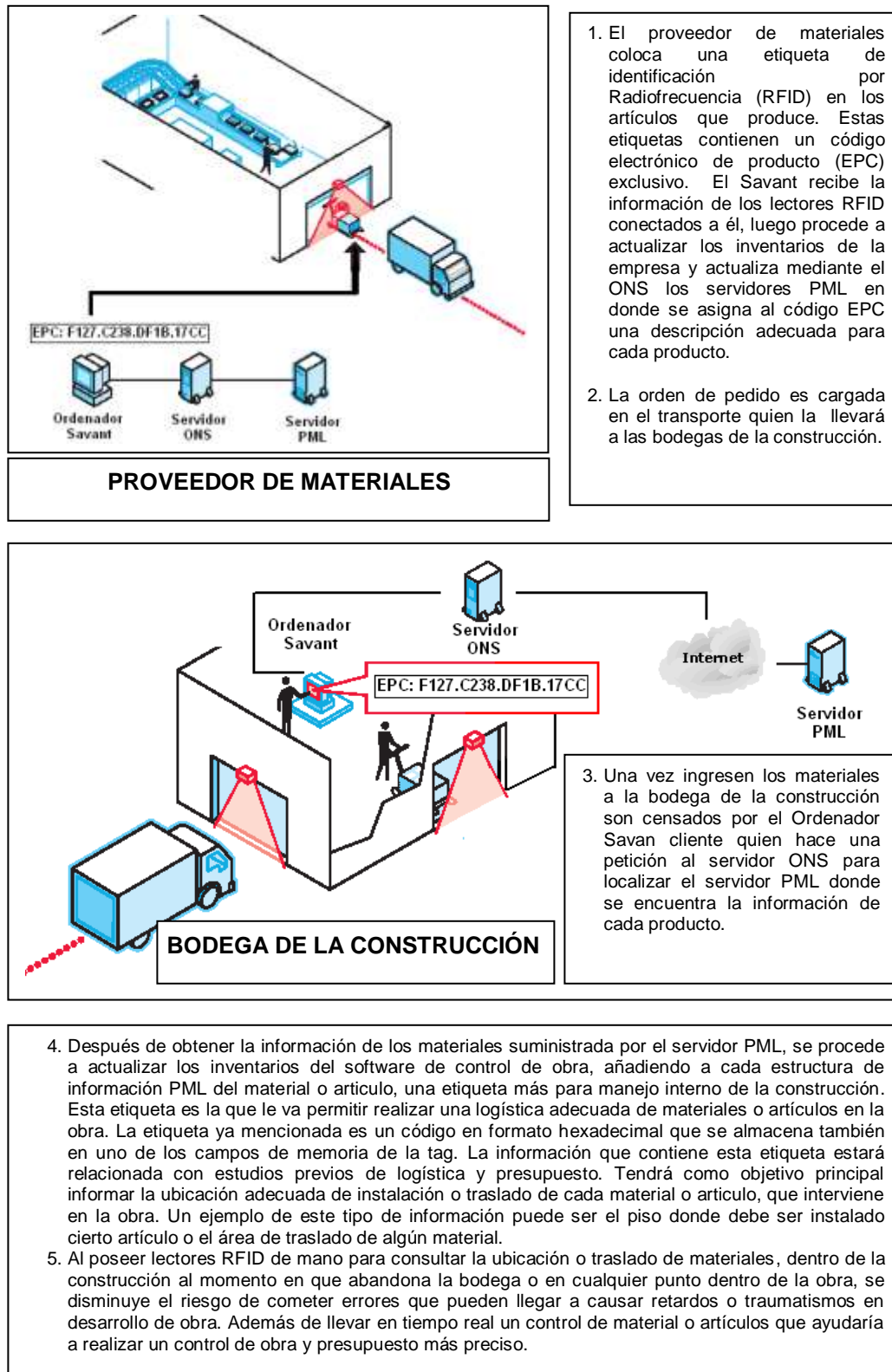


Figura 5 Arquitectura AUTO ID en el Contexto de la Construcción

2.4. EL ESPECTRO DE RADIO FRECUENCIA Y LA RED EPC

Los sistemas RFID deben operar en áreas de “aire libre” del espectro de telecomunicaciones inalámbricas a lo largo de las fronteras de regulación de los países y las regiones geográficas. En la actualidad, estas áreas no han sido armonizadas a nivel mundial, lo cual limita los beneficios potenciales que promete el EPC/RFID o cualquier red basada en tecnología RFID.

En general, las regulaciones dentro de un mismo país relacionadas con las comunicaciones de radio varían y son reguladas a través de diferentes organismos. Las regulaciones estadounidenses son más liberales. Las etiquetas pueden operar entre 902 y 915 Mhz, permitiendo un grado de frecuencia equitativo que avanza y extiende la distancia a lo largo de la cual las etiquetas pueden ser leídas. En Europa, hay más limitaciones. Por ejemplo, las redes de teléfono móviles utilizan esta parte del espectro y debajo de este espectro, la frecuencia 865-868MHz está reservada para los servicios médicos.

Asimismo, el rendimiento de lectura está respaldado por los niveles de energía, por lo tanto el tema está ligado al debate del espectro de radio frecuencia, la adjudicación de un rango de frecuencias tiene asociado una adjudicación de potencia de emisión. Los niveles de energía se limitan a 0.5 vatios en la mayoría de los países europeos. Sin embargo, el método para medir la energía promedio en los Estados Unidos no es el mismo que en Europa. La expectativa es que Europa tendrá una regulación de energía de 2 vatios en un futuro cercano, que será equivalente aproximadamente a 4 vatios en los Estados Unidos. A nivel europeo, los

siguientes organismos están involucrados en el desarrollo de los estándares para llegar a acuerdos relacionados con las regulaciones (figura 6):

- El European Telecommunications Standards Institute (ETSI) actualmente está trabajando en los estándares propuestos para el uso del espectro de radio frecuencia a lo largo de toda Europa.
- La European Radiocommunications Office (ERO) recomienda la aprobación de estos estándares.
- La European Commission publica directivas para legitimizar estos estándares.

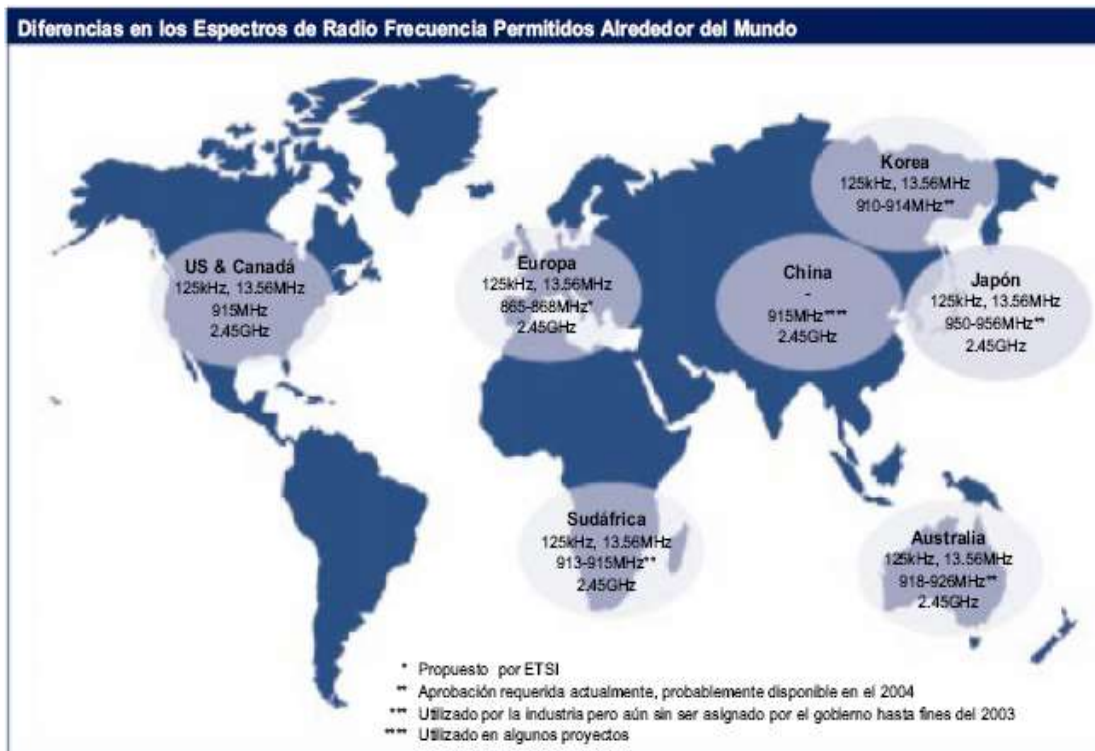


Figura 6. Espectro de frecuencias alrededor del Mundo [8]

A nivel mundial, la International Telecommunications Union (ITU) es responsable de regular el uso de radio frecuencia alrededor del mundo. El diagrama anterior

muestra algunas de las frecuencias propuestas y permitidas en la actualidad alrededor del mundo.

La resolución del tema de la interface aérea es esencial para lograr una adopción estandarizada a nivel masivo de la red EPC y los organismos que establecen los estándares y las regulaciones están trabajando activamente para llegar a un nivel de compatibilidad en todo el mundo, por ejemplo, las etiquetas UHF pueden ser leídas en todo el rango de 860-960 MHz sin perder demasiada eficiencia, y además a distancias similares cuando utilizan aproximadamente iguales rangos de energía. [8]

La figura 7 ilustra sobre los beneficios, desventajas y aplicaciones de la banda de radio frecuencia.

Beneficios, Desventajas & Aplicaciones de la Banda de Radio Frecuencia			
Frecuencia	Beneficios	Desventajas	Aplicaciones Comunes
Baja (9-135 KHz)	<ul style="list-style-type: none"> • Aceptado en Todo el Mundo • Funciona Cerca de Metal • En Pleno Uso en la Actualidad 	<ul style="list-style-type: none"> • Alcance de Lectura <1.5m • Impráctico para las Operaciones de Depósito • No está en los Estándares EPC 	<ul style="list-style-type: none"> • ID Animal • Barriles de Cerveza • Cerraduras y Llave Automática • Libros de Biblioteca
Alta (13.56 MHz)	<ul style="list-style-type: none"> • Aceptado en Todo el Mundo • Funciona en la Mayoría de los Entornos • En Pleno Uso Hoy 	<ul style="list-style-type: none"> • Alcance de Lectura <1.5m • No Funciona Cerca de Metal 	<ul style="list-style-type: none"> • Seguimiento a Nivel Artículo • Equipaje de Aerolíneas • Acceso a Edificios
Ultra Alta (300-1200 MHz)	<ul style="list-style-type: none"> • Potencial de Alcance de Lectura más Largo, >1.5m Alcance de Lectura • Uso Comercial en Aumento 	<ul style="list-style-type: none"> • No está Listo para ser Utilizado en Japón • No Funciona en Entornos Húmedos • Se Desintoniza cuando las Etiquetas están Próximas Físicamente 	<ul style="list-style-type: none"> • Seguimiento de Cajas, Pallets & Contenedores • Seguimiento de Camiones y Trailers
Micro-onda (2.45 o 5.8 GHz)	<ul style="list-style-type: none"> • Mayor Potencial de Alcance de Lectura, Alcance de Lectura >1.5m 	<ul style="list-style-type: none"> • Sin licencia de Uso Comercial en Partes de UE • Desarrollo de Sistemas Complejos 	<ul style="list-style-type: none"> • Control de Acceso (Vehículos)

Figura 7. Ventajas y Desventajas de la Banda de Frecuencias

3. DESCRIPCIÓN DE LA SOLUCIÓN DESARROLLADA

Este capítulo describe la solución al problema planteado para este trabajo de grado, el cual busca proponer una implementación de la tecnología RFID bajo la propuesta AUTO ID en un área de potencial aplicación como es la construcción de grandes obras civiles, aportando conocimiento en temas relacionados con los componentes y estructura funcional de esta nueva tecnología, además de visualizar una nueva red basada en artículos que se impondrá en unos pocos años en todo el mundo.

Esta solución esta basada en el Modelo de Construcción de Soluciones (MCS) [9]. Este modelo establece cuatro fases de referencia, de las cuales las dos primeras **Estudio de Prefactibilidad** y **Formulación del Proyecto** fueron abordadas en el desarrollo del anteproyecto. Se describe entonces en este capítulo las dos ultimas fases propuestas por este modelo.

Después de realizar la investigación acerca de la tecnología RFID y la propuesta de la nueva red de artículos planteada por EPC Global, se procedió a realizar un prototipo del proyecto, donde se le dio al sistema las capacidades operacionales necesarias para aterrizar el alcance real del proyecto. Todo esto basándonos en la arquitectura propuesta por EPC e identificando el propósito de la aplicación a

desarrollar. Identificamos en esta fase del proyecto dónde estaría ubicado dentro de la arquitectura, la aplicación software desarrollada para solucionar el problema planteado. Se consigna también en esta capítulo las pruebas desarrolladas y la forma como se solucionaron las limitaciones que se tienen al no estar establecida de forma real la red EPC.

3.1. DESCRIPCIÓN DE LA APLICACIÓN

Basándonos en la arquitectura planteada por la red EPC ya descrita en el capítulo 2 de este escrito, entramos a explicar cuál es la funcionalidad y ubicación dentro de esta arquitectura, de la aplicación bautizada por nosotros como Mediador EPC para la construcción (MEPCC).

En la figura 8 se explica de forma gráfica la ubicación dentro de la arquitectura EPC del MEPCC, para posteriormente entrar en detalle, sobre su funcionalidad y forma de interactuar dentro de la red propuesta por EPC. Dado que la red EPC está en prueba, no se dispone realmente de sus componentes en la actualidad, por lo cual se hizo necesario obviar el servidor ONS y desarrollar un simulador para el servidor PML que hace las veces de sistema de información de EPC. Además se integró al MEPCC a una aplicación sencilla que hace las veces de sistema de control de obra o repositorio de datos del cliente.



Figura 8. La Red EPC y el MEPCC.

En el caso del servidor ONS fue obviado dado que, su función principal es la de hacer la traducción de el código EPC de producto, en una dirección IP de servidor PML. La simulación del servidor PML es realmente un servidor Web de información para los diferentes productos que tiene incorporado un código EPC.

Observando la anterior grafica y comparándola con la arquitectura EPC original (Figura 4 Arquitectura de la red EPC), se puede establecer la ubicación de la aplicación MEPCC y determinar, que realmente actúa como un Savant dentro de la misma, adicionando algunas funcionalidades como: la solicitud de información de forma directa al servidor PML de artículos, y la actualización del sistema de base de datos de la aplicación empresarial de control de obra, para la cual debe realizar un interpretación del metalenguaje PML basado en XML. Se debe tener en cuenta que la arquitectura EPC descrita en el segundo capítulo de este escrito, describe la red en cada uno de sus componente y la interacción de los de los productores de

artículo con ellas, es decir el papel que cumplen dentro de la red como por ejemplo, el ser alimentadores de los sistemas de información de EPC, para que aplicaciones de usuario final como la propuesta en este trabajo de grado, puedan beneficiarse de esta nueva red propuesta por EPC Global, cabe anotar que en el desarrollo de esta aplicación se contó, con la API de manejo y control de lectores RFID suministradas por el Ingeniero Gustavo Ramírez las cuales facilitaron el desarrollo de la aplicación, se tiene que tener en cuenta entonces, que esta API es propietaria.

Para aclarar un poco más, la siguiente es la explicación de cada uno de los componentes de la red EPC simulada y el papel que cumple la aplicación MEPCC en esta arquitectura. Figura 8:

3.1.1. Lectores

Esta es la primera capa dentro de la arquitectura que simula la red ECP, en ella se contemplan los dispositivos RFID llamados lectores o interrogadores. El funcionamiento de estos dispositivos fue descrito en el segundo capítulo. Estos dispositivos son los responsables de leer y escribir si fuese necesario sobre las tag de radiofrecuencia quienes poseen el código EPC. También ellos pueden incorporar sensores de diferente tipos como: temperatura, humedad radioactividad, etc. Además poseen puertos I/O para ser incorporados a eventos externos dentro de cualquier cadena de suministro o automatización.

Para este trabajo de grado se contó con dos tipos de lectores dentro de la gran gama de fabricantes y tecnologías. El primero de ellos pertenece en el rango de frecuencia UHF 905MHz a 928MHz de marca ALIEN ALR-9780 y el otro en el rango HF 13.56MHz de la fábrica TEXAS SR-6400. Estos dos rangos de frecuencias están estandarizados bajo normas propuestas para la red EPC. Las Tags son ALIEN Generación 1 Clase1b a 915MHz ALR-9338-02 y TEXAS clase1 a 13.56MHz.

3.1.2. MEPCC

Esta es la aplicación central dentro de toda la arquitectura propuesta, como ya se mencionó en el apartado anterior, hace las veces de Savant, esto comparando con la arquitectura de red EPC. Cabe aclarar que tiene en gran parte, la funcionalidad del Savant definido para la red EPC dentro del estándar WD-savant-1_0-20030911 como es el manejo de lectores. Se le adiciono la capacidad de realizar peticiones de información a los servidores PML, la interpretación del metalenguaje PML de producto y la integración hacia el sistema de control de obra mediante la actualización de las bases de datos de este tipo de aplicación empresarial. Esto contribuye en parte a solucionar el problema de inventarios en la obra.

Este aplicativo al estar integrado a la aplicación empresarial puede realizar peticiones de información sobre algún código EPC en particular, con la finalidad de establecer la ubicación exacta de traslado del artículo, que abandona las bodegas de la construcción. Esto mediante el despliegue en pantalla de la información pertinente, por ejemplo el piso o área al que pertenece el artículo.

Toda la aplicación se desarrolló en el IDE Eclipse sobre lenguaje java y el JRE1.4.0. Se adicionó la API de manejo del lector ya mencionada y la API Javaxcomm para el puerto serial RS232.

La API de manejo suministradas por el ingeniero Gustavo Ramírez fue creada bajo el estándar ya mencionado para el Savant. El interprete de información en formato PML, que hace las veces de filtro hacia la base de datos del sistema de control de obra, es un interprete de lenguaje XML, utilizando las librerías jdom.

3.1.3. ONS

La funcionalidad de esta capa dentro de la arquitectura es la de hacer la traducción de código ECP en una dirección IP de servidor de información PML, para de esta forma realizar la conexión e iniciar el intercambio de datos. El símil dentro de una red Ethernet es el DNS, con la diferencia de que el ONS actúa como un directorio telefónico invertido, ya que el da la información en base a un número. Por razones ya expuestas en la introducción de este capítulo esta capa fue obviada.

3.1.4. Servicio de Información EPC

Es realmente un servidor PML o servidor de información de producto. Esta capa es en esencia un servidor WEB, el cual al recibir una petición de información de un código de producto EPC, responde construyendo un archivo en formato PML y

enviándolo al emisor de la solicitud. Este archivo contiene la información del producto que reposa sobre la base de datos del sistema de información EPC. El servidor WEB fue desarrollado totalmente en lenguaje java: IDE eclipse-SDK-2.1-win32, servidor de aplicaciones jboss-3.2.1 y plataforma de servicios J2EE lomboz21_02.

3.1.5. Aplicación Empresarial

Esta es una aplicación que simula un modulo en particular que puede ser parte de una ERP, en este caso un control de obra. La simulación entonces integra el sistema MEPCC a un sistema de bases de datos que hace parte del módulo control de obra. Agrega además la capacidad de actualizar los campos correspondientes de ubicación de cada artículo dentro de la obra, funcionalidad esta que permite dar solución al problema planteado en este trabajo de grado, ya que al consultar el código EPC del artículo que abandona la bodega en el sistema de control de obra, se sabrá con exactitud el lugar al cual debe ser trasladado el artículo, permitiendo con esto contribuir a una correcta logística de construcción, disminuyendo los costos en que se incurre, al equivocar el traslado de artículos al interior de la obra.

3.2. EJECUCIÓN DEL PROYECTO

Se decidió construir una aplicación con dos componentes para simular la red EPC, el primer componente que se diseño fue el MEPCC integrado a sistema de base de

datos que hace las veces de sistema de control de obra, el segundo fue un servidor WEB o servidor PML de información.

Se tomo la decisión después de la investigación realizada sobre la red propuesta por EPC Global, desarrollar la aplicación en Java 2 Enterprise Edition (J2EE), al ser esta una arquitectura multicapa para implementar aplicaciones de tipo empresarial y aplicaciones basadas en la Web, se ajusta a lo propuesto por esta nueva red.

3.2.1. Componentes J2EE para la Aplicación

La aplicación J2EE la conforman dos componentes. Cada componente es una unidad de software funcional auto-contenida que se ensambla dentro de la aplicación J2EE con sus clases de ayuda y ficheros. Basándonos en la especificación J2EE definimos los componentes así:

Las **Aplicaciones Cliente**, se ejecutan en el lado del cliente. En nuestra aplicación tiene el nombre MEPCC, es el responsable de servir de mediador en la arquitectura y además es el punto de integración para todo el sistema propuesto para solucionar el problema planteado en este trabajo de grado.

La **Aplicaciones Servidor**, contiene los componentes: **Java Servlet** y **Enterprise JavaBeans**. La tecnología JavaServer Pages son componentes Web que se ejecutan en el lado del servidor y los (beans enterprise) son componentes de negocio que se ejecutan en el servidor de aplicación. En nuestra aplicación este componente desarrolla la funcionalidad del servidor PML propuesto para la red EPC.

La figura 9 ilustra la arquitectura del sistema e incluye los repositorios de datos tanto de cliente como de servidor, además de la interacción con el hardware RFID (lectores y tags).

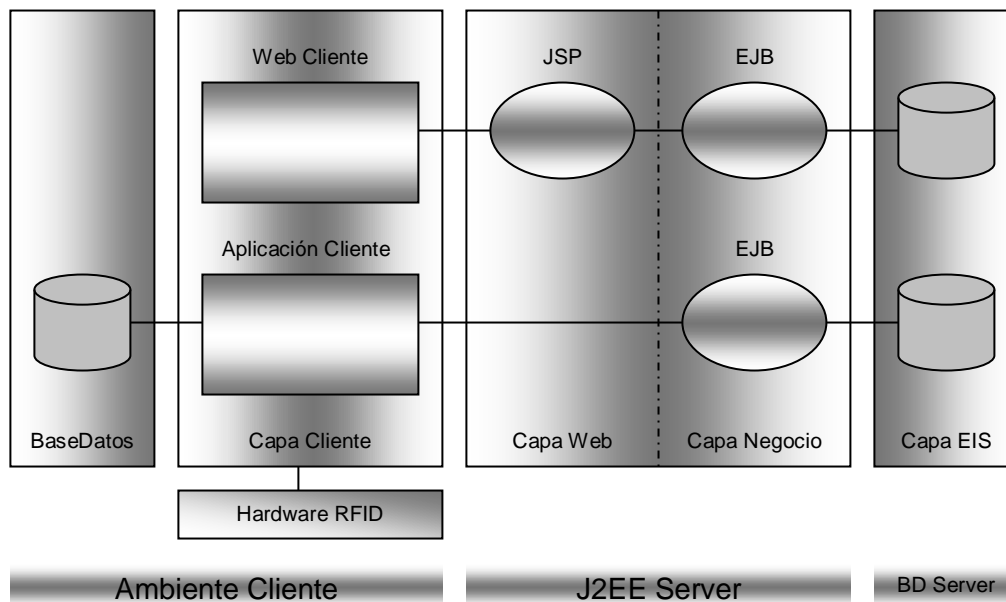


Figura 9 Aplicación J2EE Arquitectura

3.2.1.1. Cliente J2EE MEPCC

Se desarrollo dos tipos de clientes: El cliente Web y la Aplicación Cliente como se observa en la figura 9.

El cliente Web consta de dos partes, página Web dinámicas que contiene distintos tipos de lenguajes de marcas (HTML, XML, y otros), que son generados por los componentes Web que se ejecutan en la capa Web, y un navegador Web, que dibuja las páginas recibidas desde el servidor.

La aplicación cliente se ejecuta sobre una máquina cliente y proporciona la forma en que los usuarios pueden manejar la interface que contiene toda la funcionalidad del mediador del sistema, esto incluye: Configuración e interacción con el Hardware RFID, consultas al servidor PML, sincronización de la información suministrada por el servidor PML y la base de datos del cliente, además de establecer las pautas para la logística en la construcción. La interface gráfica de usuario (GUI) creada con los APIs **Swing** proporciona una manera fácil de interactuar con todas las funcionalidades ya mencionadas. La aplicación cliente accede directamente a los enterprise beans que se ejecutan en la capa de negocio. Pero si se necesita un cliente Web pueden abrir una conexión HTTP para establecer comunicación con un servlet que se ejecute en la capa Web.

3.2.1.2. Servidor J2EE Componente Web

El componente Web en la arquitectura planteada J2EE es una página **JSP**. Esta pagina JSP es un documento basados en texto que se ejecutan como **servlets** pero permiten una aproximación más natural para crear contenido estático. Esta capa incluye componentes **JavaBeans** para manejar la entrada del usuario y enviar esta entrada a los **enterprise beans** que se ejecutan en la capa de negocio para su procesamiento.

3.2.1.3. Servidor J2EE Componente de Negocio

El código de negocio, que es la lógica que resuelve o cumple las necesidades de un negocio particular, se maneja mediante enterprise beans que se ejecutan en la capa de negocio. Donde un enterprise bean recibe datos de los programas clientes, los procesa (si es necesario), y los envía a la capa del sistema de información empresarial para su almacenamiento. Un enterprise bean también recupera datos desde el almacenamiento, los procesa (si es necesario), y los envía de vuelta al programa cliente.

Hay tres tipos de enterprise beans: beans de sesión (con o sin estado), beans de entidad (manejados por el bean o por el contenedor) y beans dirigidos a mensaje. Un bean de sesión representa una conversación temporal con un cliente. Cuando el cliente finaliza su ejecución, el bean de sesión y sus datos desaparecen. Por el contrario, un bean de entidad representa datos persistentes almacenados en una fila de una tabla/relación de una base de datos. Si el cliente se termina o si se apaga el servidor, los servicios subyacentes se aseguran de grabar el bean. Un bean dirigido-a-mensaje combina las características de un bean de sesión y de un oyente de Java Message Service (JMS), permitiendo que un componente de negocio reciba asíncronamente mensajes JMS.

Para la aplicación se desarrollo un beans de sesión sin estado, con el cual se cumplió el objetivo de simulara el servidor PML.

Se trabajó de dos forma: En la primera, la aplicación cliente accede directamente al enterprise bean y solicita verificar un ID de producto, el enterprise bean atiende la

solicitud la procesa y extrae de la base de datos los ítem adecuados para forma el archivo PML, que retorna luego al cliente, en este archivo están las características del artículo. En la segunda, se creó un servicio Web para ser accedido desde la aplicación cliente, por intermedia de una página JSP que se comunica directamente con el enterprice bean y que realiza el mismo trabajo ya descrito para la consulta del ID de producto.

3.2.1.4. Capa EIS (Sistema de Información Empresarial)

La capa del sistema de información empresarial maneja el software del sistema de información empresarial e incluye la infraestructura del sistema así como los sistemas de planeación de recursos (ERP), procesamiento de transacciones a mainframes, sistemas de bases de datos y otros sistemas de información legales (personalizados). Los componentes de aplicaciones J2EE podrían necesitar acceder a estos sistemas de información empresariales para conectividad con sus bases de datos.

En la aplicación la parte que representa la capa IES es una base de datos que contiene las características de producto enlazadas a su código EPC.

3.2.2. Aplicación J2EE Proyecto ServidorPML

Creamos el proyecto J2EE **ServidorPML** con las siguientes características:

Ver ANEXO Configuraciones.

- El **Java Runtime Environment (JRE)** utilizado fue **j2re1.4.1_01**.
- **Integrated Development Environment (IDE) Eclipse 2.1**.
- Plug-in **Lomboz 2.1_02** para el desarrollo de aplicaciones **J2EE** y su integración con **JBOSS**.
- **JBOSS-3.2.1** como servidor de aplicaciones, junto con su servidor Web **Jetty** y su base de datos integrada **Hypersonic SQL**.
- Fichero de configuración de servidor **jBoss3.2.1all.server**

J2EE Modules:

- EJB Modules: **ServidorPMLMgr**
- WEB Modules: **OnlineServidorPML**

Paquetes:

- **TexasRFID.jar**
- **AlienRFID.jar**
- **comm.jar**
- **jdom.jar**

Para acceder a los datos de la base de datos y para hacer operaciones de negocio, crearemos varios componentes **J2EE** incluyendo **JavaBeans Enterprise** de Sesión y además clientes Web que utilizan **Servlets** y páginas **JSP**.

La figura 10 presenta una idea de la arquitectura de la aplicación **ServidorPML**.

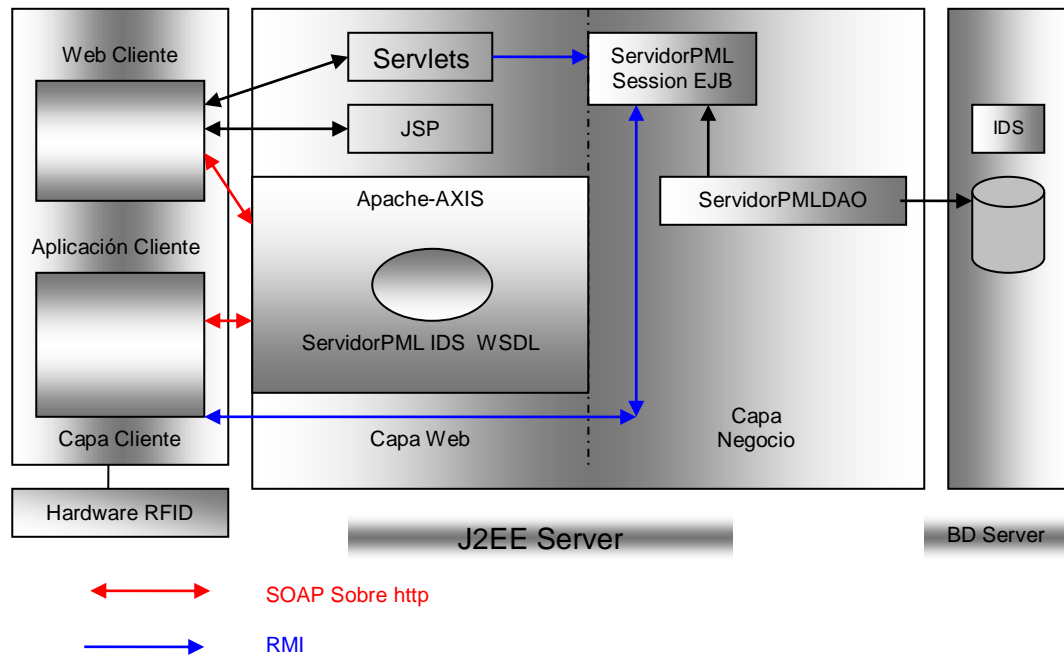


Figura 10. Aplicación J2EE ServidorPML

A continuación describimos su estructura completa en detalle:

3.2.2.1. Base de Datos EPC Sistema de Información Empresarial

Construimos una pequeña base de datos que contiene la tabla de **IDS** que reemplazan la información asociada al código EPC con estructura de 96 Bits según el estándar. Abajo tenemos una imagen de la base de datos. Ver ANEXO Configuraciones.

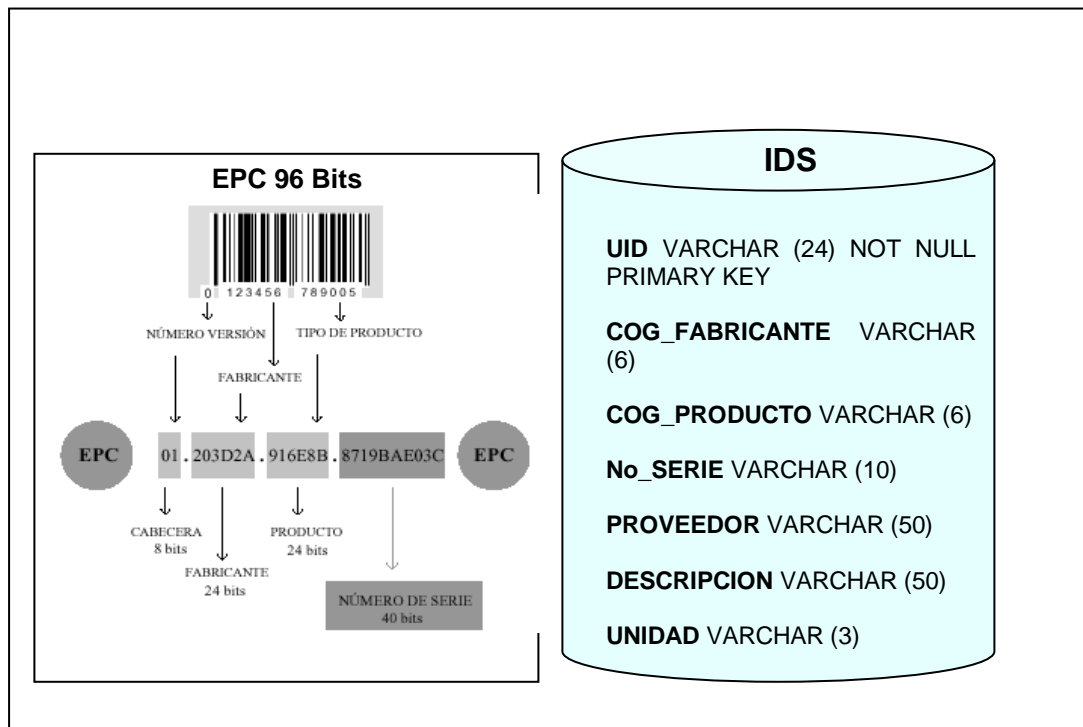


Figura 11. Base de Datos EPC Sistema de Información Empresarial

3.2.2.2. Bean de Sesión sin Estado ServidorPMLAccessBean

Creado bajo la siguiente jerarquía de paquetes **src/au.comm.tus.session/ ServidorPMLAccessBean** dentro del proyecto. Es el responsable de recibir y gestionar la petición de consulta de información desde el cliente, ya sea desde la aplicación cliente o por medio del componente JSP cuando es invocado el servicio WEB. Realiza su labor comunicándose con la base de datos utilizando un **DAO (Data Access Objet)** que encapsula el código JDBC. Un **DAO** tiene todos los atributos (campos) y comportamientos (métodos) correspondientes al bean que lo está utilizando. El método responsable de realizar esta tarea en el bean **ServidorPMLAccessBean** se llama **IDS**, recibe como parámetro un cadena de 24

caracteres código **EPC** y retorna un tipo de dato **Document** que contiene el archivo **PML**. El archivo **PML** es construido gracias a que todos los artículos en la base de datos tienen relacionado un **UID** que representa un código EPC de 96 Bits. Este código según el estándar representa: el fabricante, el producto y un número de serie, además se añadió tres campos en la tabla **UIDS**: proveedor, descripción y unidad representativa. Esta última información es de interés para el sistema de control de obra.

Este bean no tiene persistencia en la información. Ninguna información es almacenada durante la sección. Este tipo de bean es utilizado como **endpoint** para el **servicio Web**. Los **Beans de Sesión sin Estado**, son los únicos que se pueden utilizar para este propósito según se especifica en JSR-109.

El método **IDs** es un método de negocio, lleva a cabo operaciones o procesamientos de valores sobre el componente **EJB**. Desde la perspectiva del cliente, sólo pueden ver los métodos de negocio e invocarlos.

Cuando es creado el **ServidorPMLAccessBean** se crea a nivel de clases la etiqueta **@ejb.bean** que tiene asignados el tipo del bean, su nombre y el nombre JNDI que se generará en el interface **Home**. Esta etiqueta también generará los descriptores de despliegue en los ficheros **ejb-jar.xml** y **jboss.xml**. Cuando se utiliza la herramienta de generación de EJB del Plug-in **Lomboz 2.1_02**. El fragmento de código a continuación ilustra lo expuesto y la figura 12 muestra los archivos de configuración generados por esta herramienta.

```

package au.com.tusc.session;
import javax.ejb.SessionContext;
import org.jdom.Document;

/**
 *
 * @ejb.bean name="ServidorPMLAccess"
 * jndi-name="ServidorPMLAccess"
 * type="Stateless"
 * transaction-type="Container"
 *

```

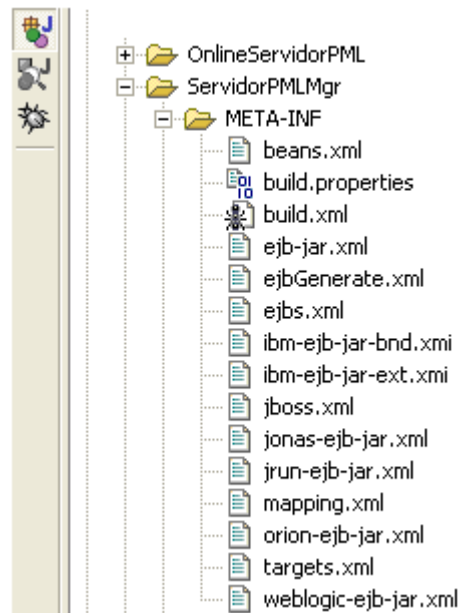
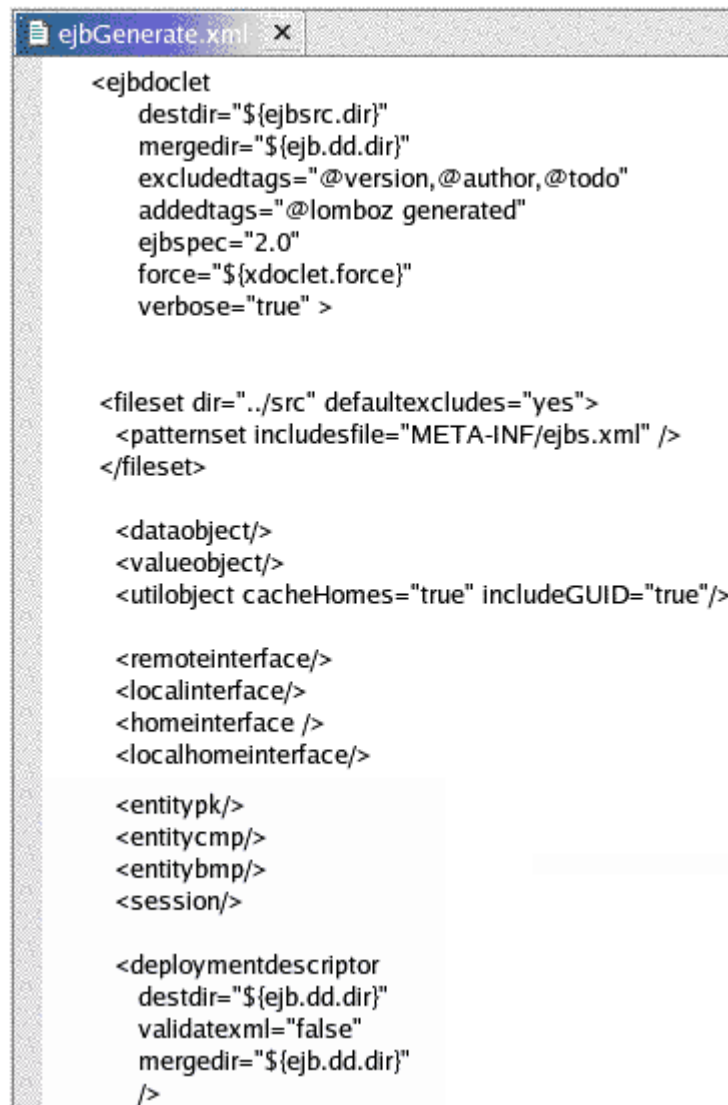


Figura 12. Archivos de Configuración Generados por Lombok

Lombok utiliza el fichero `ejbGenerate.xml` para generar los interfaces y clases de ayuda necesarias. El fichero `ejbGenerate.xml` se genera sólo una vez cuando creamos el módulo EJB. El archivo **`ejbGenerate.xml`**. a continuación se muestra parte de este archivo.



```

<ejbdoclet
  destdir="{ejbsrc.dir}"
  mergedir="{ejb.dd.dir}"
  excludedtags="@version,@author,@todo"
  addedtags="@lomboz generated"
  ejbspec="2.0"
  force="{xdoclet.force}"
  verbose="true" >

<fileset dir="../src" defaultexcludes="yes">
  <patternset includesfile="META-INF/ejbs.xml" />
</fileset>

<dataobject/>
<valueobject/>
<utilobject cacheHomes="true" includeGUID="true"/>

<remoteinterface/>
<localinterface/>
<homeinterface />
<localhomeinterface/>

<entitypk/>
<entitycmp/>
<entitybmp/>
<session/>

<deploymentdescriptor
  destdir="{ejb.dd.dir}"
  validatexml="false"
  mergedir="{ejb.dd.dir}"
  />

```

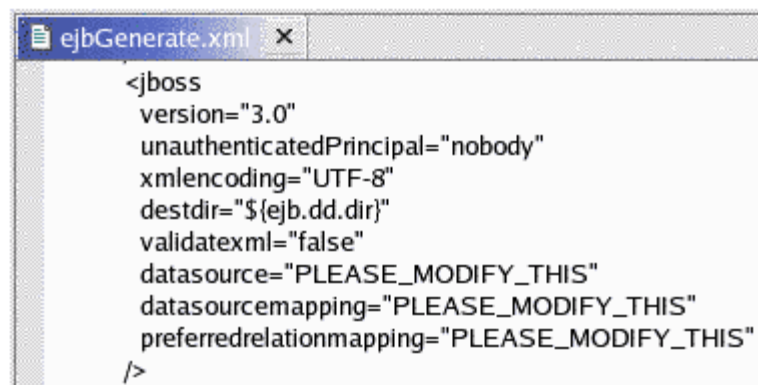
Figura 13 Archivo.ejbGenerate.xml

Como se puede ver a partir del fragmento de código mostrado en la figura13, se han definido las siguientes etiquetas:

- **<dataobject/>** genera objetos de datos para contener valores de campos persistentes del **EJB**, que corresponden con columnas en la tabla de la base de datos asociada. Se ha quedado obsoleto en favor de `value object` que es más poderoso en términos de relaciones (1-1, 1-n y n-m).

- **<utilobject/>** Crea un método para generación de GUID y para acceder a los objetos Home local y remoto.
- **<remoteinterface/>** Genera los interfaces remotos para EJBs.
- **<localinterface/>** Genera los interfaces locales para EJBs.
- **<homeinterface/>** Genera los interfaces home remotos para EJBs.
- **<localhomeinterface/>** Genera los interafaces home locales para EJBs.
- **<entitypk/>** Genera clases de claves primarias para EJBs de entidad.
- **<entitybmp/>** Crea clases de bean de entidad para EJBs de entidad BMP.
- **<entitycmp/>** Crea clases de bean de entidad para EJBs de entidad CMP.
- **<session/>** Genera clases de bean de sesión.

Como se puede ver en la figura 14, en este otro fragmento del código, también se ha definido la siguiente etiqueta:



```

<jboss
  version="3.0"
  unauthenticatedPrincipal="nobody"
  xmlencoding="UTF-8"
  destdir="{ejb.dd.dir}"
  validatexml="false"
  datasource="PLEASE_MODIFY_THIS"
  datasourcemapping="PLEASE_MODIFY_THIS"
  preferredrelationmapping="PLEASE_MODIFY_THIS"
/>


```

Figura 14 Archivo ejbGenerate.xml Etiquetas Jboss

- **<jboss/>** es una etiqueta específica de JBOSS que éste necesita. Se especificaron datasource, datasourcemapping y preferredrelationmapping, para la base de datos interna Hypersonic SQL, se comentaran sus valores más adelante. Ver también el ANEXO Configuraciones

Los otros dos ficheros importantes para nosotros son `ejb-jar.xml` y `jboss.xml`. El primero tiene todos los descriptores de despliegue para los beans y el segundo tiene los descriptores de despliegue específicos de JBOSS requeridos por el propio JBOSS.

Para configurar el DAO (Figura 15) personalizamos el fichero `ejbGenerate.xml`. Se añadió la etiqueta `<dao>` que especifica el directorio de destino para el interface DAO generado y qué patrón utilizar.



```

ejbGenerate.xml x
<ejbdoclet
  destdir="{ejbsrc.dir}"
  mergedir="{ejb.dd.dir}"
  excludedtags="@version,@author,@todo"
  addedtags="@lomboz generated"
  ejbspec="2.0"
  force="{xdoclet.force}"
  verbose="true" >

  <fileset dir="./src" defaultexcludes="yes">
    <patternset includesfile="META-INF/ejbs.xml" />
  </fileset>

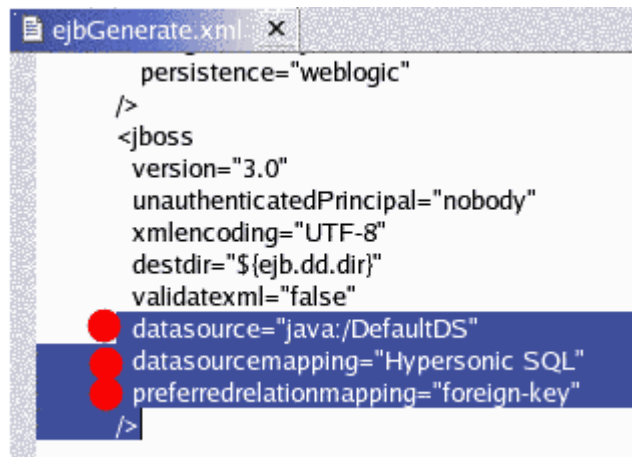
  <dataobject/>
  <valueobject/>
  <utilobject cacheHomes="true" includeGUID="true"/>
  <dao pattern="{0}" destDir="{ejbsrc.dir}" />

  <remoteinterface/>
  <localinterface/>
  <homeinterface />
  <localhomeinterface/>

```

Figura 15 Archivo `ejbGenerate.xml` Etiqueta `dao`

Hemos incluido el **datasource**, el **datasoucremapping** y el **preferredrelationmapping** mostrados en el siguiente fragmento de código: ver Figura 16.



```

persistence="weblogic"
/>
<jboss
version="3.0"
unauthenticatedPrincipal="nobody"
xmlencoding="UTF-8"
destdir="{ejb.dd.dir}"
validatexml="false"
datasource="java:/DefaultDS"
datasourcemapping="Hypersonic SQL"
preferredrelationmapping="foreign-key"
/>

```

Figura 16 Archivo ejbGenerate.xml Etiquetas data

- **datasource="java:/DefaultDS"** es un nombre JNDI local para la fuente de datos a utilizar.
- **datasourcemapping="Hypersonic SQL"** mapea los objetos/valores a las columnas y los tipos de datos asociados con esas columnas.
- **preferredrelationmapping="foreign-key"** define el tipo de base de datos a utilizar.

Como utilizamos la base de datos Hypersonic, estos parámetros son los apropiados. Estos parámetros están relacionados con el fichero de configuración standardjbosscmp-jdbc.xml que controla los mapeos CMP-a-JDBC para JBOSS.

3.2.2.3. La Interface DAO

Como se utilizó un DAO para acceder a la base de datos para este Bean sin estado, tuvimos que crear una clase **DAOImpl** llamada **ServidorPMLAccessDAOImpl** bajo la jerarquía de directorios

`src/au.com.tusc.dao` en el proyecto que implemente el interface **DAO** generado.

Figura 17.

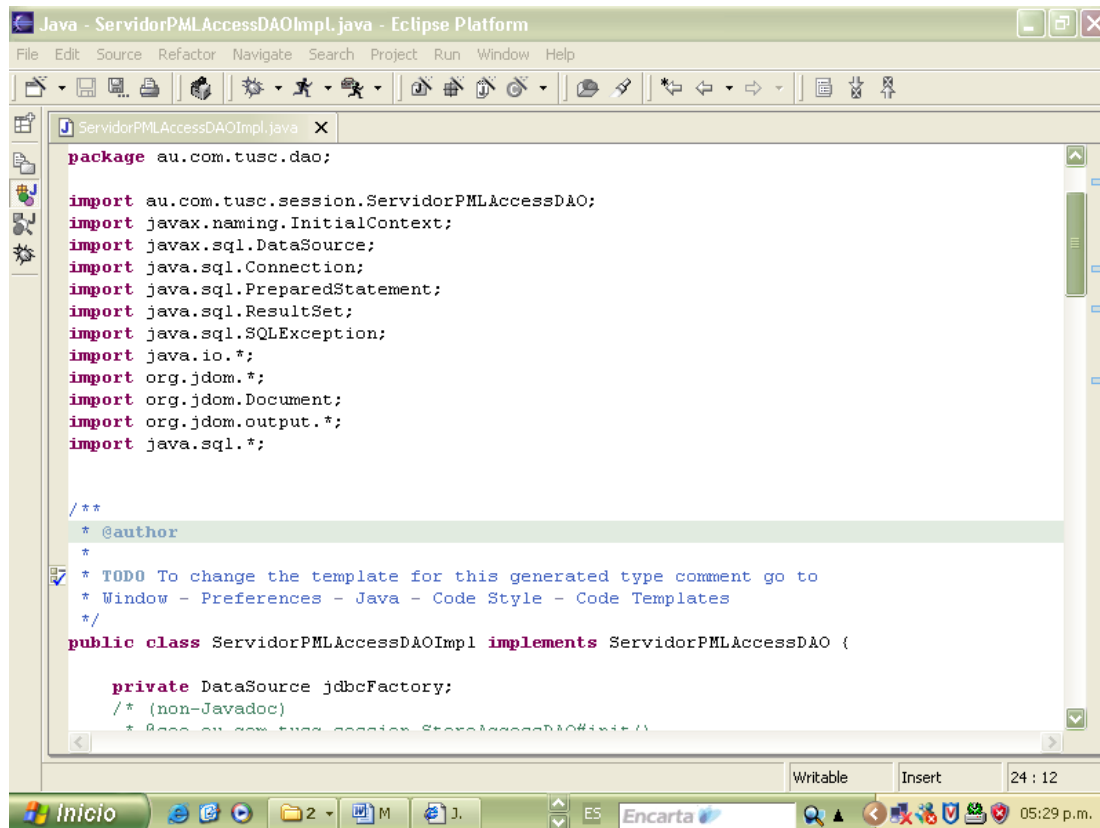


Figura 17 Interface DAO

Para genera el DAO agregamos la siguientes etiquetas a nivel de clases en el Bean **ServidorPMLAccessBean**.

```

*
* @ejb.dao class="au.com.tusc.session.ServidorPMLAccessDAO"
* impl-class="au.com.tusc.dao.ServidorPMLAccessDAOImpl"
*

```

Al generar los módulos EJB y sus clases con la herramienta **Lomboz** (Ver ANEXO Configuraciones) Las interfaces EJB y las clases de ayuda se generan en el directorio `ejbsrc/au.com.tusc.session` del proyecto como se muestra en la figura 18.

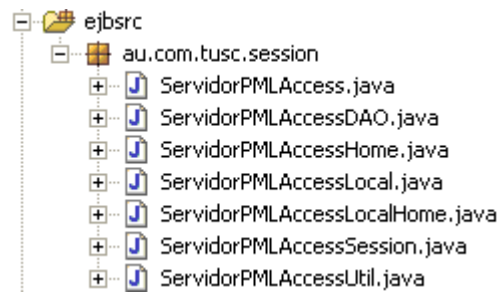


Figura 18 Clases ejbs Generadas en el Paquete session

Son 7 ficheros:

- **ServidorPMLAccess** es el interface remoto.
- **ServidorPMLAccessLocal** es el interface local.
- **ServidorPMLAccessSession** extiende de nuestra clase bean llamada **ServidorPMLAccesBean**.
- **ServidorPMLAccessHome** es el interface home remoto.
- **ServidorPMLAccessLocalHome** es el interface home local.
- **ServidorPMLAccessUtil** es una clase de ayuda que tiene métodos para acceder a los interfaces Home y LocalHome junto con la generación del GUID.
- **ServidorPMLAccesDAO** es el interface DAO que se utilizará para implementar nuestro **ServidorPMLAccessDAOImpl** en **au.com.tusc.dao**.

3.2.2.4. El Método de Negocio IDs

Este es el método de negocio responsable de generar la interface remota con la etiqueta a nivel de método **@ejb.interface** en el **ServidorPMLAccessBean**, invoca también al método del **DAO**, para comunicarse con la base de datos. Para eso se añadió la etiqueta **dao**, para que en la interface **DAO** se genere un método con esta

firma, para luego implementarlo en la clase **DAOImpl**. Entonces este método de negocio puede invocar el método de la clase **DAOImpl** para obtener el resultado deseado. La fracción de código es mostrada abajo.

```
/**
 * @ejb.interface-method
 *     view-type="remote"
 *
 * @dao.call name="IDs"
 **/
    public Document IDs(String UID){
        return null;
    }
```

Al generar las clases de ayuda con la herramienta **Lomboz**, en la interface **ServidorPMLAccesDAO** se crean dos métodos **init()** por defecto y **long.jdom.Document IDs(java.lang.String UID)** por la etiqueta **dao** ya mencionada. En la clase de ayuda **ServidorPMLAccessSession** se crean dos métodos de interés **getDao()** que crea un ejemplar de la clase **DAOImpl** y **org.jdom.Document IDs(java.lang.String UID)** que llama al método **Document IDs(String UID)** de la clase **ServidorPMLAccessDAOImpl**.

3.2.2.5. Implementación DAO

Se implementaron los métodos de la clase **ServidorPMLAccessDAOImpl**, importando los paquetes adecuados (Ver ANEXO Código Fuente), tanto los de conexión a base de datos como los paquetes de manejo de archivos XML de **Jdom**.

En el método **init()**, busca la referencia **jdbc/DefaultDS** utilizando el API JNDI, y almacena la referencia en la variable **jdbcFactory**. El string de búsqueda es **java:comp/env/jdbc/DefaultDS**.

En el método **Document IDs(String UID)** , primero obtenemos la conexión a la base de datos utilizando **jdbcFactory.**, creamos una sentencia SQL que busque el artículo y sus características en la tabla **IDS** de la base de datos, tomando el argumento **String UID** que representa el código **EPC**. Luego de extraer de la tabla los valores correspondientes construimos el **document** que representa el archivo **PML** del producto y lo retornamos como parámetro. (Ver ANEXO Código Fuente).

Hay que anotar aquí que al llamar al método **Document IDs(String UID)** de **ServidorPMLAccessDAOImpl**, ya está siendo invocado por el método **IDs(String UID)** de la clase **ServidorPMLAccessSession** que desciende de la clase **ServidorPMLAccessBean**, es decir, la clase **ServidorPMLAccessSession** ha sobrescrito este método.

3.2.2.6. Métodos de Retrollamada

Añadimos los siguientes métodos de retrollamada a la clase **ServidorPMLAccessBean**: **setSessionContext** y **UnsetSessionContext**. Es el contenedor EJB el que invoca a estos métodos de retro-llamada. (Ver ANEXO Código Fuente).

La clase **ServidorPMLAccessSession** desciende de la clase abstracta **ServidorPMLAccessBean** e implementa **SessionBean**, que sobrescribirá todos los métodos del interface **SessionBean**. Por eso, después de finalizar estos métodos

de la clase **bean**, generamos de nuevo las clases **EJB**. Ver ANEXO Configuraciones.

Ahora en los interfaces **Home** y **Remote** generados. En el caso del interface **Remote** todos los métodos de negocio declarados en el **bean** se han generado con la misma firma. Esto es a causa de las etiquetas a nivel de clases declaradas en el **bean ServidorPMLAccess** que hemos discutido antes. Por eso, **IDs(String)** se ha generado en un Interface Remote llamado **ServidorPMLAccess** a causa de la etiqueta `view-type="remote"` en el **bean**. En el caso del interface **Home** sólo se ha creado un método llamado **create**, que se genera por defecto a causa de la etiqueta `<homeinterface/>` que hay en **ejbGenerate.xml**: Además de esto, están **JNDI_NAME** y **COMP_NAME** (que es el nombre lógico para buscar el componente), estos se generan mediante la etiqueta declarada a nivel de clase en la clase **ServidorPMLAccessBean**. Ver ANEXO Código Fuente.

Para el despliegue del Bean ver Anexo Configuraciones.

3.2.2.7. Servicio Web

Se utilizo Apache-AXIS 1.1 para crear y desplegar el servicio Web. Para la configuración e integración a JBOSS ver ANEXO Configuraciones.

Jboss.3.2.x viene con soporte integrado para los servicios Web proporcionado por Jboss.Net. Jboss.NET tiene Apache AXIS como un plug-in de servicio para el Jboss-Microkernel. Jboss.Net utiliza el sistema de despliegue interno de JBOSS que

entiende el empaquetado de ficheros WSR [**Web Service Archive** (WSR- o .wsr-) que normalmente son ficheros **.jar** normales que contienen, los byte codes necesarios para conectarse a la maquinaria **Axis**, un descriptor de despliegue **Axis** ligeramente extendido en su entrada "META-INF/web-service.xml".

Lomboz-2.1_02 con Eclipse 2.1 y Lomboz 2.1.1 con Eclipse 2.1.1 no tienen un wizard para crear un servicio web (es decir un Web Service Deployment Descriptor [WSDD]) y desplegarlo. Por eso tendremos que hacerlo manualmente. Ambas versiones de Lomboz proporcionan un wizard para crear clientes SOAP. Ver Figura 10.

Para crear un servicio web necesitamos crear un **endpoint**. En este caso nuestro bean sin estado **ServidorPMLAccessBean** actuará como un **endpoint**. Se llamará al método **IDs** desde el lado del cliente utilizando este servicio Web, por eso tenemos que crear un fichero Web Service Deployment Descriptor (**WSDD**) describiendo la firma del método **IDs** junto con alguna otra información necesaria.

Creamos en el proyecto: el paquete con nombre **au.com.tusc.WebService**, el **archivo deploy.wsdd** y añadimos las siguientes etiquetas XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<deployment xmlns="http://xml.apache.org/axis/wsdd/"
xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
  <service name="ServidorPMLIDsService" provider="java:EJB">
    <parameter name="beanJndiName" value="ServidorPMLAccessLocal"/>
    <parameter name="homeInterfaceName"
value="au.com.tusc.session.ServidorPMLAccessLocalHome"/>
    <parameter name="remoteInterfaceName"
value="au.com.tusc.session.ServidorPMLAccessLocal"/>
  </service>
</deployment>
```

```

    <parameter name="allowedMethods" value="IDs"/>
    <parameter name="jndiURL" value="jnp://localhost:1099"/>
    <parameter name="jndiContextClass"
value="org.jnp.interfaces.NamingContextFactory"/>
  </service>
</deployment>

```

Examinemos estos descriptores de despliegue:

```
<service name="ServidorPMLIDsService" provider="java:EJB">
```

Esta etiqueta es la responsable de dar el nombre del servicio y el proveedor. El proveedor indica que el servicio web es un servicio EJB-Web. El nombre del servicio será **ServidorPMLIDsService** y el proveedor será java:EJB.

```
<parameter name="beanJndiName" value="ServidorPMLAccessLocal"/>
```

El primer parámetro es beanJndiName que especifica dónde está el bean (EJB) en el árbol JNDI. Estamos utilizando los interfaces locales de **ServidorPMLAccessBean**, por lo tanto es **ServidorPMLAccessLocal**.

Utilizamos interfaces locales para **ServidorPMLAccessBean**, cuando se accede al bean. Como todos los componentes están en la misma JVM, normalmente utilizamos interfaces locales ya que éstos tienen muchas ventajas en comparación con los interfaces remotos:

- a) Modelo de programación más sencillo, no necesitan capturar excepciones remotas que no pueden suceder en un entorno local.
- b) Como no se necesita tener RemoteException en todas las firmas de métodos, nuestro interface de métodos de negocio del EJB es independiente. Esto puede liberar a los llamadores de la dependencia del API EJB.

- c) Como se están ejecutando en la misma JVM y no están en un entorno distribuido, aunque están codificados como llamada-por-valor se ejecutan como llamada-por-referencia.

```
<parameter name="homeInterfaceName"
value="au.com.tusc.session.ServidorPMLAccessLocalHome"/>
```

Este parámetro tiene el valor del Interface Remote Home Local que es **au.com.tusc.session.ServidorPMLAccessLocalHome**

```
<parameter name="remoteInterfaceName"
value="au.com.tusc.session.ServidorPMLAccessLocal"/>
```

El valor de este parámetro es el interface Remote Local que es **au.com.tusc.session.ServidorPMLAccessLocal**

```
<parameter name="allowedMethods" value="IDs"/>
```

Este parámetro tiene el valor **IDs** para indicar que este método será el único que se hará público y al que se podrá acceder desde el cliente. El resto de parámetros en el archivo **deploy.wsdd** son específicos del servidor de aplicaciones, en este caso JBOSS.

Para el despliegue del servicio web ver Anexo Configuraciones.

Luego accedamos a nuestro servicio Web mediante su correspondiente Web Service Description Language (WSDL). El enlace wsdl mostrará el WSDL generado por AXIS para el servicio web desplegado. El archivo WSDL es grabado en el paquete **au.com.tusc.WebService** con nombre **SevidorPMLIDsService.wsdl**, ya que lo necesitaremos para crear el cliente SOAP utilizando el wizard de Lombos.

El wizard de Lombos para crear clientes SOAP utiliza la herramienta **WSDL2Java**, que construirá proxies Java y esqueletos para servicios con descriptores WSDL. Esta herramienta viene con la distribución de Apache-AXIS

3.2.2.7.1. El cliente Web

Lomboz (versiones 2.1_02 y 2.1.1) proporcionan un wizard para crear clientes SOAP para publicar servicios Web utilizando sus ficheros WSDL correspondientes. Se necesita todas las librerías necesarias en el classpath para generar el esqueleto para el cliente SOAP. Esta son las mismas ya utilizadas para \$AXISCLASSPATH. Ver Anexo Configuraciones.

Esto creará el esqueleto (*stub*) para el cliente utilizando los cuatro ficheros que hay en `au.com.tusc.WebService`: ver figura 19.

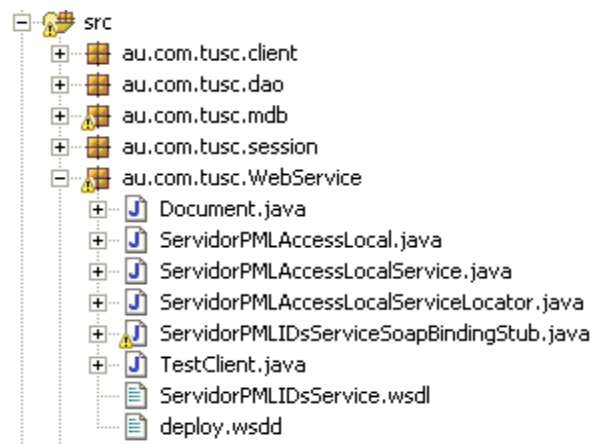


Figura 19 Archivo del Paquete WebService Generados

La clase cliente es implementada: figura 20.

```

package au.com.tusc.WebService;
import org.apache.axis.AxisFault;
/**
 * @author Administrador
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class TestClient {

    public static void main(String[] args) {
        try{
            ServidorPMLAccessLocalServiceLocator service =
                new ServidorPMLAccessLocalServiceLocator();
            ServidorPMLAccessLocal port = service.getServidorPMLIDsService();

            Document doc=port.IDs("ACEFB236CDE512BCAB1234BC");
            Object Root = doc.getRootElement();
            System.out.println(Root.toString());
        }
    }
}

```

Figura 20 La clase Cliente

Las sentencias principales son:

```

ServidorPMLAccessLocalServiceLocator service =

    new ServidorPMLAccessLocalServiceLocator();

```

Esta sentencia es necesaria para que el objeto ServiceLocator devuelva el endpoint del servicio web que utilizaremos para llamarlo. En este caso el punto final es <http://localhost:8080/axis/services/ServidorPMLIDsService>

```

ServidorPMLAccessLocal port = service.getServidorPMLIDsService();

```

Esta sentencia devuelve la referencia al interface local del bean para poder llamar a los métodos necesarios.

3.2.2.8. Aplicación Cliente

Esta aplicación llamada Mediador EPC para la Construcción **MEPCC**, es el punto central de la arquitectura. Con este desarrollo se puede interactuar con la red EPC supuesta, con el hardware de RFID y con el sistema de información empresarial y de control de obra representado por un sistema sencillo de base de datos. Fue desarrollado en lenguaje java y con la plataforma eclipse versión 3.0.1 sumado al plug -in VE 1.0.1. para desarrollos de interface grafica.

Esta aplicación contempla 2 paquetes principales y su interacción con el servidor PML, ya sea por el bean de sesión sin estado EJB de la capa de negocio o por intermedio del Web service.

El paquete **control** y sus clases: Build, CommandsHF y Serialport tienen la función de interactuar con el hardware de **RFID**, aquí hay que hacer una aclaración este paquete hace parte del desarrollo de propietario ya explicado con anterioridad.

El paquete **gui** y su clase **GuiMedidiadorEPC**, crean la interface de usuario e interactúa con el paquete control y los servicios de la aplicación de simulación PML.. A continuación se describe su funcionamiento.

3.2.2.8.1. TEXAS INSTRUMENTS TECHNOLOGY

Esta interface contempla la integración al sistema cliente, la consulta al servidor de PML y cuatro submenús, todo esto diseñado para poder explicar la interacción con

toda la arquitectura EPC. Los submenús son: Comunicaciones, Programar UID, Leer UID y SPC Base de Datos.

En **Comunicaciones** se establece la configuración del puerto RS232 al cual se va conectar el lector RFID. Al cargar el programa por primera vez el sistema toma una configuración predefinida en el archivo **ConfPuerto.xml** ubicado en la carpeta de la aplicación, cada cambio realizado en este submenú queda reflejada en este archivo y se muestra en la pantalla de comandos ubicada a la derecha.

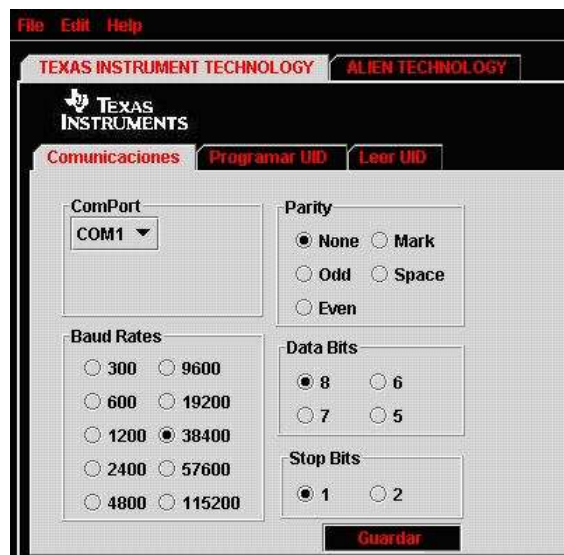


Figura 21. El menú Comunicaciones

Leer UID establece la acción de captura del código EPC de la tag para su posterior consulta al servidor PML, además se realiza la acción de logística grabando en la unidad de memoria la ubicación dentro de la construcción del artículo para una futura consulta. La consulta del código EPC se realiza con el botón **EPC?** y la integración al sistema cliente mediante el botón **SPC?**, estos botones están ubicados en la parte inferior de la pantalla de comandos.

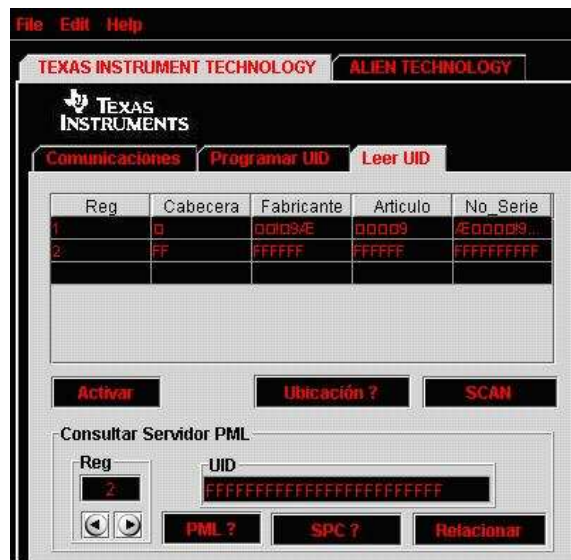


Figura 22. El menú Leer UID

Programar UID establece la posibilidad de interpretar toda la capacidad de memoria y configurar tag para la demostración.

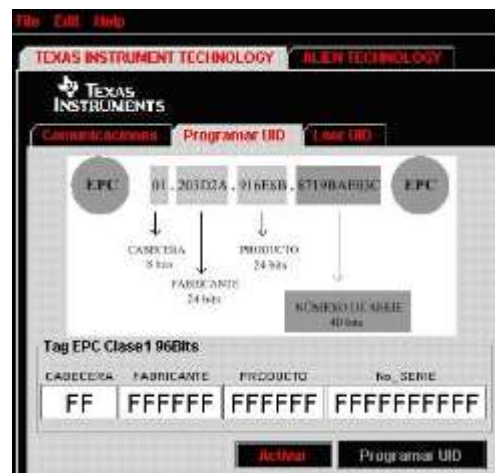


Figura 23. El menú Programar UID

El Botón botón **EPC?** (Figura 22) es la comprobación de la integración con el sistema cliente. Una vez realizada la consulta al servidor PML y establecer la logística al interior de la construcción se procede a realizar el almacenamiento de la información en el sistema de base de datos del cliente.

Con el botón **Ubicación** podemos realizar futuras consultas a la base de datos cliente que permiten dar solución al problema planteado para este trabajo de grado, con el cual conoceremos exactamente el lugar de destino del artículo o elemento que abandona la bodega o almacén . Para mirar más detalles, ver Anexo Manual de Usuario.

4. RECOMENDACIONES Y CONCLUSIONES

Ante el cuestionamiento de si es posible utilizar RFID para ayudar a la logística y control de una obra, con la estructura tecnológica que existe hoy. La respuesta será que es absolutamente posible hacer prototipos de trabajo que traten muchas necesidades en la cadena suministro en la industria de construcción. Pero no es factible o creíble considerar una puesta en práctica a gran escala de la tecnología hoy. El camino para la adopción de RFID estaría más probablemente con proyectos experimentales más pequeños que probarán los argumentos de que es posible o no realizar con esta tecnología. Hay muestras que la industria es convergente en la importancia de la aceptación para los estándares internacionales. La construcción podría conseguir aliados grandes en el mercado y de esta forma exigir el uso de RFID y la propuesta EPC, a los fabricantes de artículos para la construcción y a las firmas logísticas.

Las siguientes son las conclusiones generales y las sugerencias producto de de este trabajo de grado:

- Una masa crítica de fabricantes y minoristas han llevado a cabo pruebas utilizando la tecnología RFID y ahora se están preparando para una adopción más amplia. Es necesario iniciar estudios más amplios sobre la influencia en el mercado colombiano, este trabajo de grado simplemente orienta y ejemplariza la

necesidad de mirar los diferentes escenarios en los esta tecnología tendrá aplicación, en nuestro caso particular la construcción, que es uno de los renglones que más impacto tiene en al economía de nuestro país.

- Cada empresa incluyendo el sector de la construcción y sus proveedores de software, debería considerar la posibilidad de adoptar la visión EPC ahora mismo. Existe un nivel de entusiasmo cada vez mayor en el sector alrededor del mundo a medida que las pruebas realizadas por los primeros adoptantes están arrojando resultados muy exitosos. El importante valor de estas pruebas también ha estimulado a estos primeros adoptantes a continuar su camino hacia el EPC.
- Se espera que la actividad de la competencia y las solicitudes de los socios comerciales más importantes logren acelerar la implementación de la visión EPC en todo el mundo. Por lo tanto, es esencial entender cuanto antes qué efectos ejercerá sobre las empresas colombianas.
- Inevitablemente, el progreso logrado hasta la fecha ha sido impulsado por las grandes organizaciones multinacionales. Sin embargo, un aspecto importante de la visión EPC es que esta tecnología eventualmente estará disponible en cada comercio independiente local o pequeños minoristas, como ocurre con los lectores y códigos de barra en la actualidad.
- Se debería pensar en una implementación EPC como si se tratara de un proceso paso a paso, en lugar de ser un enfoque inmediato con efecto avalancha.

- En Colombia se encuentra el organismo SG1 Colombia, , que es el encargado de administrar y desarrollar el código EPC, así como de integrar todos los servicios necesario para ponerlo en funcionamiento. Este organismo pertenece al grupo SG1 anteriormente denominado EAN- UCC. Existen varias empresas como NOEL, Off Corss, las cuales están realizando pilotos en su cadena de producción y suministro.
- Se requieren mayores desarrollos en las áreas de estándares, tecnología, restricciones de las regulaciones y comunicaciones antes de que la visión EPC pueda realizarse completamente a gran escala. Sin embargo, cuando hablamos del cumplimiento de las promesas y de los beneficios de RFID ya no es más una cuestión de “si esto es posible...” sino “Cuando esto se implemente...”. No existe absolutamente ningún elemento que compone la visión EPC hoy que no pueda franquearse. Sin embargo, no existe razón para que las compañías esperen. Ellas pueden comenzar a realizar pruebas de los conceptos EPC hoy mismo.

REFERENCIAS

- [1] **SISTEMA EAN-UCC - RFID - IDENTIFICACION POR RADIO FRECUENCIA.** Noviembre de 1999.
- [2] **Auto-ID Center** (www.autoidcenter.org)
- [3] **The Networked Physical World Proposals for Engineering the Next Generation of Computing, Commerce & Automatic-Identification.** Sanjay Sarma, David L. Brock & Kevin Ashton. AUTO ID center. Enero 2001
- [4] **LA TECNOLOGÍA DE IDENTIFICACIÓN DE OBJETOS AUTO-ID: HACIA LA INTEGRACIÓN TOTAL DE SISTEMAS DISTRIBUIDOS.** Jaime Moraleda Novo, La Mancha, Andrés García Higuera, M^a Carmen Carnero Moya, Universidad de Castilla.
- [5] **TECHNOLOGY GUIDE.** auto-id center Massachusetts institute of technology.
- [6] **RFID/EPC La Solución Global para el Seguimiento y Control Total de los Procesos.**
- [7] **PML Core Specification 1.0. white paper.** Christian Floerkemeier, Dipan Anarkat, Ted Osinski, Mark Harrison. Enero, 2004.
- [8] **Mapa de Ruta del Global Commerce Initiative.** Global Commerce Initiative BM. Noviembre de 2003.
- [9] **Modelo de Construcción de Soluciones (MCS).** Carlos Enrique Serrano.
- [10] **Texas Instruments** (www.ti.com/tiris/default.htm).
- [11] **Rfid Journal** (<http://www.rfidjournal.com>).

BIBLIOGRAFIA

- ANDRESEN, et .al. A framework for measuring it innovation benefits.
(<http://www.itcon.org/2000/4/>)
- BYGGELOGISTIK I. Materialestyring i byggeprocessen ; sophiehaven : forste etape.1993. ISBN: 87-601-3714-2.
- FINKENZELLER, Klaus. RFID handbook second Edition. 2003. ISBN: 0-470-84402-7.
- HUGH, Beyer & HOLTZBLATT, Karen. Contextual design. 1998. ISBN: 1-55860-411-1.
- LONSETH, Marte. Logistikk i byggeprosessen; Bachelor thesis at norwegian university of science and technology. Trondheim.
- RAMIREZ, Gustavo. Marco de referencia para la integración de sistemas físicos y aplicaciones software desde la alternativa RFID; Tesis Maestría en Telemática. Universidad del Cauca. 2006.