



Universidad de Castilla-La Mancha

Escuela Superior de Ingeniería Informática

Departamento de Sistemas Informáticos

Programa Oficial de Postgrado en Tecnologías Informáticas Avanzadas

Trabajo Fin de Máster

**Implementación del estándar EPCIS en arquitecturas
basadas en Servicios Web para aplicaciones de
identificación y localización**

Septiembre de 2011

Alumno: José Iván San José Vieco

**Directores: Dr. D. Antonio Fernández Caballero
Dr. D. José Manuel Pastor García**

Índice General

1. Introducción	1
1.1. Introducción	1
1.2. Motivación	1
1.3. Estructura de la memoria	2
2. Asignaturas cursadas	3
2.1. Asignaturas optativas del primer cuatrimestre	3
2.1.1. Generación de Documentos Científicos en Informática	3
2.1.2. Introducción a la Programación de Arquitecturas de Altas Prestaciones	5
2.1.3. Nuevos Paradigmas en HCI	7
2.2. Asignaturas optativas del segundo cuatrimestre	9
2.2.1. Calidad de Interfaces de Usuario: Desarrollo Avanzado	9
2.2.2. Programación Internet con Lenguajes Declarativos Multiparadigma	11
2.2.3. Tecnología Software Orientada a Objetos	13
3. Estado del arte	15
3.1. Redes de Sensores Inalámbricas (WSN, Wireless Sensor Network)	16
3.1.1. Introducción	16
3.1.2. Componentes	16
3.1.3. Tipología básica	17
3.1.4. Campos de aplicación	18
3.2. ZigBee	19
3.2.1. Introducción	19
3.2.2. Características	19
3.2.3. Dispositivos	19
3.2.4. Arquitectura	20
3.2.5. Topologías de red	22
3.2.6. Campos de aplicación	23
3.3. Web Services	24
3.3.1. Introducción	24
3.3.2. Estándares	24
3.3.3. Ventajas e inconvenientes	25
3.3.4. Web 2.0	25
3.4. EPC Information Services (EPCIS)	28
3.4.1. Introducción	28
3.4.2. Especificación del Framework de EPCIS	28
3.4.3. Tipos de datos	32
3.4.4. Protocolo de lectura	32
3.5. Internet de las Cosas	33
3.5.1. Introducción	33

3.5.2.	Definición de Internet de las Cosas	33
3.5.3.	<i>Objetos Inteligentes</i>	34
3.5.4.	Dominio de aplicación	34
3.5.5.	Campos de aplicación de Internet de las Cosas	35
3.5.6.	Concepto Web de las Cosas	36
3.6.	Localización - Geolocalización	38
3.6.1.	Introducción	38
3.6.2.	Funcionamiento de un sistema SIG	38
3.6.3.	Software SIG	39
3.6.4.	Futuro de los SIG	40
4.	Trabajo de investigación	42
4.1.	Introducción	42
4.2.	Motivación	42
4.3.	Trabajo en desarrollo	44
4.3.1.	Fosstrak EPCIS Project	45
4.3.2.	Elementos hardware	49
5.	Conclusiones	51
6.	Futuras líneas y trabajo futuro	52
7.	Currículum Vitae	53
7.1.	Datos personales	53
7.2.	Formación Académica	53
7.3.	Participación en proyectos I+D	53
7.4.	Asociaciones Científico-Técnicas	54
7.5.	Líneas de Investigación	54
7.6.	Publicaciones	54
7.7.	Idiomas	54
7.8.	Otros méritos	54
Bibliografía		59

Índice de Figuras

3.1. Ejemplo de la arquitectura de una red de sensores	17
3.2. Topologías Redes de Sensores	17
3.3. Modelo de capas de Zigbee	21
3.4. Estándar EPCIS y otros estándares EPC	29
3.5. Capas EPCIS	30
3.6. Dominio de aplicación de Internet de las Cosas	35
4.1. Capas inferiores del estándar EPCIS	43
4.2. Capas intermedias del estándar EPCIS	44
4.3. Implementación Fosstrak EPCIS	46
4.4. Esquema del evento ObjectEvents	48
4.5. Esquemas de los vocabularios Business, Transaction, Diposition y ReadPoint . .	48
4.6. Esquema de las suscripciones	49

Capítulo 1

Introducción

1.1. Introducción

En este punto se describe el trabajo que se ha realizado durante el **Máster en Tecnologías Informáticas Avanzadas** de la Universidad de Castilla La Mancha durante el curso 2010-2011.

El Máster se compone de dos partes. La primera parte consta de **seis asignaturas optativas**, tres asignaturas por cuatrimestre, de 5 créditos *ECTS* cada una, lo que hacen un total de 30 créditos *ECTS*. Las asignaturas cursadas pueden consultarse en el capítulo 2 de esta memoria.

La segunda parte consiste la realización de un **Trabajo Fin de Máster** de 30 créditos *ECTS*, cuyo contenido servirá como punto de partida para la realización de la futura Tesis Doctoral. El trabajo que se está desarrollando será descrito en detalle en el capítulo 4 de esta memoria.

1.2. Motivación

El incremento en la miniaturización de componentes electrónicos y los avances en las modernas tecnologías de comunicación, lideran el desarrollo de nodos de sensores inteligentes, cada vez más pequeños y baratos. Estos nodos están formados por sensores, actuadores, procesadores de bajo consumo, poca memoria y un módulo de comunicación.

Estos avances han abierto nuevas líneas de investigación, líneas con las que el Grupo Autolog ha ido trabajando a través de una serie de proyectos I+D, en los que se ha utilizado el estándar *EPCIS* de *EPCglobal*, se ha trabajado con las capas inferiores de dicho estándar, siendo este trabajo siempre más cercano a la parte *hardware* que a la parte *software*.

En estos proyectos se ha trabajado con identificación y trazabilidad de productos (paneles de hormigón, industriales, farmacéuticos, de consumo alimenticio, etc.) a través del uso de la tecnología *RFID* y también se lleva tiempo trabajando con Redes de Sensores, sobre todo en temas de desarrollo *hardware* y comunicación utilizando protocolos de bajo nivel.

Una vez que se ha trabajado con las capas inferiores de este estándar, se nos plantea centrar nuestro trabajo en las capas superiores, enfatizando en algunas tecnologías como Internet de las Cosas y Servicios Web.

Por lo tanto, nuestra principal motivación será la de realizar una serie de aplicaciones, utilizando las capas superiores del estándar para así poder complementar los trabajos *hardware* que ya hemos realizado.

Como principal trabajo, proponemos la realización de una arquitectura de Servicios Web, utilizando como base el estándar *EPCIS* de *EPCglobal*, en la que podamos automatizar una serie de Servicios Web, que serán definidos posteriormente.

Tras definir e implementar esta arquitectura, propondremos la implementación de un primer Servicio Web que consiste en la realización de una aplicación basada en la localización de personas con algún tipo de enfermedad o minusvalía y con la necesidad de tenerlas siempre localizadas en entornos abiertos a través de una Red de Sensores Inalámbrica, formada por una serie de elementos que describiremos en capítulos posteriores de este trabajo.

Esta Red de Sensores Inalámbrica está formada nodos, los cuales pueden medir las condiciones del entorno, realizar cálculos, localizar personas o animales, añadir información y transmitir esos datos obtenidos a una estación base. Cientos de nodos, de una gran Red de Sensores Inalámbricos, pueden utilizarse para vigilar o tener controladas grandes extensiones de terreno.

La información que se transfiera entre los distintos nodos, los coordinadores y la estación base se realizará utilizando Servicios Web, enviando dicha información, normalmente en un fichero de texto o *XML*, con el formato que nos indique el estándar *EPCIS*. Para la realización de este trabajo, utilizaremos *Fosstrak EPCIS Project*, que es una implementación en código abierto de *EPCIS* y certificada por *EPCglobal*.

1.3. Estructura de la memoria

Esta memoria se ha estructurado en ocho capítulos, el primero de los cuales es el capítulo de Introducción.

En el capítulo 2 se hace una descripción de las distintas asignaturas cursadas en este Máster, explicando tanto los trabajos como las prácticas que se han realizado en cada una de ellas.

Dentro del capítulo 3, se ha descrito el estado del arte en torno al cual gira la línea de investigación que se presenta en este documento. Los conceptos principales que vamos a desarrollar son: Redes de Sensores Inalámbricas (*Wireless Sensor Network, WSN*) (punto 3.1), *ZigBee* (punto 3.2), Servicios Web (*Web Services*) (punto 3.3), *EPC Information Services (EPCIS)* (punto 3.4), Internet de las Cosas (*Internet of Things*) (punto 3.5) y Localización - Geolocalización (punto 3.6).

Posteriormente, en el capítulo 4 se ha descrito el trabajo de investigación que estamos llevando a cabo con dichas tecnologías.

En el capítulo 5 se describen las conclusiones a las que hemos llegado tras la realización de este trabajo. En el capítulo 6, indicaremos las futuras líneas de investigación que completarán, en el futuro, este trabajo.

Para finalizar, en el capítulo 7 describimos brevemente el Currículum Vitae del autor de esta memoria y en el último capítulo indicaremos las referencias que hemos utilizado.

Capítulo 2

Asignaturas cursadas

En este apartado de la memoria, se indican las asignaturas que se han cursado durante el desarrollo del Máster en Tecnologías Informáticas Avanzadas de la Universidad de Castilla La Mancha. Se realizará una descripción general de cada una de ellas, especificando los objetivos, los trabajos y las prácticas que se han realizado en cada una de ellas.

2.1. Asignaturas optativas del primer cuatrimestre

2.1.1. Generación de Documentos Científicos en Informática

Descripción general

Esta asignatura ha sido impartida por los Doctores José Antonio Gámez Martín, Francisco Parreño Torres y Luis de la Ossa Jiménez, teniendo una carga lectiva de 5 Créditos ECTS.

La asignatura está dividida en tres bloques:

- **Metodología de investigación.** En este primer bloque de la asignatura, se ha hecho una introducción a los principales medios de divulgación científica (revistas, conferencias, congresos, etc.), indicándonos una serie de herramientas para la búsqueda de información, cómo estructurar y escribir correctamente un documento científico, cómo hacer una buena presentación y cómo evaluar la investigación realizada, para considerar si nuestras investigaciones son o no de calidad. También se nos introducen distintos índices para medir el impacto de una publicación y plataformas para poder recopilar dichas publicaciones como *Google Scholar*[1], *Scopus*[2] e *ISI WoS*[3].
- **Contraste de hipótesis.** En este segundo bloque, se realizó una introducción a **R**[4], que es un lenguaje y entorno de programación para análisis estadístico y gráfico, a **R-commander**, que es el interfaz gráfico de **R**, se ha realizado una introducción a los estimadores y se han explicado una serie de contrastes **paramétricos** y **no paramétricos** que posteriormente se utilizaron para realizar el trabajo final de la asignatura.
- **Composición de documentos y presentaciones con \LaTeX .** En el bloque final, se ha realizado una descripción de \LaTeX [5], que es un paquete de macros que permite al usuario componer e imprimir su trabajo con una gran calidad tipográfica, utilizando una disposición de página predeterminada. También se utilizó el paquete **Beamer**[6] para hacer la presentación del trabajo de la asignatura.

Objetivos de la asignatura

Los objetivos de esta asignatura son los que citamos a continuación:

- Introducir al alumno en el contexto investigador.
- Explicar las fuentes de información habituales en Informática (bases de datos electrónicas, repositorios, revistas, etc.).
- Describir los conocimientos básicos que debe adquirir cualquier investigador: aprender a buscar, explicar y evaluar el trabajo realizado por el investigador.
- Enseñar técnicas estadísticas que nos sirven para evaluar y analizar experimentos de una manera objetiva, utilizando conceptos estadísticos.
- Aprender a utilizar un editor de textos científicos de calidad: L^AT_EX.

Descripción del trabajo realizado

En esta asignatura se han realizado dos trabajos: el primero fue el **Cálculo del índice h en el Grupo Autolog** y el trabajo final de la asignatura titulado **Contrastes de hipótesis paramétricos y no paramétricos: Anova de un factor y Mann-Whitney**.

En el trabajo **Cálculo del índice h en el Grupo Autolog**, se calculó el *índice h* en las plataformas *ISI WoS* y *Scopus* y se estudiaron los resultados obtenidos. Para la realización de este trabajo, se trató a todos los integrantes de dicho Grupo como un único autor para realizar los cálculos y a la hora de realizar las búsquedas, se realizó una búsqueda de los investigadores principales, ya que aparecen prácticamente en la totalidad de artículos del Grupo Autolog.

Se realizó una descripción del *índice h* y cómo se calcula, haciendo también una breve introducción de las plataformas utilizadas *ISI WoS* y *Scopus* para, posteriormente, mostrar los resultados obtenidos tras la realización de las búsquedas.

Para el trabajo final de la asignatura **Contrastes de hipótesis paramétricos y no paramétricos: Anova de un factor y Man-Whitney**, se hizo una descripción de dichos contrastes, indicando cómo habría que realizar los cálculos de cada uno de ellos en *R-commander*, realizando un ejemplo de aplicación para cada test utilizado.

Para el test *Anova de un factor* utilizamos 24 muestras una por cada hora del día, divididas en tres partes y una columna en la que recogíamos el ruido que generaban los vehículos al pasar por una determinada zona de Cuenca. Nuestra hipótesis quería comprobar si las partes del día afectan o no al ruido que generan los vehículos cuando pasan por dicho punto.

Para el test *Mann-Whitney*, que para realizarlo con *R-commander* teníamos que utilizar el test de *Wilcoxon* ya que es equivalente, también utilizábamos 24 muestras y el número de vehículos que pasaban por dicha zona de Cuenca en dos días distintos. En este caso, nuestra hipótesis quería comprobar si el número de coches que pasa por allí es el mismo los dos días.

2.1.2. Introducción a la Programación de Arquitecturas de Altas Prestaciones

Descripción general

Esta asignatura ha sido impartida por los Doctores Enrique Arias Antúnez, Diego Cazorla López y Juan José Pardo Mateo, teniendo una carga lectiva de 5 Créditos ECTS.

La asignatura está dividida en dos bloques: un bloque en el que se encuentra la **parte teórica** de la asignatura y otro bloque de la **parte práctica**.

En la **parte teórica** se tratan los siguientes temas, que son la base para la realización posterior de las prácticas:

- Introducción a la programación de arquitecturas de altas prestaciones.
- Programación orientada a bloques.
- Introducción a las arquitecturas paralelas.
- Paradigmas de computación paralela.
- Diseño de programas en arquitecturas paralelas.
- Software de programación de arquitecturas de altas prestaciones.

En la **parte práctica** tratábamos los siguientes temas que citamos a continuación:

- Optimizaciones secuenciales.
- Programación paralela con *MPI* y *BLAS*.
- Programación paralela con *BLACS* y *PBLAS*.

Objetivos de la asignatura

Los objetivos de la asignatura son los que indicamos a continuación:

- Presentar las ideas básicas de la programación secuencial orientada a bloques.
- Presentar los modelos y las ideas básicas de la computación paralela.
- Adquirir una metodología de diseño y evaluación de los algoritmos paralelos.
- Introducir diversas librerías de programación e arquitecturas de altas prestaciones orientadas a la resolución de problemas de álgebra lineal numérica.
- Realizar diferentes sesiones prácticas para consolidar los conceptos que han sido introducidos en teoría.

Descripción del trabajo realizado

Para esta asignatura, se ha realizado un trabajo titulado **UPC: Unified Parallel C**[7], que ha consistido en realizar una introducción a dicho lenguaje de programación paralela, siendo una extensión de *ANSI C*[8].

En este trabajo se han comentado algunas características de *UPC*, como el uso de un modelo de programación con particiones del espacio de direcciones global de la memoria, modelo que se conoce como *modelo de distribución de programación de memoria compartida*. *UPC* mantiene los conceptos y características del *lenguaje C*[9], agregándole paralelismo, acceso a memoria global, con una comprensión de lo que es memoria remota y memoria local, y la posibilidad de leer y escribir en memoria remota con una serie de instrucciones sencillas.

Se han descrito los distintos parámetros de *UPC*: modelo de programación, modelo de memoria, cómo se distribuyen los datos y cómo conseguimos que estos datos sean coherentes y el uso de *APIs* para poder simplificar lo máximo posible la entrada y salida paralela.

También se han indicado los tipos de datos que se utilizan en *UPC*, el uso de punteros y una serie de librerías de funciones para facilitar el trabajo a los desarrolladores de aplicaciones con este lenguaje.

Descripción de las prácticas

Como se ha especificado anteriormente, la parte de prácticas está dividida en el desarrollo de tres programas. El problema a resolver consistía en realizar un **producto de matrices paralelas** y calcular el tiempo necesario para realizar dicho producto, con matrices cuadradas, matrices no cuadradas y distintos tamaños de bloque, según la práctica.

- En la **primera práctica**, se han realizado diferentes versiones del producto de matrices secuencial, que consistía en un producto de matrices básico, otro código que realice el producto de matrices bloque y para finalizar teníamos que utilizar una serie de funciones de *BLAS*.
- En la **segunda práctica**, se han utilizado una serie de procesos y en cada proceso se realizaba una tarea. También teníamos que utilizar una función para poder realizar el producto de matrices.
- En la **tercera práctica**, se han usado *PBLAS* y *ScaLAPACK* para la programación del algoritmo utilizando, como en la práctica anterior, una serie de procesos para realizar las distintas tareas necesarias para realizar el producto de matrices.

2.1.3. Nuevos Paradigmas en HCI

Descripción general

Esta asignatura ha sido impartida por los Doctores Antonio Fernández Caballero, María Teresa López Bonal y José Pascual Molina Massó, teniendo una carga lectiva de 5 Créditos ECTS.

La asignatura estaba dividida en cuatro bloques:

- En el primer bloque, se han explicado los **Interfaces de Usuario en 3D**, sus fundamentos, aspectos de diseño y cómo programar dichos interfaces 3D. Se realizó un recorrido sobre la evolución de estos interfaces y se presentó una metodología para el desarrollo de interfaces y mundos 3D.
- El segundo bloque se ha dedicado al estudio de la **Visión Artificial** y en él se han mostrado algunas técnicas y algoritmos habituales para su desarrollo, así como algunos ejemplos de trabajos desarrollados por el **Grupo LoUISE**[10] de la UCLM.
- En el tercer bloque se han estudiado los **Agentes Software**, definiendo el concepto de Sistema Multiagente, haciendo un recorrido sobre los distintos tipos de agentes, su uso y cómo desarrollar dichos agentes.
- El último bloque se ha dedicado a la descripción de la **Computación Afectiva**, profundizando en los distintos aspectos relacionados con los interfaces de usuario y las peculiaridades respecto a la interacción con este tipo de plataformas.

Objetivos de la asignatura

Los objetivos de la asignatura son los que indicamos a continuación:

- Presentar los aspectos más relevantes de la visión artificial en la interacción hombre-máquina.
- Detectar, segmentar, localizar y reconocer objetos en imágenes.
- Realizar el seguimiento de un objeto en una secuencia de imágenes.
- Mapear una escena para generar un modelo tridimensional de dicha escena.
- Interactuar con entornos de realidad virtual, realidad aumentada y computación ubicua.
- Conceptos generales a los Sistemas Multiagente asociados a la interacción hombre-máquina (agentes de interfaz y agentes para Internet).

Descripción del trabajo realizado

Para esta asignatura, se ha realizado un trabajo titulado **Sistemas Multiagente en entornos industriales dinámicos no deterministas**.

Este trabajo se ha comenzado con una introducción de por qué los sistemas de producción y control necesitan un nuevo enfoque dentro de las actividades de gestión y control de las empresas, incidiendo en la necesidad de actualización de los sistemas monolíticos y poco flexibles que existe en muchas de las empresas actuales.

También se han indicado los grandes problemas a los que se enfrentan las empresas, es decir, sistemas distribuidos, necesidad de tener los recursos y productos monitorizados en todo momento para el correcto funcionamiento de dicha empresa y la gestión de la gran cantidad de información que generan las empresas diariamente. Para poder solucionar estos problemas, han surgido nuevas tecnologías que nos permiten disminuir en gran medida las dificultades actuales de producción y control.

Uno de estos avances es la *Identificación por Radiofrecuencia*, o *RFID* en sus siglas en inglés, tecnología que nos permite identificar de forma fácil y eficaz nuestros productos y recursos, sin que afecte al funcionamiento normal de la empresa.

Otra de las tecnologías propuestas para la gestión de la información son los *Sistemas Multiagente*, que es un área dentro de la inteligencia artificial dirigida a los *Sistemas Distribuidos*. Con los agentes no sólo podemos gestionar la información para el normal funcionamiento de la empresa, sino también generar conocimiento y sistemas expertos para obtener mejores soluciones.

La idea del trabajo era la de proporcionar una *Arquitectura Multiagente* para la gestión y el control de entornos industriales dinámicos y no deterministas para, posteriormente, ampliarla con nuevas tecnologías para la toma de decisiones dentro de una empresa. Estas nuevas tecnologías abarcan desde dispositivos móviles, *GPS*, *Google Maps*, triangulaciones Wifi, geolocalización y Redes de Sensores.

En base a esto explicamos también la *Arquitectura IMS-DUO*, arquitectura de agentes en la que se diferencian dos puntos de vista: uno más relacionado con la parte física y otro más cercano a los sistemas de información de la empresa.

Cada uno de estos puntos de vista da lugar a dos tipos de sociedades de agentes: *Sociedad de Agentes Físicos* y *Sociedad de Agentes del Sistema de Información*, sociedades que pueden comunicarse entre ellas y que se utilizan en nuestra *Arquitectura Multiagente*.

2.2. Asignaturas optativas del segundo cuatrimestre

2.2.1. Calidad de Interfaces de Usuario: Desarrollo Avanzado

Descripción general

Esta asignatura ha sido impartida por los Doctores Pascual González López, Víctor Manuel López-Jarquero y Francisco Montero Simarro, teniendo una carga lectiva de 5 Créditos ECTS.

La asignatura está dividida en tres bloques:

- En el primer bloque, se ha introducido al alumno a los principios y bases de las interfaces de usuario y a la disciplina que se encarga del estudio de la interacción de las personas con los interfaces de usuario *IPO* (Interacción Persona-Ordenador). Para ello, se han descrito los principales conceptos que se tendrán que tener en cuenta a la hora del desarrollo de cualquier tipo de interfaz.
- En el segundo bloque, se hace una introducción a un enfoque formal basado en modelos para desarrollar interfaces de usuario, indicando qué modelos se utilizan (de tareas, aplicación, usuario, presentación y diálogo) para desarrollar una interfaz. También se han introducido una serie de lenguajes de descripción para interfaces de usuario y una serie de transformaciones de modelos para el desarrollo.
- El tercer y último bloque se ha centrado en la calidad y desarrollo de interfaces de usuario, se han explicado conceptos como la usabilidad, calidad de uso, cómo lograr la calidad e integrar la calidad en el desarrollo de interfaces de usuario basado en modelos. También se han comentado una serie de técnicas y herramientas para poder evaluar la usabilidad de un sistema.

Objetivos

Los objetivos de esta asignatura, son los que citamos a continuación:

- Resaltar el papel que juegan y la importancia que tienen las interfaces de usuario en los productos software.
- Caracterizar el desarrollo de interfaces de usuario y ligarlo al concepto de modelo.
- Caracterizar la calidad de una interfaz de usuario.
- Prestar especial atención a requisitos no explícitamente funcionales como son la usabilidad, la adaptación y la colaboración.
- Difusión del trabajo que se está realizando en un grupo de investigación de la Universidad de Castilla la Mancha, Grupo LoUISE, directamente relacionado con las interfaces de usuario.
- Propuesta y difusión de líneas de investigación relacionadas con el desarrollo sistemático de interfaces de usuario.

Descripción del trabajo realizado

Para esta asignatura se ha realizado un trabajo que titulamos **Internet de las Cosas e Interfaces de Usuario**.

En este trabajo, se hizo una introducción al concepto Internet de las Cosas, explicando qué es este concepto y qué son las llamadas *Cosas*, qué son los objetos inteligentes u objetos susceptibles de ofrecernos cierto tipo de información pudiendo interactuar con ellos y que ámbitos pueden verse afectados por dichos objetos, indicando también el alcance de esta tecnología y en qué entornos se está aplicando y en cuáles podría aplicarse.

Se explican los conceptos de *mundo real*, *mundo virtual* y cómo están relacionados entre sí, así como una serie de tecnologías que están relacionadas con Internet de las Cosas, las cuales pueden proporcionar a los *objetos* esa inteligencia de la que hablábamos en el párrafo anterior. Estas tecnologías son: *RFID*, los códigos *QR* y *NFC*, *Near Field Communication*.

También se indica cómo interactuar con los objetos inteligentes, explicando que a la hora de interactuar con ellos, con el *mundo virtual*, utilizamos el *mundo real* como interfaz y cuales son los parámetros que pueden darnos *calidad de interacción* con dichos objetos y cómo podemos mejorar la experiencia y satisfacción del usuario a la hora de realizar esa interacción.

Para finalizar, comentamos algunos ejemplos de interacción con *objetos inteligentes*, véase una *cafetera RFID*, la cual, gracias a un *tag RFID* incluido en la taza, sabe a qué persona pertenece dicha taza y los gustos del propietario, tipo de café, azúcar, cantidad de leche, etc... Además, esta cafetera también podría controlarse remotamente. Un segundo ejemplo serían los *hogares/lugares domóticos*, los cuales podrían detectar la posición del usuario en una determinada sala y así enviarle algún tipo de información.

2.2.2. Programación Internet con Lenguajes Declarativos Multiparadigma

Descripción general

Esta asignatura ha sido impartida por los Doctores Ginés Moreno Valverde, Pascual Julián Iranzo y Francisco Pascual Romero Chicharro, teniendo una carga lectiva de 5 Créditos ECTS.

Esta asignatura se ha dividido en tres bloques, que comentamos a continuación:

- En el primer bloque se han mostrado los conceptos principales tanto de la **programación lógica** como de la **programación lógica difusa**. También se han comentado mecanismos básicos aplicables a la programación lógica, como son la unificación y la resolución, y cual es la semántica necesaria para trabajar en entornos lógicos, semántica que se divide en operacional y declarativa. Una vez conocidas las bases de la programación lógica, se pasa a la programación lógica difusa, explicándose la teoría y los fundamentos de ésta, basándose concretamente en la parte de *resolución débil*.
- El segundo bloque se han introducido la forma en la que la programación lógica e Internet pueden relacionarse entre sí. Además, se muestra una aproximación multiadjunta de la programación lógica difusa: un marco general, transformaciones entre programas e implementación.
- En el último bloque, se extiende la aplicación de la programación lógica sobre Internet.

Objetivos

Los objetivos de esta asignatura, son los que citamos a continuación:

- Conocer los fundamentos y características de los lenguajes declarativos multiparadigma, que integran la programación lógica con el paradigma difuso (fuzzy).
- Estudiar los mecanismos operacionales y la semántica de algunas de las propuestas de integración. También, técnicas de transformación de programas que permiten optimizarlos.
- Conocer las facilidades de un lenguaje declarativo concreto para la programación Internet.
- Presentar el tópico de la búsqueda en Internet desde una perspectiva declarativa.

Descripción del trabajo realizado

Para esta asignatura, se ha realizado un trabajo que titulamos **XWT: XML Wavelet Tree. Implementación eficiente de una extensión para XPath**.

Este trabajo ha consistido en la explicación de *XWT*, representación comprimida y autoindexada de *XML*, extensión para *XPath* que tiene el objetivo de permitir la especificación flexible del contenido del texto y del documento. Será el usuario el que decida el grado de flexibilidad para identificar la información más importante.

Una de las ventajas que nos ofrece *XWT* es que nos devuelve un *ranking* de coincidencias con los resultados obtenidos, mientras que *XPath* sólo produce una serie de resultados.

Para ello se reorganiza el código *XML*, reorganización que consiste en colocar los distintos bytes de las palabras clave en los distintos nodos del árbol, en lugar de concatenarlos secuencialmente, como se haría en un típico texto comprimido. Para posibilitar esta reorganización se utiliza el **compresor (s-c)-DC**, que codifica cada palabra como una secuencia de *bytes*, de tal forma que permite seleccionar un rango de estos *bytes* para utilizarlos como los últimos de las palabras clave.

Otra de las ventajas que nos ofrece *XWT* es que consigue representar cualquier documento *XML* utilizando un 35 % del tamaño original de *XML* y realiza la reordenación de *bytes* en muy poco tiempo.

A su vez, se ha explicado que, a la hora de realizar esta reordenación y la posterior representación, se crean dos tipos de vocabularios: uno con las **etiquetas** y otro con el **resto de palabras** del documento. Con esto se consigue una mayor eficiencia y mejor compresión del documento.

Se explicaron los procedimientos básicos que se realizan con *XWT*, que son el de **buscar** una o varias palabras en el texto y el de **decodificar** una palabra o conjunto de palabras que están en una determinada posición. Para ello, se utilizan las instrucciones **select** y **rank** respectivamente.

Se incluyen también una serie de restricciones, llamadas **near** y **below** que se utilizan para poder realizar de forma eficiente las búsquedas dentro del documento. Se puede decir que *below* es una generalización de *near*.

Una de las características de esta extensión consiste en que cualquier operación que se realice sobre un *XWT*, también podría realizarse sobre *XML* directamente, solo que no sería tan eficiente la realización de dichas operaciones, al perder las propiedades de compresión y auto-indexación que nos ofrece *XWT*.

Para finalizar el trabajo, se realizaron una serie de ejemplos para poder ver la eficiencia y rapidez de las búsquedas en documentos *XML* de distintos tamaños.

2.2.3. Tecnología Software Orientada a Objetos

Descripción general

Esta asignatura ha sido impartida por los Doctores Elena María Navarro Martínez, María Dolores Lozano Pérez y Víctor Ruiz Penichet, teniendo una carga lectiva de 5 Créditos ECTS.

La asignatura está dividida en tres módulos, que son los que comentamos a continuación:

- En el primer módulo, se ha realizado una introducción a *DSDM* (**Desarrollo Software Dirigido por Modelos**) y se ha hecho un análisis de la propuesta *MDA* (**Model Driven Architecture**) y los distintos niveles de abstracción que se proponen para el modelado de un sistema software.

Se han analizado las distintas propuestas existentes de *MDA* para la industria, siendo las *Software Factories* de *Microsoft* y algunos entornos de *Eclipse Modeling Project* algunas de éstas y se han realizado una serie de prácticas dónde se han utilizado los *frameworks* *Eclipse Modeling Framework*, *EMF*, *Graphical Modeling Framework*, *GMF* y transformaciones *Model to Model*, *M2M*.

- En el segundo módulo, se ha realizado una introducción a *MB-UIDE* (**Model-Based User Interface Development Environment**), describiendo las tendencias actuales en el Desarrollo de Interfaces de Usuario y las distintas generaciones por las que ha pasado el desarrollo de interfaces de usuario.

Se ha explicado el Entorno Metodológico Basado en Modelos para el Desarrollo de Interfaces de Usuario, indicando los distintos modelos existentes para el desarrollo de interfaces de usuario (modelo de Tareas, modelo de Dominio o Aplicación, modelo de Usuario, modelo de Diálogo, modelo de Presentación) y los tipos de objetos de interacción abstractos *AIOs* y concretos *CIOs*. Otro tema que se trató en este módulo son las tendencias actuales de desarrollo de interfaces de usuario y se realizó un ejemplo de un caso de estudio.

- En el último módulo, se ha realizado una introducción del **Modelo de Procesos para el Desarrollo de Interfaces de Usuario Colaborativas**, describiendo el estado del arte, introduciendo los conceptos básicos de los entornos *CSCW* y su modelo conceptual, la herramienta *TOUCHE CASE Tool* y se desarrolló un ejemplo de un caso de estudio.

Objetivos

Los objetivos de esta asignatura, son los que citamos a continuación:

- Definir formalmente el concepto de modelo. Aplicación a diferentes campos (tecnologías): *MDA*, *MOF*, compilación de modelos, interoperabilidad, semántica, plataformas de gestión de modelos, persistencia. Aplicación práctica de los conceptos introducidos.
- Conocer los modelos específicos empleados para el desarrollo de Interfaces de Usuario y su trazabilidad a lo largo del proceso de desarrollo. Aprender las bases de *MB-UIDE* (*Model-Based User Interface Development Environments*) y algunas de sus aplicaciones. Conocer algunas extensiones al lenguaje de modelado *OO UML* para Sistemas Interactivos e Interfaces de Usuario.
- Conocer la problemática de los nuevos entornos colaborativos y las soluciones actuales aplicadas a este tipo de entornos. nuevos métodos y herramientas de desarrollo de sistemas colaborativos (*CSCW*, *Groupware*) y su aplicación en casos concretos.

Descripción del trabajo realizado

Para esta asignatura, se ha realizado un trabajo que titulamos **Metodologías para el desarrollo de SMA a través de MDD**.

En este trabajo se ha indicado que la investigación dentro del campo de los *SMA* (Sistemas Multiagente) es un área muy activa, pero sólo una pequeña parte de esta investigación se dedica al desarrollo de herramientas adecuadas para el desarrollo de sistemas basados en agentes.

Se ha realizado una descripción de cuatro de las metodologías actuales dedicadas al **Desarrollo de Sistemas Basados en Agentes** que utilizan un **Desarrollo Basado en Modelos**. Las metodologías seleccionadas son: *PIM4AGENTS*, *DSML4MAS*, *INGENIAS* y *VigilAgent*.

Una vez descritas y comentadas cada una de las metodologías se ha realizado una comparación entre ellas, indicando las similitudes y diferencias existentes. Indicar que *DSML4MAS* está basada en la metodología *PIM4AGENTS* y que *VigilAgent* utiliza lo mejor de tres metodologías, una de ellas es *INGENIAS*, para su desarrollo.

El objetivo final de las distintas metodologías, es el de facilitar el desarrollo de *SMA* a través de modelos reduciendo la complejidad de los procesos necesarios para el desarrollo, ser independientes de la plataforma de destino y seguir un proceso, más o menos estructurado, para realizar el desarrollo.

Descripción de las prácticas

Asociadas a los temas de teoría de los módulos 1 y 2 de esta asignatura, se han realizado una serie de ejercicios prácticos, en los que se ha utilizado el entorno de programación *Eclipse* [11] y herramientas de modelado *Eclipse Modeling Tools* [12], para así complementar las clases de teoría.

Capítulo 3

Estado del arte

En este capítulo, se ha realizado una descripción de las tecnologías relacionadas con nuestras investigaciones.

Para poder desarrollar nuestro trabajo el primer paso es el de identificar las tecnologías relacionadas con la línea de investigación que se ha elegido. De esta forma, será más fácil el encontrar algún problema que sea necesario resolver o dar una solución distinta a algún problema que ya ha sido resuelto.

Tras el paso de identificación de las tecnologías se realizará un estudio de éstas, para así obtener información de cada una de ellas y sobre las investigaciones o trabajos que se han realizado al utilizarlas, obteniendo así un punto de partida para nuestras investigaciones.

Las tecnologías relacionadas necesarias para el desarrollo de nuestras investigaciones y que serán descritas a lo largo de este capítulo, son las que citamos a continuación:

- Redes de Sensores Inalámbricos (*WSN*).
- *ZigBee*.
- Servicios Web (*Web Services*).
- *EPC Information Services*.
- Internet de las Cosas (*Internet of Things*).
- Localización-Geolocalización.

3.1. Redes de Sensores Inalámbricas (WSN, Wireless Sensor Network)

3.1.1. Introducción

Las **Redes de Sensores Inalámbricas** se definen como un conjunto de dispositivos autónomos encargados de monitorizar las condiciones ambientales o sucesos de entorno, a través de un conjunto de sensores embebidos que disponen de transceptores de radio para la creación de redes *ad-hoc*¹ sin infraestructura física predefinida [13].

Las Redes de Sensores se aprovechan de las capacidades del estándar **IEEE 802.15.4** para definir estructuras de monitorización inalámbricas fácilmente configurables y expansibles de bajo consumo. De este modo se consiguen unos costes de instalación y mantenimiento mínimos.

Además, como veremos en el punto 3.2, gracias a la interoperabilidad de dispositivos con distintos perfiles de aplicación *ZigBee*, ya sean definidos por *ZigBee Alliance* o por otra asociación, cualquier nuevo subsistema de red basado en *ZigBee* puede ser incorporado a cualquier otra red *ZigBee* ya preestablecida.

De este modo, los nuevos elementos introducidos en la red, aprovechan la infraestructura *LR-WPAN* establecida para comunicar sus eventos y también contribuirán a aumentar las capacidades de enrutamiento de la red para aquellos elementos definidos como *routers*, propiciando así la expansión de actuación de sensores, no siendo necesario crear canales formados por dispositivos dedicados a una misma aplicación.

3.1.2. Componentes

Las Redes de Sensores cuentan con un **coordinador** que está encargado de generar la red actuando, en la mayoría de los casos, como puerta de enlace (*Gateway*) para la conexión con un equipo o para la conexión con otras redes para realizar una supervisión remota de la red inalámbrica de sensores.

Otro dispositivo de la red es la **estación base**, que se encarga de recoger todos los datos que son enviados por los nodos. Como estación base, suele utilizarse un ordenador común o un sistema embebido.

Además, también disponemos de **sensores** que se encargan de obtener la información del medio y convertir esa información en señales eléctricas. Los **nodos**, toman los datos de los sensores y los envían a la estación base.

Uno de los objetivos de estos dispositivos es el de optimizar la vida útil de la batería mediante cortos ciclos de actividad para reportar la información captada mediante los sensores embebidos.

Podemos ver un ejemplo de la arquitectura básica de una red inalámbrica en la figura 3.1.

¹Una red *ad-hoc* es una red inalámbrica descentralizada en la que cada nodo está preparado para reenviar datos a los demás y la decisión sobre qué nodos reenvían los datos se toma de forma dinámica en función de la conectividad de la red.

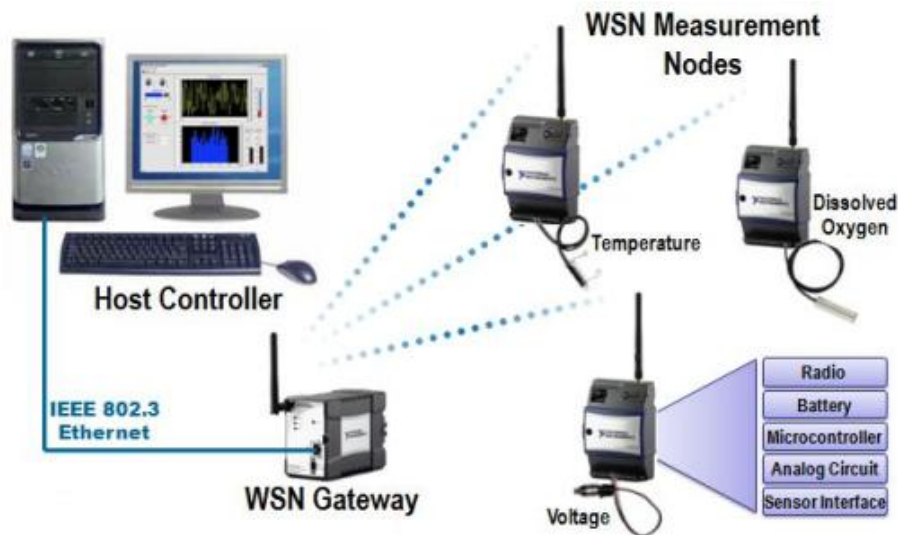


Figura 3.1: Ejemplo de la arquitectura de una red de sensores

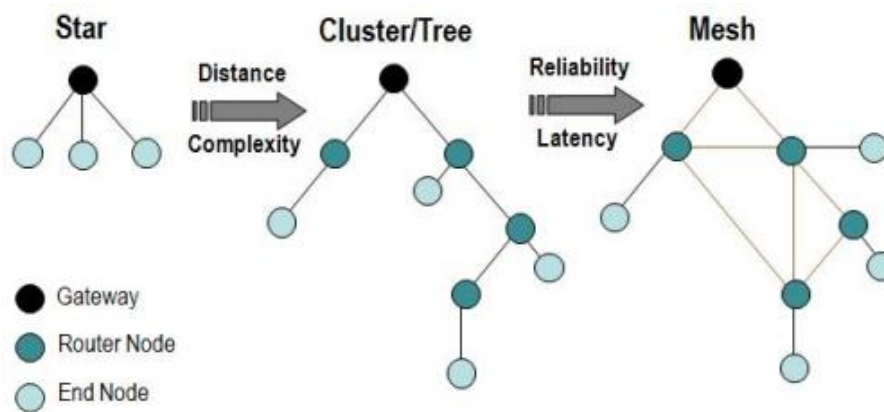


Figura 3.2: Topologías Redes de Sensores

3.1.3. Tipología básica

Los nodos de las Redes de Sensores inalámbricas, están típicamente organizados en tres topologías de red [14]:

- **Topología en estrella**, en la que cada nodo se conecta directamente al coordinador.
- **Topología en árbol**, en la que cada nodo se conecta a otro nodo de mayor jerarquía dentro de dicho árbol y después al coordinador. Los datos se envían desde el nodo de menor jerarquía hasta el coordinador.
- **Redes tipo malla**, cuya principal característica es que los nodos se pueden conectar a múltiples nodos dentro del sistema y pasar los datos por el camino disponible de mayor confiabilidad.

En la figura 3.2, podemos ver una representación de las distintas topologías explicadas.

3.1.4. Campos de aplicación

La lista de aplicaciones donde las Redes de Sensores inalámbricas encuentra utilidad puede ser ingente y abierta a la creatividad de los desarrolladores [15]. Dejando a un lado las aplicaciones militares, las Redes de Sensores inalámbricas podrían utilizarse en los siguientes ámbitos [16]:

- **Eficiencia energética.** Redes de Sensores utilizadas para controlar el uso eficaz de la electricidad.
- **Entornos de alta seguridad.** Lugares en los que se requieren altos niveles de seguridad para evitar cualquier tipo de ataque, como pueden ser centrales nucleares, aeropuertos, edificios del gobierno, etc.
- **Sensores ambientales.** Control ambiental de bosques y océanos. Se podrían controlar múltiples variables, como temperatura, humedad, fuego, actividad sísmica, etc.
- **Sensores industriales.** Para realizar complejos sistemas de control de calidad.
- **Automoción.** Las Redes de Sensores suelen ser un complemento ideal para las cámaras de tráfico, ya que pueden informar de la situación del tráfico en ángulos muertos que no cubren dichas cámaras y también pueden informar a los conductores de alguna incidencia en su camino.
- **Domótica.** Debido a su tamaño, economía y velocidad de despliegue, hacen de las Redes de Sensores una tecnología ideal para domotizar un hogar a un precio bastante asequible.
- **Medicina.** Para tener controladas las constantes vitales de pacientes en cualquier momento, localizar personas con algún tipo de enfermedad o minusvalía, etc. Es un campo bastante prometedor.

En nuestro caso, como ya indicamos anteriormente, utilizaremos la Red de Sensores Inalámbricas para la localización de personas con algún tipo de enfermedad grave o minusvalía en entornos abiertos y así podremos tener su localización exacta en cualquier momento.

3.2. ZigBee

3.2.1. Introducción

ZigBee [17] [18] es un estándar de comunicaciones inalámbricas diseñado por **ZigBee Alliance** [19], que es un conjunto de más de 70 compañías que se han asociado para crear el estándar *ZigBee*. No es una tecnología propiamente dicha, sino un conjunto de protocolos que pueden ser implementados por cualquier fabricante.

ZigBee está basado en el estándar **IEEE 802.15.4** [20] de Redes Inalámbricas de área personal (*Wireless Personal Area Network, WPAN*) y tiene como objetivo las aplicaciones que requieren comunicaciones seguras, con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

La especificación **1.0** de *ZigBee* fue aprobada en diciembre de 2004. En la actualidad se utiliza la revisión que se realizó en el año 2006, aprobada en diciembre de ese mismo año.

3.2.2. Características

Algunas de las características de *ZigBee* son las que citamos a continuación:

- Uso de la banda *ISM*², en concreto se utilizan las bandas de 868 MHz en Europa, 915 MHz en Estados Unidos y 2'4 GHz en todo el mundo, siendo esta última banda la más utilizada en entornos industriales.
- Tiene una velocidad de transmisión de 250 Kbps y un rango de cobertura que va desde 10 a 75 metros.
- Cada red *ZigBee* tiene un identificador de red único, lo que permite que coexistan varias redes en un mismo canal de comunicación sin ningún problema.
- Capacidad de operar en redes de gran densidad de nodos.
- *ZigBee* es un protocolo de comunicación *multisalto*, es decir, se puede establecer comunicación entre dos nodos aún cuando éstos se encuentren fuera del rango de transmisión, siempre que existan otros nodos intermedios que los interconecten.
- Su topología en malla, permite a la red auto-recuperarse de problemas en la comunicación aumentando su confiabilidad.

3.2.3. Dispositivos

Se definen tres tipos de dispositivos *ZigBee* según su papel en la red:

- **Coordinador ZigBee (ZC)**. Es el tipo de dispositivo más completo y debe existir uno por red. Sus funciones son las de encargarse de controlar la red y los caminos que deben seguir los dispositivos para conectarse entre ellos.
- **Router ZigBee (RZ)**. Interconecta dispositivos separados en la topología de la red, además de ofrecer un nivel de aplicación para la ejecución de código de usuario.

²*Industrial, Scientific and Medical*, son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, médica y científica.

- **Dispositivo Final Zigbee (ZED)**. Posee la funcionalidad necesaria para comunicarse con su nodo padre (coordinador o *router*), pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar en reposo la mayor parte del tiempo, aumentando la vida media de sus baterías.

Basándose en su funcionalidad, podemos realizar una segunda clasificación de dispositivos:

- **Dispositivos de funcionalidad completa (FFD)**, también conocidos como nodos activos. Estos dispositivos son capaces de recibir mensajes en formato *IEEE 802.15.4*. Gracias a la memoria adicional que poseen y a la capacidad de realizar operaciones, pueden funcionar como coordinador o *router ZigBee* o usados en dispositivos de red que actúen de interfaz con los usuarios.
- **Dispositivos de funcionalidad reducida (RFD)**, también conocidos como nodos pasivos. Tienen capacidad y funcionalidad limitadas con el objetivo de conseguir un bajo coste y una gran simplicidad. Básicamente, son los sensores/actuadores de la red *ZigBee*.

Un nodo *ZigBee*, tanto activo como pasivo, reduce su consumo gracia a que puede permanecer en reposo la mayor parte del tiempo. Cuando se requiere su uso, dicho nodo es capaz de despertar en un tiempo ínfimo, para volver al estado de reposo una vez deja de ser requerido.

3.2.4. Arquitectura

El estándar *ZigBee* es una pila de protocolos que, de manera similar al *modelo OSI*³ está constituido por una serie de capas, cada una independiente de la otra. En la figura 3.3 podemos ver la distribución del modelo *ZigBee*.

La capa de más bajo nivel, la **capa física (PHY)** es la encargada de ejercer de interfaz con el medio de transmisión (en este caso, radio) y comunicarse con la capa superior **capa de control de acceso al medio (MAC)**. Estas dos capas son definidas por el estándar *IEEE 802.15.4*. Las funciones de control del medio de transmisión de la capa física son las siguientes:

- Establecimiento del canal.
- Configuración a nivel de bit: modulación, codificación y sincronización.

La **capa MAC** se ubica dentro de la capa de enlace y sus funciones básicas son las siguientes:

- Provee de servicios de asociación/disociación a la red *PAN*.
- Establece mecanismos de acceso al canal.
- Generar balizas en el caso de ser un coordinador *PAN* cuando existen requerimientos de sincronización.
- Se encarga de la gestión de tiempo de acceso garantizados.

³Modelo de interconexión de sistemas abiertos, es el modelo de red descriptivo utilizado por la Organización Internacional para la Estandarización den el año 1984. Es un modelo de red que divide a la red en siete capas: aplicación, presentación, sesión, transporte, red, enlace de datos y física.

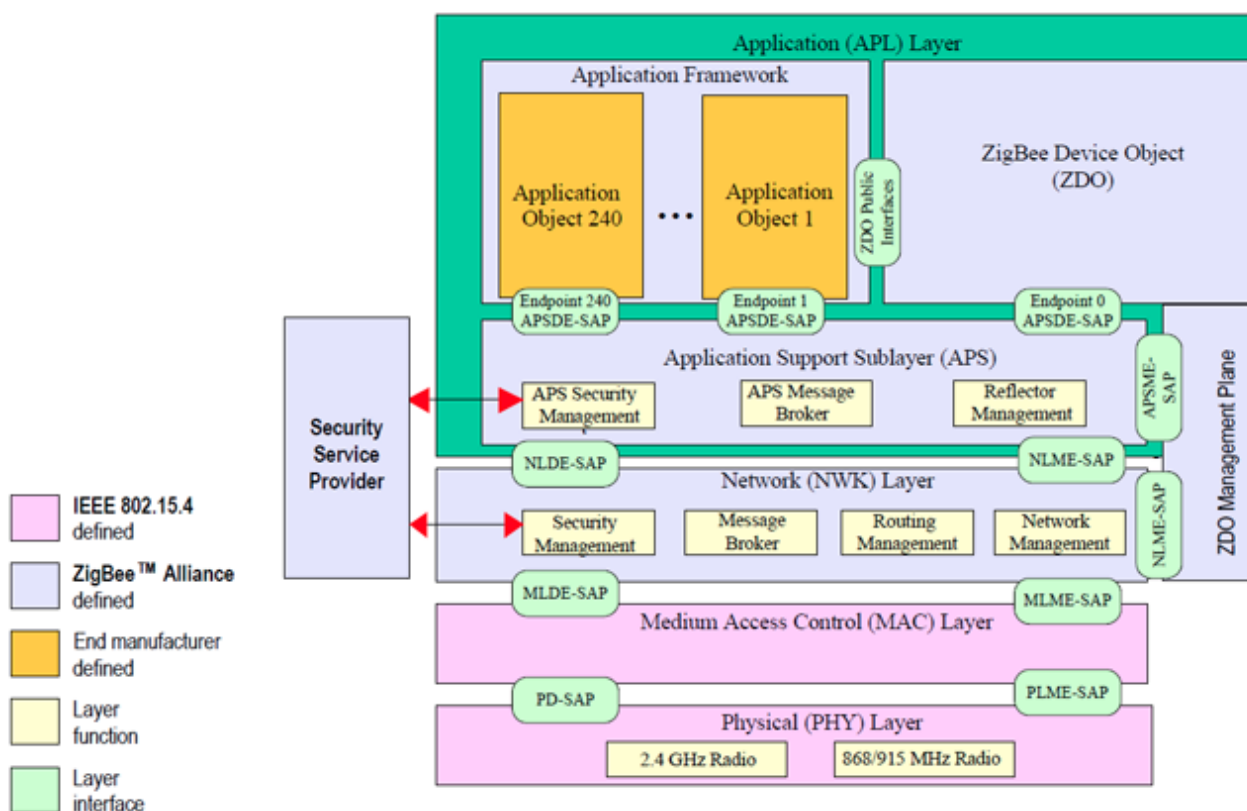


Figura 3.3: Modelo de capas de Zigbee

La **capa de red** (*NWK*) tiene como objetivo principal permitir el correcto uso del subnivel *MAC* y ofrecer una interfaz adecuada para su uso por parte de la capa de aplicación. En esta capa, se brindan los métodos necesarios para:

- Iniciar la red.
- Unirse a la red.
- Enrutar paquetes dirigidos a otros nodos en la red.
- Proporcionar los medios para garantizar la entrega del paquete al destinatario final.
- Filtrar paquetes recibidos, cifrarlos y autentificarlos.

El algoritmo de enrutamiento que usa esta capa es el de *enrutamiento en malla*, el cual se basa en el protocolo *Ad Hoc On-Demand Vector Routing (AODV)*. Cuando esta capa se encuentra cumpliendo la función de unir o separar dispositivos a través del controlador de red, implementa seguridad y encamina tramas a sus respectivos destinos.

Además, esta capa es responsable de crear una nueva red y asignar direcciones a los dispositivos de la misma. Es en esta capa donde se implementan las distintas topologías de red que soporta *ZigBee*: topología en árbol, topología en estrella y topología en malla.

La siguiente capa es la **capa de soporte a la aplicación** (*APS*) que es responsable de mantener el rol que el nodo juega en la red, es decir, filtrar paquetes de aplicación, mantener la relación de grupos y dispositivos con los que la aplicación interactúa y simplificar el envío de datos a los diferentes nodos de la red. Las capas *NWK* y *APS* son definidas por *ZigBee Alliance*.

En el nivel conceptual más alto, se encuentra la **capa de aplicación** (*APL*) es decir, la aplicación en sí misma y de la que se encargan los distintos fabricantes. En esta capa se encuentran los *ZigBee Device Objects* (*ZDO*) que son los encargados de definir el papel del dispositivo en la red, si actuará como coordinador, enrutador o dispositivo final, la subcapa *APS* y los objetos de aplicación definidos por cada uno de los fabricantes.

Cada capa se comunica con sus capas subyacentes a través de una interfaz de datos y otra de control. Las capas superiores solicitan servicios a las capas inferiores y éstas envían los resultados a las superiores. Además de las capas mencionadas, están integrados otro par de módulos: **módulo de seguridad**, que se encarga de proveer los servicios para cifrar y autenticar los paquetes, y el **módulo de administración del dispositivo**, que es quien se encarga de administrar los recursos de la red del dispositivo local, además de proporcionar a la aplicación funciones de administración remota de red.

3.2.5. Topologías de red

ZigBee permite, como se indicó anteriormente, tres topologías de red[21]:

- **Topología en estrella.** El coordinador se sitúa en el centro.
- **Topología en árbol.** El coordinador será la raíz del árbol.
- **Topología de malla.** Al menos uno de los nodos tendrá dos o más conexiones.

La topología más interesante es la de malla ya que permite, si un nodo del camino falla, poder seguir la comunicación entre el resto de nodos por otro camino.

Existen varias **estrategias de conexión** de los dispositivos en una red *ZigBee*. Las redes *ZigBee* han sido diseñadas para conservar la potencia en los nodos esclavos, consiguiendo un bajo consumo de potencia. La estrategia consiste en que, durante mucho tiempo, un dispositivo esclavo se encuentra en reposo, de tal forma que sólo se activa por muy poco tiempo para confirmar que *está vivo* en la red de la que forma parte.

En estas redes, se pueden usar dos tipos de entornos o sistemas:

- **Con balizas.** Es un mecanismo de control del consumo de potencia en la red que permite a todos los dispositivos saber cuándo pueden transmitir. En este modelo, los dos caminos de la red tienen un distribuidor que se encarga de controlar el canal y dirigir las transmisiones. Las balizas, que dan nombre a este entorno, se usan para poder sincronizar todos los dispositivos que conforman la red. Este método es recomendable cuando el coordinador de red trabaja con una batería.
- **Sin balizas.** Se usa el acceso múltiple al sistema *ZigBee* en una red punto a punto cercano. En este tipo, cada dispositivo es autónomo, pudiendo iniciar una conversación en la cual los otros pueden interferir. Puede darse el caso que el dispositivo destino no pueda oír la petición o que el canal esté ocupado. Este método suele usarse en sistemas de seguridad.

3.2.6. Campos de aplicación

ZigBee puede aplicarse a una gran cantidad de campos distintos. Algunos de los campos de aplicación son los que indicamos a continuación:

- Automatización comercial de edificios.
- Telecomunicaciones.
- Medición automatizada.
- Automatización del hogar.
- Control industrial.
- Aplicaciones médicas.

3.3. Web Services

3.3.1. Introducción

Los **Servicios Web** o *Web Services*[22] [23], son tecnologías que integran un conjunto de protocolos y estándares para intercambiar datos entre aplicaciones desarrolladas en distintos lenguajes de programación y ejecutadas en cualquier plataforma. Existen gran cantidad de aplicaciones desarrolladas en lenguajes de programación diferentes y ejecutadas sobre cualquier plataforma, que puede utilizar los Servicios Web para intercambiar datos tanto en redes de ordenadores privadas como en Internet.

La interoperabilidad se consigue mediante estándares abiertos[24]. Organizaciones como *OASIS*⁴ y *W3C*⁵ son los comités responsables de la arquitectura y reglamentación de los Servicios Web.

Para mejorar la interoperabilidad entre las distintas implementaciones de Servicios Web existentes, se ha creado el organismo *WS-I*[25], que está encargado del desarrollo de diversos perfiles para definir, de manera más exhaustiva, estos estándares.

3.3.2. Estándares

Existen una serie de estándares para el desarrollo de Servicios Web, que indicamos a continuación:

- **Web Services Protocol Stack.** Es una colección de protocolos y estándares para redes de computadores que son utilizados para definir, localizar, implementar y hacer que un Servicio Web interactue con otro. Comprende cuatro áreas: servicio de transporte, mensajería *XML*, descripción del servicio y descubrimiento de servicios.
- **XML: eXtensible Markup Language.** Nos indica el formato estándar para los datos que se van a intercambiar.
- **SOAP: Simple Object Access Protocol** o **XML-RPC: XML Procedure Call.** Son los protocolos sobre los que se establece el intercambio de datos. También pueden enviarse los datos a través de los protocolos tradicionales como: *HTTP*, *FTP* o *SMTP*.
- **WSDL: Web Services Description Language.** Es el lenguaje de la interfaz pública para los Servicios Web. Consiste en una descripción basada en *XML* de los requisitos funcionales necesarios para establecer una comunicación con los Servicios Web.
- **UDDI: Universal Description, Discovery and Integration.** Protocolo que sirve para publicar la información de los Servicios Web. Permite comprobar qué Servicios Web están disponibles en cada momento.
- **WS-Security: Web Service Security.** Protocolo de seguridad aceptado como estándar por *OASIS*. Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

⁴*OASIS* es el acrónimo de *Advanced Open Standards for the Information Society* y es un consorcio internacional sin ánimo de lucro que orienta el desarrollo, la convergencia y la adopción de los estándares de comercio electrónico y Servicios Web. <http://www.oasis-open.org/>

⁵*World Wide Web Consortium*, *W3C*, es un consorcio internacional que produce recomendaciones para la *World Wide Web*. <http://www.w3.org/>

3.3.3. Ventajas e inconvenientes

Las principales **ventajas** del uso de Servicios Web son las que citamos a continuación:

- Aportan interoperabilidad entre aplicaciones, independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Fomentan el uso de estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y *software* de diferentes compañías ubicadas en distintos lugares geográficos, puedan ser combinados fácilmente para proveer servicios integrados.

Los **inconvenientes** que nos encontramos a la hora de utilizar Servicios Web, son los siguientes:

- Está menos avanzado que otros estándares abiertos de computación distribuida, como es el caso de *CORBA*⁶.
- Bajo rendimiento si se compara con otros modelos de computación distribuida, como *RMI*⁷, *CORBA* o *DCOM*⁸.
- Al apoyarse en el protocolo *HTTP*, se pueden esquivar medidas de seguridad basadas en algún tipo de *firewall*, cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.

3.3.4. Web 2.0

El término **Web 2.0**[26][27] está comúnmente asociado a las aplicaciones Web que facilitan compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la *World Wide Web*. Algunos de los ejemplos de la *Web 2.0* son:

- Comunidades Web.
- Servicios Web.
- Aplicaciones Web.
- Servicios de la red social.
- Servicio de alojamiento de videos.
- Blogs y Wikis.

⁶*Common Object request Broker Architecture*, es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

⁷*Remote Method Invocation* es un mecanismo ofrecido por *Java* para invocar un método de manera remota. Forma parte del entorno estándar de ejecución de *Java* y proporciona un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas en *Java*.

⁸*Distributed Component Object Model* es una tecnología propietaria de *Microsoft* para desarrollar componentes *software* distribuidos sobre varios ordenadores y que se comunican entre sí.

La *Web 2.0* está asociada estrechamente con Tim O'Reilly, fundador y presidente de *O'Reilly Media* y creador de *Global Network Navigator*, primer portal Web de Internet, que posteriormente se convertiría en el portal *America Online*. Aunque el término sugiere una nueva versión de *World Wide Web*, no se refiere a una actualización de las especificaciones técnicas de la Web, sino más bien a cambios acumulativos en la forma en la que los desarrolladores del software y usuarios utilizan la Web.

Antes de la llegada de las tecnologías de la *Web 2.0*, se utilizaban páginas estáticas programadas en *HTML*, que no solían ser actualizadas frecuentemente. El éxito de las páginas Web dependía de hacerlas más dinámicas, dándoles el nombre de *Web 1.5*, donde los sistemas de gestión de contenidos servían páginas *HTML* dinámicas creadas *al vuelo* desde una base de datos actualizada. En ambos sentidos, conseguir visitas y el contenido visual eran considerados como factores importantes.

Los teóricos de la aproximación a la *Web 2.0* creen que el uso de la Web está orientado a la interacción y a las redes sociales, que pueden servir contenido que explota los efectos de dichas redes, creando o no Webs interactivas y más visuales. Es decir, los sitios *Web 2.0* actúan más como puntos de encuentro o Webs dependientes de los usuarios, que como Webs más tradicionales.

Para compartir información en la *Web 2.0* disponemos de una serie de **herramientas**, entre las que podemos citar:

- **Blogs.** La *Blogsfera* es el conjunto de blogs que hay en Internet. Un blog es un espacio Web personal en el que su autor, o autores, pueden escribir cronológicamente artículos, noticias, etc. También es un espacio colaborativo donde los lectores pueden participar escribiendo comentarios a los artículos publicados por el autor.
- **Wikis.** Una wiki es un espacio Web corporativo, organizado mediante una estructura hipertextual de páginas donde varias personas autorizadas elaboran contenidos de manera asíncrona. Basta con pulsar un botón de edición del documento para poder acceder a los contenidos y modificarlos. Suelen mantener un archivo histórico de las versiones anteriores y facilitan la realización de copias de seguridad de los contenidos.
- **Entornos para compartir recursos.** Todos estos entornos nos permiten almacenar recursos en Internet, compartirlos y visualizarlos cuando nos convenga. Constituyen una inmensa fuente de recursos y lugares donde publicar materiales para su difusión mundial.

Algunas de las **tecnologías típicas** que utilizan los sitios *Web 2.0* son las que comentamos a continuación:

- **Ajax.** Es una combinación de *XML* y *Javascript* que posibilita la creación de aplicaciones Web ejecutables en el cliente, reduciendo considerablemente el tráfico de datos y la carga de trabajo del servidor, consiguiendo una mayor interactividad.
- **Mashups.** También conocidos como aplicaciones Web híbridas, integran de manera transparente los datos de otros Servicios Web, gracias a que algunos de éstos han creado *APIs* que permiten desarrollar nuevas aplicaciones que accedan a sus datos gratuitamente.

- **Software social.** Supone un elemento clave en todo el desarrollo de la *Web 2.0*, especialmente en lo que se refiere al aprovechamiento de la inteligencia colectiva. Se entiende como tal al conjunto de aplicaciones que permiten a los individuos comunicarse unos con otros y seguir las conversaciones a través de la Web. También se suelen incluir herramientas y servicios que permiten compartir información y objetos digitales, como vídeos y fotografías.
- **Otras tecnologías utilizadas.** Aquí podemos englobar hojas de estilo *CSS*⁹, *Java Web Start*¹⁰, redifusión/agregación de datos en *RSS*¹¹/*ATOM*¹², *XUL*¹³, *JSON*¹⁴ y *JCC*¹⁵.

⁹ *Cascade Style Sheets (CSS)* es un lenguaje usado para definir la presentación de un documento estructurado escrito en *HTML* ó *XML*.

¹⁰ *Java Web Start* es la implementación de referencia de la especificación *Java Networking Launching Protocol (JNLP)* mediante la cual se permite arrancar aplicaciones *Java* que están en un servidor Web de aplicaciones, comprobando previamente si el cliente tiene la versión actualizada de dicha aplicación y si no es así, se descargará la última versión y se ejecutará en modo local.

¹¹ *Really Simple Syndication (RSS)* es un formato *XML* para syndicar o compartir contenido en la Web, utilizándose para difundir información actualizada frecuentemente a usuarios que se han suscrito a una determinada fuente de contenidos.

¹² *ATOM* es un formato usado para redifusión Web, similar a *RSS*.

¹³ *XML-based User-interface Language (XUL)* es la aplicación de *XML* a la descripción de la interfaz de usuario en el navegador *Mozilla Firefox*. No es una aplicación estándar, pero su principal ventaja es que aporta una definición de interfaces de usuario simple y portable.

¹⁴ *JavaScript Object Notation (JSON)* es un formato ligero para el intercambio de datos, siendo un subconjunto de la notación literal de objetos de *JavaScript* que no requiere el uso de *XML*.

¹⁵ *JavaScript Client Communication (JCC)* hace referencia a las técnicas de programación que, utilizando objetos *JSI* en el navegador en el lado del cliente, facilitan la integración de la misma página Web de aplicaciones y servicios, a priori independientes.

3.4. EPC Information Services (EPCIS)

3.4.1. Introducción

EPC Information Services (*EPCIS*) [28] [29] [30] es un estándar diseñado por *EPCglobal* [31] [32] para permitir la compartición y puesta en común de datos entre distintas empresas. El objetivo de compartir datos es el de permitir a los participantes de la *EPCglobal Network* obtener una visión común de la disposición de los distintos objetos dentro del entorno empresarial.

La versión inicial de *EPCIS* estándar fue aprobada el 12 de Abril de 2007 y proporcionaba los requisitos, que la comunidad *EPCglobal* había identificado como mínimos, de un conjunto de casos de uso analizados. Como tal, el estándar *EPCIS* se puede utilizar en cualquier aplicación y en cualquier tipo de industria. Sin embargo, el estándar admite la extensibilidad para que, los usuarios finales y los grupos industriales, puedan abordar la aplicación y los casos específicos de la industria a través del uso de extensiones personalizadas o especificaciones de la empresa.

El estándar *EPCIS* define interfaces estándar para permitir a los datos relacionados con *EPC* ser capturados y que más tarde serán requeridos para ser usados por un conjunto de operaciones y un modelo de datos asociado. La captura y consulta de los datos relacionados con *EPC* típicamente incluye el uso de bases de datos persistentes, también puede ocurrir sin el uso de bases de datos persistentes a través de una compartición *aplicación-a-aplicación*.

El estándar sólo especifica el interfaz entre aplicaciones que pueden capturar los datos relacionados con *EPC* y lo necesario para poder acceder a ellos. No especifica cómo el servicio de operaciones o las bases de datos tienen que ser implementadas. Las interfaces hacen posible la interoperabilidad mientras que las implementaciones permiten la competición. En la figura 3.4 podemos ver el estándar *EPCIS* y otros estándares de *EPC*.

3.4.2. Especificación del Framework de EPCIS

La especificación del *Framework de EPCIS* nos indica que tiene que tener una estructura en capas, ser extensible y modular.

Estructura de capas

La especificación del *Framework de EPCIS* está organizada en una serie de capas, que indicamos en la figura 3.5 y que describimos a continuación:

- **Abstract Data Model Layer.** Esta capa especifica la estructura genérica de los datos *EPCIS*. Es la única capa que no es extensible por otros mecanismos que no sean por una revisión de *EPCIS*. Especifica los requisitos generales para la creación de definiciones de datos junto con *Data Definition Layer*.
- **Data Definition Layer,** especifica qué dato es intercambiado a través del *EPCIS*, de qué estructura abstracta se trata y su significado. Un módulo de definición de datos es definido junto con su especificación, llamada **Core Event Types Module**. Las definiciones de datos en esta capa son especificadas de forma abstracta, siguiendo las reglas definidas en *Abstract Data Model Layer*.
- **Service Layer.** Esta capa define servicios de interfaz a través de los cuales los clientes *EPCIS* interactúan. En la especificación que estamos tratando, se definen dos módulos de esta *Service Layer*.

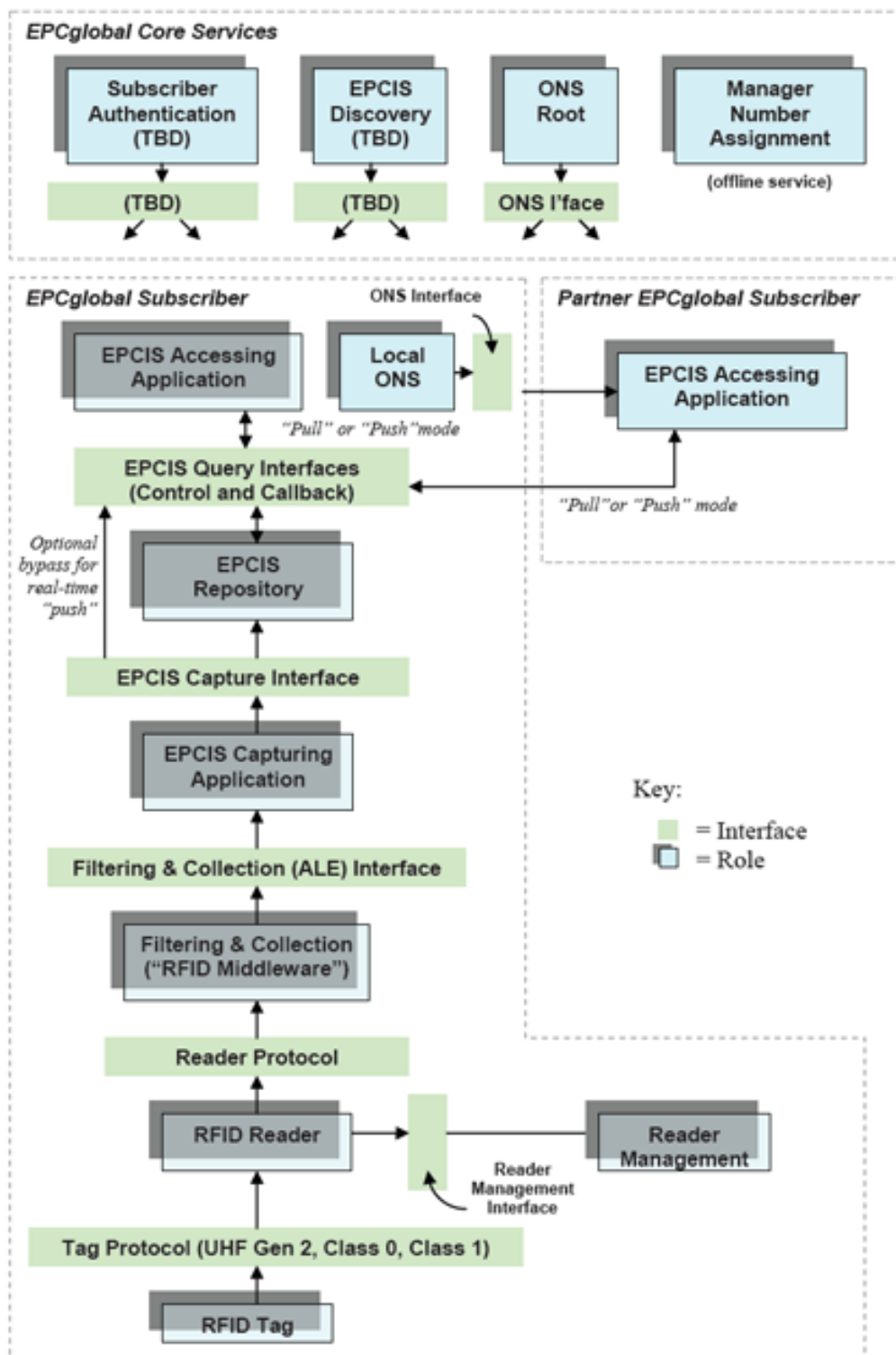


Figura 3.4: Estándar EPCIS y otros estándares EPC

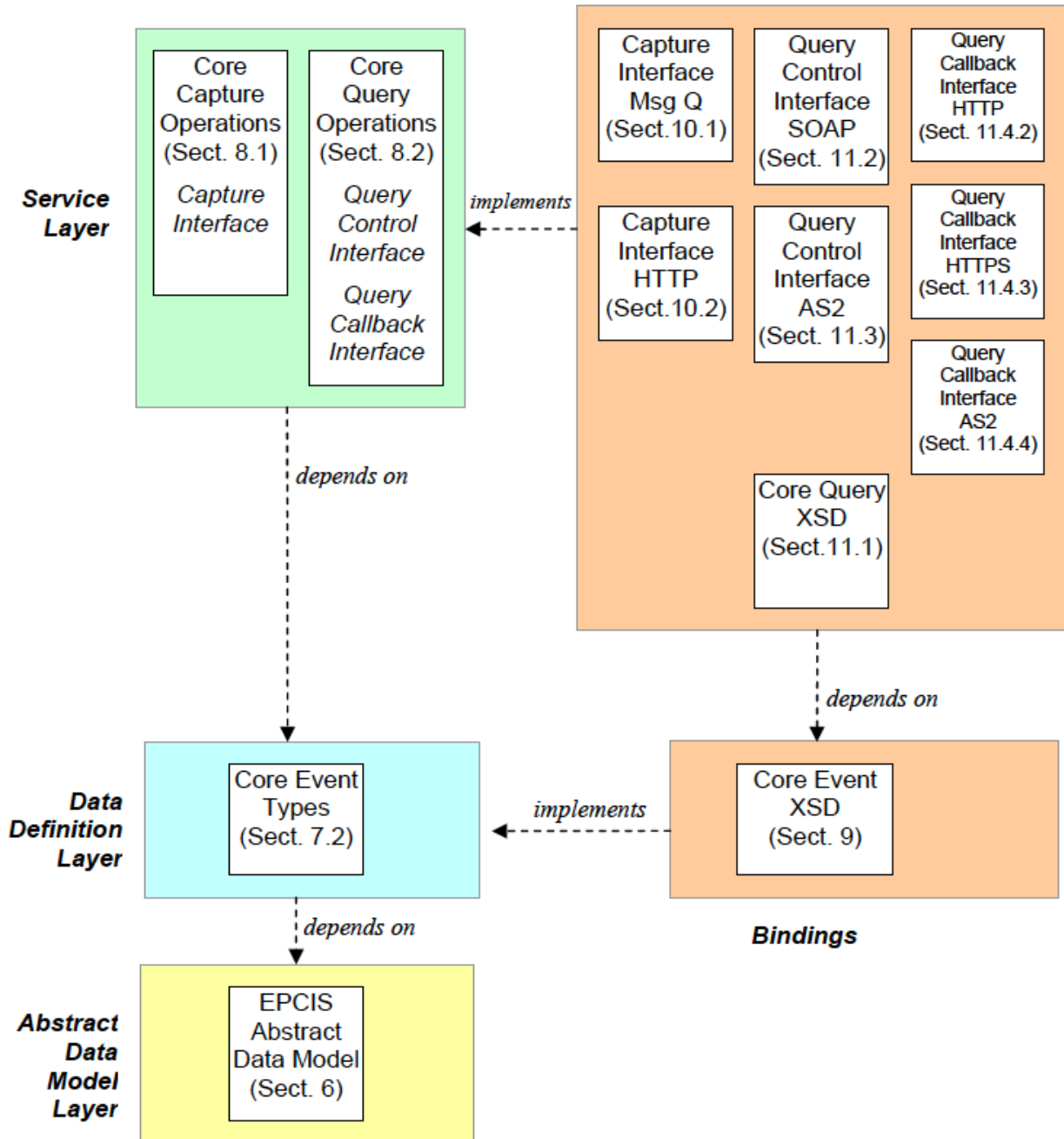


Figura 3.5: Capas EPCIS

El **Core Capture Operations Module** define una interfaz de servicio (*EPCIS Capture interface*) a través de la cual *EPCIS Capturing Applications* utiliza para entregar *Core Event Types* a las distintas partes.

El **Core Query Operations Module** define dos interfaces de servicios (*EPCIS Query Control Interface* y *EPCIS Query Callback Interface*) que *EPCIS Accessing Applications* utilizan para obtener los datos previamente capturados. Las definiciones de interfaz en la *Service Layer* están especificadas, de forma abstracta, usando *UML*.

- **Bindings** (enlaces). Especifica la realización concreta de *Data Definition Layer* y de *Service Layer*. Existen gran cantidad de *bindings* definidos para cualquier módulo *Data Definition* o *Service*. En la especificación que estamos tratando, existen nueve *bindings* especificados para los tres módulos definidos en *Data Definition* y *Service Layer*.

Extensibilidad

La técnica de especificación en capas, promueve la extensibilidad como una capa que puede ser reutilizada por más de una implementación en otra capa. Además de la extensibilidad inherente a esta técnica de especificación capeada, la especificación *EPCIS* incluye varios mecanismos específicos para realizar esa extensibilidad:

- **Subclassing**. Las definiciones de datos en *Data Definition Layer* se definen utilizando *UML*, la cual permite una nueva definición de datos para crear una subclases de una clase ya existente. Una subclase es un nuevo tipo de datos que incluye todos los campos existentes de la clase padre, pudiendo extender esos campos. Una instancia de una subclase puede ser usada en cualquier contexto en el cual podría utilizarse una instancia de su clase padre.
- **Extension Points**. La definición de datos y las especificaciones de servicios pueden incluir *extension points*, los cuales pueden ser usados para extender una determinada funcionalidad sin necesidad de crear subclases.

Modularidad

La especificación del *Framework de EPCIS* indica que ha sido diseñado para ser modular, es decir, no consiste en una especificación sencilla, sino que consiste en una colección de especificaciones individuales que están interrelacionadas. Esto permite a *EPCIS* que crezca y evolucione de forma distribuida. La estructura en capas y los mecanismos de extensibilidad proporcionan los elementos principales para lograr la modularidad, al igual que la agrupación en módulos.

Mientras que las especificaciones son modulares, no es necesario que los límites de las especificaciones del módulo sea visible o explícita dentro de las implementaciones de *EPCIS*.

3.4.3. Tipos de datos

Existen dos tipos de datos en *EPCIS*: **event data** y **master data**.

Los *event data* son creados en el proceso de llevar a cabo los distintos procesos de negocio y son capturados a través del *EPCIS Capture Interface* y están disponibles para búsquedas a través del *EPCIS Query Interface*.

Los *master data* son datos adicionales que proporcionan el contexto necesario para la interpretación de los *event data*. Están disponibles para búsquedas a través de *EPCIS Query Control Interface*.

El estándar define el significado de la estructura de *event data* y *master data* a través de *Abstract Data Model*, *Data Definition Layer* y el módulo *Core Event Types*. El módulo *Core Event Types* especifica los eventos que han sido creados por *EPCIS Capturing Applications* y han sido publicados por una infraestructura de *EPCIS* utilizando el *EPCIS Capture Interface* indicado anteriormente. Estos eventos son devueltos a *EPCIS Accessing Applications* en respuesta a consultas realizadas a través de *EPCIS Query Control Interface* o en respuesta a las consultas realizadas por un *EPCIS Accessing Application*.

Los cinco eventos definidos junto con *Core Event Types* son los que indicamos a continuación:

- *EPCISEvent*. Clase de todos los tipos de eventos disponibles.
- *ObjectEvent*. Evento que ocurre por una o más entidades de *EPCs*.
- *AggregationEvent*. Evento que ocurre por una o varias entidades de *EPCs* que se encuentran físicamente agregadas.
- *QuantityEvent*. Evento que está interesado en una serie de entidades compartidas en una clase de *EPC*, pero donde las identidades individuales de entidades no están especificadas.
- *TransactionEvent*. Evento en el que una o más entidades de *EPCs* se asocian o se separan de una o más transacciones empresariales.

3.4.4. Protocolo de lectura

El protocolo de lectura (*Reader Protocol, RP*) es un interfaz estándar que especifica las interacciones entre un dispositivo capaz de leer o escribir *tags* o etiquetas y una aplicación software. El objetivo es el de definir un interfaz abierto y extensible de un lector que pudiese ser utilizado por vendedores y que pudiese abarcar un número de operaciones estándar, permitiendo la implementación de extensiones a cargo de los proveedores y la inclusión de nuevas operaciones.

La *versión 1.1* se aprobó y se hizo pública en Agosto de 2006. Incluía notables características como instrucciones de lectura, escritura y eliminación de *tags*, accesos a memoria de usuario, configuración extensible, presentación enriquecida de informes, notificaciones asíncronas, múltiples *Message Transport Bindings* incluidos de serie, protocolos de transporte *TCP* y *HTTP*, formatos de mensaje de texto y *XML*, *Read Points* y *Sources* y mecanismos de extensibilidad enriquecidos.

3.5. Internet de las Cosas

3.5.1. Introducción

Internet de las Cosas [33] [34] es un concepto en el que la tecnología del llamado **mundo virtual** de la información se integra con el **mundo real** de las cosas.

El *mundo real* se convierte en más accesible con ordenadores y otros dispositivos conectados en negocios, así como en tareas más cotidianas. Indicar que Internet de las Cosas es algo más que una simple herramienta de negocio que se utiliza para gestionar procesos de forma más eficiente y efectiva que con las herramientas disponibles hasta ahora.

Desde que los fundadores de *Auto-ID Center* [35] crearon el término *Internet of Things*, ha sido utilizado por investigadores y profesionales para describir la combinación del *mundo real* con el *mundo virtual* de información tecnológica, por medio de las tecnologías de identificación automática, sistemas de localización en tiempo real, sensores y actuadores.

Gracias a los avances recientes en miniaturización y los bajos costes de determinadas tecnologías como la Identificación por Radiofrecuencia (*RFID*), Redes de Sensores Inalámbricas (*WSN*), *Near Field Communication* (*NFC*), *WIFI*, etc., Internet de las Cosas se ha convertido en una tecnología relevante para la industria, para los investigadores y para los usuarios finales. Poder detectar el estado de las *Cosas* a través de sensores, junto con la obtención y el procesamiento detallado de los datos obtenidos, nos permite dar una respuesta prácticamente inmediata a los cambios que se produzcan en el *mundo real*.

Con la llegada de tecnologías como *NFC* y *RFID* a los distintos mercados de consumo junto con la posibilidad de tener conexión a *Internet* en los dispositivos móviles (teléfonos, *PDA*, *Tablet Pc*, etc.) y la información escalable compartiendo infraestructuras, se abre un gran espacio para que los desarrolladores de aplicaciones puedan ampliar estas tecnologías.

3.5.2. Definición de Internet de las Cosas

Aunque el término Internet de las Cosas se lleva utilizando muchos años, todavía no existe una definición que se adapte completamente a dicho término, ya que abarca una gran cantidad de conceptos distintos. En las distintas definiciones estudiadas de este término, nos encontramos con el problema de que en todos los casos se dejan conceptos sin incluir a la hora de realizar la definición, por lo que nos encontramos con definiciones que son incompletas.

Podríamos dar una definición minimalista para el concepto Internet de las Cosas. Esta definición tiene que incluir **Cosas**, **Internet** y **conexión entre ambas**.

Las *Cosas* son objetos físicos identificables, independientes de la tecnología que se usa para su identificación y que nos permite saber el estado de los objetos y del entorno que les rodea. *Internet*, en este caso, se refiere a todo lo que va más allá de una red interna o *Extranet*, por lo que requiere el acceso a la información para un gran número de personas o empresas. *Internet* actúa como una infraestructura de almacenamiento y comunicación, que permite una representación virtual de las *Cosas* que unen la información con el objeto real.

3.5.3. *Objetos Inteligentes*

El concepto de **Objeto Inteligente** (*Smart Object*) [36] hace referencia a todos los objetos que son susceptibles de ofrecer una cierta información (por ejemplo, su localización física, origen, estado en el que se encuentran, uso, etc.), llevando este concepto más allá del dispositivo electrónico, como son los electrodomésticos, dispositivos móviles o incluso, aquellos productos de alto desarrollo tecnológico e industrial, como pueden ser los vehículos.

La idea que se persigue con este tipo de objetos, es la de dotar a elementos de uso cotidiano, como ropa, mobiliario, comida o materiales de construcción, de una capacidad de procesamiento de información, mediante transductores embebidos, y/o de indentificación única y comunicación (utilizando tecnologías como *RFID*, *NFC*, Redes de Sensores o actuadores, con el fin de obtener el máximo beneficio de unir el *mundo virtual* de la información con el *mundo real*.

Añadiéndole inteligencia a esos objetos cotidianos, podemos realizar distintas tareas de una forma más sencilla, es decir, al incluir inteligencia a esas *Cosas*, facilitamos su usabilidad, efectividad, eficiencia y conseguimos una mayor satisfacción de uso por parte de los usuarios.

3.5.4. Dominio de aplicación

El término *Cosas* puede ser percibido de forma distinta según el dominio en el que se vaya a utilizar [37]. Tenemos tres dominios de aplicación diferenciados: Industria, Entorno que nos rodea y Sociedad.

En el dominio de la **Industria**, el término *Cosas* se utiliza para referirse al producto, los equipos, medios de transporte, etc. En definitiva, se refiere a todos los elementos que participan en el ciclo de vida de cualquier producto.

En dominio del **Entorno**, el término puede referirse a los árboles, edificios, bancos, papele-ras, coches, etc. Es decir, todos los elementos que nos rodean habitualmente.

Por último, en el dominio de la **Sociedad**, el término *Cosas* se refiere a todos los dispositivos que existen en los espacios públicos, aplicaciones para dispositivos que nos facilitan la vida diaria, etc. En la figura 3.6 se muestra el dominio de aplicación de Internet de las Cosas de una forma gráfica.

Estos tres dominios de aplicación tienen las siguientes características:

- **Industria.** Dentro de este dominio, tenemos actividades que involucran tanto operaciones comerciales como financieras entre compañías, organizaciones de distinta índole y otras entidades. Entre estas actividades podemos citar las siguientes operaciones: de fabricación, logísticas, sector servicios, financieras, con intermediarios, etc.
- **Entorno.** En este dominio existen actividades con respecto a la protección, seguimiento y desarrollo de todos los recursos naturales. Algunas de estas actividades son: reciclaje, agricultura, gestión energética, gestión de servicios del entorno, etc.
- **Sociedad.** Este dominio abarca actividades o iniciativas sobre el desarrollo o la inclusión de sociedades, ciudades y personas. Estas actividades incluyen: servicios del gobierno para los ciudadanos y otras estructuras de la sociedad (*e-commerce*, *e-services*, *e-participación*), *e-inclusión* (personas mayores, personas con algún tipo de minusvalía), etc.

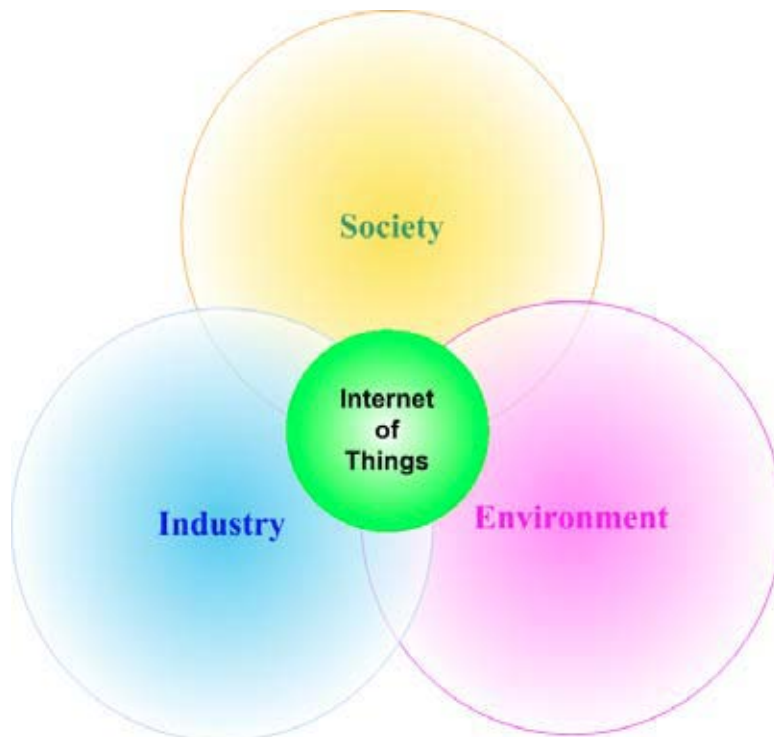


Figura 3.6: Dominio de aplicación de Internet de las Cosas

3.5.5. Campos de aplicación de Internet de las Cosas

El campo de aplicación de Internet de las Cosas es muy extenso y abarca campos tan distintos como la industria aeroespacial, pasando por la industria automovilística, la agricultura, el reciclaje o el entretenimiento.

Algunos de los campos de aplicación son:

- **Industria aeroespacial y aviación.** Sistemas de monitorización y seguridad.
- **Industria automovilística.** Sistemas de monitorización de estado, comunicación vehículo-vehículo y comunicación vehículo-infraestructura.
- **Telecomunicaciones.** Fusión de distintas tecnologías de telecomunicación y creación de nuevos servicios.
- **Edificios inteligentes.** Aplicaciones domóticas, automatización de procesos en el hogar, monitorización *WIFI* y reducción del consumo energético.
- **Tecnología médica y salud.** Junto con tecnologías como *RFID*, *NFC*, *Bluetooth*, *ZigBee*, *6LoWPAN*[38] y *WIFI*, redes de área personal, monitorización de parámetros, posicionamiento y sistemas de localización en tiempo real.
- **Industria farmacéutica.** Productos farmacéuticos y seguridad de dichos productos. Uso de etiquetas inteligentes para monitorizar productos.
- **Logística.** Junto con la tecnología *RFID*, optimización de tareas de chequeo automático de productos, monitorización del *stock* en tiempo real y trazabilidad de productos.
- **Fabricación.** Actividades relacionadas con el ciclo de vida de los productos.

- **Procesos industriales.** Actividades relacionadas con combustibles y gas. Uso de arquitecturas escalables para monitorizar combustibles en situaciones críticas, trazabilidad de estos productos y mantenimiento de equipos necesarios para dicha monitorización.
- **Agricultura y cría de animales.** Trazabilidad de animales y de sus movimientos, localización en tiempo real de animales en el campo, monitorización de animales enfermos para evitar contagios.
- **Reciclaje.** Junto con alguna tecnología inalámbrica, monitorización de las emisiones de los vehículos a la atmósfera para supervisar la calidad del aire, recogida de materiales reciclables y reutilización de elementos electrónicos.
- **Trazabilidad de alimentos.** Monitorización de alimentos, total o parcial, durante su ciclo de vida, para evitar problemas posteriores con productos caducados o en mal estado.
- **Vida independiente.** Bienestar, movilidad, monitorización de personas mayores o con algún tipo de enfermedad.

3.5.6. Concepto Web de las Cosas

El concepto de **Web de las Cosas** (*Web of Things*)[39] [40] es una visión inspirada en Internet de las Cosas en la que los objetos cotidianos están conectados a *Internet*. A diferencia de muchos sistemas que existen para Internet de las Cosas, la Web de las Cosas trata de reutilizar los estándar Web existentes para conectar, de forma rápida, los dispositivos integrados a los *objetos inteligentes* cotidianos. Una vez entendidos y aceptados los distintos estándares, se utilizarán para acceder a la funcionalidad de los *objetos inteligentes*.

Internet de las Cosas se enfoca al establecimiento de conectividad en una gran variedad de entornos de red distintos [41]. El siguiente paso lógico sería el de desarrollar aplicaciones, por lo que es necesario centrarse en la capa de aplicación. En la Web de las Cosas, se consideran los *objetos inteligentes* como ciudadanos de primera clase de la Web.

La Web de las Cosas es un refinamiento de Internet de las Cosas con la integración de *objetos inteligentes*, no sólo en *Internet* o en la red, sino en la Web, más concretamente en su capa de aplicación. Para lograr este objetivo, se propone la reutilización y adaptación de los patrones más comunes que se usan en la Web e introducir una arquitectura en la Web de las Cosas. Se integrarían servidores Web a los *objetos inteligentes* y se aplicaría una arquitectura tipo *REST* (*Representational State Transfer*)¹⁶[42] para el *mundo virtual*.

Algunas de las propiedades de la Web de las Cosas son:

- Uso de *HTTP* como un protocolo de aplicación más que como un protocolo de transporte, como se hace en el mundo de los Servicios Web.
- Indica la funcionalidad síncrona de los *objetos inteligentes* a través del interfaz de *REST*, también conocido como *RESTful API*, y respeta los puntos de las Arquitecturas Orientadas a Recursos.
- Indica la funcionalidad asíncrona de los *objetos inteligentes* a través del uso de distintos estándares Web ampliamente aceptados.

¹⁶La **Transferencia de Estado Representacional** o *REST* es una técnica de arquitectura *software* para sistemas hipermedia distribuidos, como es el caso de la Web.

Estas características nos aseguran el acoplamiento de los servicios con los *objetos inteligentes*, además de ofrecer un interfaz uniforme para acceder y poder construir la funcionalidad de dichos *objetos inteligentes*.

3.6. Localización - Geolocalización

3.6.1. Introducción

Un **Sistema de Información Geográfica** (*SIG* o *GIS* en sus siglas en inglés)[43], es una integración organizada de *hardware*, *software* y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar la información geográficamente referenciada, con el fin de resolver problemas complejos de planificación y gestión.

También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información. En un sentido más genérico, los *SIG* son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

La tecnología *SIG* puede ser utilizada para investigaciones científicas, gestión de recursos, gestión de activos, arqueología, evaluación de impacto ambiental, planificación urbana, cartografía, sociología, logística, *marketing*, etc.

3.6.2. Funcionamiento de un sistema SIG

Un *SIG* funciona como una base de datos con información geográfica que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, señalando un objeto, se conocen sus atributos. Inversamente, preguntando por un registro de la base de datos, se puede saber su localización en la cartografía.

La razón fundamental para utilizar un *SIG* es la gestión de información espacial. El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de forma rápida y sencilla, facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no podríamos obtener de otra forma.

Las principales cuestiones que puede resolver un *SIG* son las siguientes:

- **Localización:** preguntar por las características de un lugar concreto.
- **Condición:** cumplimiento o no de unas condiciones impuestas al sistema.
- **Tendencia:** comparación entre situaciones temporales o espaciales distintas de alguna característica.
- **Rutas:** cálculo de rutas óptimas entre dos o más puntos.
- **Pautas:** detección de pautas espaciales.
- **Modelos:** generación de modelos a partir de fenómenos o actuaciones simuladas.

Indicar que estas cuestiones están ordenadas de menor a mayor complejidad.

Por ser tan versátiles, el campo de aplicación de los *SIG* es muy amplio, pudiendo utilizarse en la mayoría e las actividades como un componente espacial. La profunda revolución que han provocado las nuevas tecnologías ha incidido, de manera decisiva, en su evolución.

3.6.3. Software SIG

Dentro del software disponible para *SIG*, podemos encontrarnos siete grandes tipos de programas informáticos:

- **SIG de escritorio.** Estos programas se suelen utilizar para crear, editar, administrar, analizar y visualizar datos geográficos. También suelen clasificarse según su funcionalidad en: *Visor SIG*, *Editor SIG* y *SIG de análisis*.
- **Sistemas de gestión de bases de datos espaciales o geográficas** (*SGBD espacial*). Se utilizan para almacenar la información geográfica pero, a menudo, también proporcionan la funcionalidad de análisis y manipulación de datos. Una base de datos geográfica o espacial es una base de datos con extensiones que dan soporte de objetos geográficos, permitiendo el almacenamiento, indexación, consulta y manipulación de información geográfica y datos espaciales. El principal beneficio de estas bases de datos se centra en las capacidades que ofrecen en el almacenamiento de datos georreferenciados.
- **Servidores cartográficos.** Se utilizan para distribuir mapas a través de Internet.
- **Servidores SIG.** Proporcionan, básicamente, la misma funcionalidad que los *SIG* de escritorio, pero permiten acceder a estas utilidades de geoprosesamiento a través de una red informática.
- **Clientes Web SIG.** Permiten la visualización de datos y acceder a funcionalidades de análisis y consulta de servidores *SIG* a través de Internet o Intranet. Generalmente, se distingue entre cliente ligero¹⁷ y cliente pesado¹⁸.
- **Bibliotecas y extensiones espaciales.** Proporcionan características adicionales que no forman parte fundamental del programa, ya que pueden no ser requeridas por un usuario medio de este tipo de *software*. Estas nuevas funcionalidades pueden ser herramientas para el análisis espacial, para la lectura de formatos de datos específicos, para la correcta visualización cartográfica de datos geográficos o para la interpretación de especificaciones del *Open Geospatial Consortium*¹⁹.
- **SIG móviles.** Se utilizan para la recogida de datos en campo a través de dispositivos móviles (*PDA*, *Smartphone*, *Tablet Pc*, etc.)

¹⁷Un **cliente ligero** sólo proporciona una funcionalidad de visualización y consulta, como puede ser un navegador Web para visualizar mapas.

¹⁸Un **cliente pesado** suelen proporcionar herramientas adicionales para la edición de datos, análisis y visualización, como son herramientas tipo *Google Earth* o *SIG* de escritorio.

¹⁹*Open Geospatial Consortium (OGC)* es un consorcio de organizaciones públicas y privadas que tiene como objetivo la definición de estándares abiertos e interoperables dentro de los *SIG* y de *World Wide Web*. <http://www.opengeospatial.org/>

3.6.4. Futuro de los SIG

Una gran cantidad de disciplinas se han beneficiado del uso de la tecnología *SIG*. El mercado de *SIG* es muy activo y esto se traduce en una reducción de costes y mejoras continuas en los componentes *hardware* y *software* de los sistemas. Esto provoca que el uso de esta tecnología haya sido asimilado por universidades, gobiernos, empresas e instituciones, que lo han aplicado a distintos sectores.

En la actualidad, los *SIG* están teniendo una fuerte implantación en los llamados **Servicios Basados en Localización** (*LBS*) debido, sobre todo, al abaratamiento y masificación de la tecnología *GPS* integrada en dispositivos móviles de consumo. Los *LBS* permiten a los dispositivos móviles con *GPS* mostrar su ubicación respecto a puntos de interés fijos más cercanos (restaurantes, gasolineras, museos, etc), móviles (amigos, hijos, autobuses, coches, etc.) o para transmitir su posición a un servidor central para su visualización u otro tipo de tratamiento.

Algunas aplicaciones, tanto presentes y futuras de los *SIG*, son las que indicamos a continuación:

- **Cartografía en entornos Web.** Existen gran cantidad de aplicaciones destinadas a mostrar y editar cartografía en entornos Web como *Google Maps*[44], *Bing Maps*[45] u *Open Street Maps*[46] entre otros. Estos sitios Web, dan al público acceso a enormes cantidades de datos geográficos. Algunos de ellos, utilizan *software* que a través de una *API*, permiten a los usuarios crear aplicaciones personalizadas.

Estos servicios ofrecen, por lo general, callejeros, imágenes aéreas o de satélite, geocodificación, búsquedas en nomenclátors²⁰ o funcionalidades de enrutamiento. El desarrollo de internet y las redes de comunicación, también han permitido el surgimiento de distintos estándares de este tipo de tecnología.

- **Tres dimensiones (3D).** Los sistemas existentes en la actualidad, están básicamente sustentados en la gestión y análisis en dos dimensiones (2D), con las limitaciones que esto supone. Existen sistemas híbridos, a medio camino entre las 2D y 3D que poseen capacidades de visualización, denominadas *dos dimensiones y media* (2.5D).

En la actualidad, cada vez se requieren aplicaciones más avanzadas con funcionalidades capaces de gestionar conjuntos de datos complejos, tal y como se perciben en el mundo real por el usuario. Este entorno proporciona un conocimiento mucho mejor de los fenómenos y patrones geoespaciales, ya sea a pequeña o gran escala.

Las dificultades con las que se enfrenta un *SIG* completamente en 3D son grandes y abarcan desde la gestión de geometrías en 3D y su topología, hasta su visualización de una manera sencilla, pasando por el análisis y geoprocesado de la información. Actualmente el *OGC* trabaja en cómo abordar la combinación de los diferentes tipos de modelados resultantes de las tecnologías *GIS*, *CAD* y *BIM* de la forma más íntegra posible.

²⁰Un nomenclátor es una relación de todas las unidades poblacionales inferiores a un municipio (núcleos de población, aldeas, parroquias, caseríos, villas, barrios, etc.).

- **Semántica y SIG.** Las herramientas y tecnologías emergentes desde la *W3C Semantic Web Activity*[\[47\]](#), están resultando útiles para los problemas de integración de datos en los sistemas de información. De igual forma, esas tecnologías se han propuesto como un medio para facilitar la interoperabilidad y la reutilización de datos entre aplicaciones *SIG* y también para permitir nuevos mecanismos de análisis.

Las ontologías son un componente clave en este enfoque semántico, ya que permiten una legibilidad por parte de las máquinas de conceptos y relaciones en un dominio dado. Esto permite al *SIG* centrarse en el significado de los datos en lugar de su sintaxis o estructura.

Capítulo 4

Trabajo de investigación

4.1. Introducción

En este apartado describimos el trabajo que se está desarrollando una vez elegida la línea de investigación a seguir y que, posteriormente, se continuará y completará con el desarrollo de la futura Tesis Doctoral.

4.2. Motivación

La motivación de nuestro trabajo consiste en una propuesta de realización de una arquitectura de Servicios Web, utilizando como base el estándar *EPCIS* de *EPCglobal*, en la que podamos automatizar una serie de Servicios Web, que podrán ser definidos y añadidos posteriormente de forma sencilla a nuestra arquitectura.

A través de una serie de proyectos I+D, llevados a cabo por el Grupo Autolog y que se comentan en el capítulo 7, dentro del estándar *EPCIS*, se ha trabajado con las capas inferiores de este estándar, capas que podemos ver en la figura 4.1.

En estos proyectos se ha trabajado con la identificación y trazabilidad de productos (paneles de hormigón, industriales, farmacéuticos, de consumo alimenticio, etc.) a través del uso de la tecnología *RFID*, tecnología conocida y con la que se está trabajando desde hace tiempo, sobre todo en la parte más *hardware*, con el uso de etiquetas (*tags RFID*) para identificar productos, lectores *RFID* tanto activos como pasivos y el desarrollo de antenas más específicas, con el objetivo de que se adapten a los distintos proyectos.

Dentro de estos proyectos se han desarrollado interfaces gráficos para poder controlar, de forma sencilla e intuitiva, sistemas de gestión de almacenes automatizados, junto con *RFID* para la identificación de productos dentro del almacén.

Dentro del Grupo Autolog también se está trabajando con Redes de Sensores, sobre todo en temas de desarrollo de placas de nodos para la construcción de la Red de Sensores, desarrollo de protocolos de bajo nivel para la comunicación con los dispositivos y en temas relacionados con la eficiencia energética para minimizar el consumo de energía de los dispositivos y obtener una mayor durabilidad de las baterías utilizadas en éstos.

Además, se está se está desarrollando el *RFID Middleware* de elementos *hardware* que, posteriormente, se comunicarán con las capas más altas del estándar.

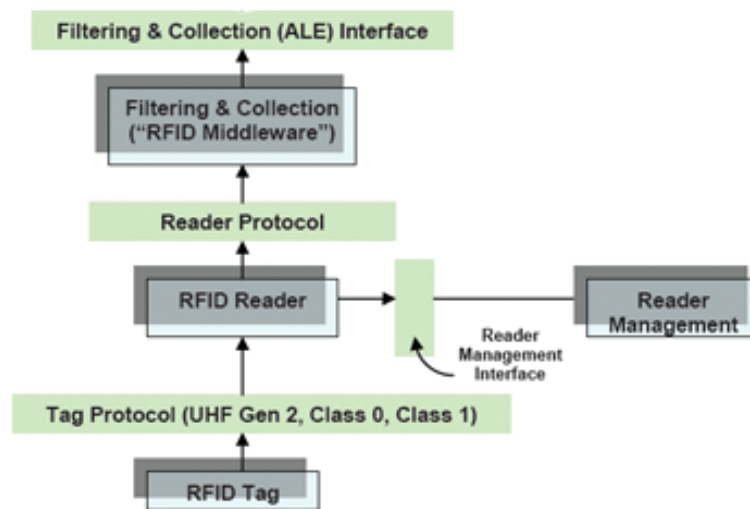


Figura 4.1: Capas inferiores del estándar EPCIS

Después de desarrollar gran parte de nuestro trabajo en las capas inferiores del estándar *EPCIS*, nos hemos planteado centrar nuestro trabajo en las capas superiores, enfatizando en algunas de las tecnologías anteriormente descritas en el capítulo 3, como son Internet de las Cosas y Servicios Web. Por lo tanto, desarrollaremos una serie de aplicaciones, utilizando las capas superiores del estándar, para así poder complementar los trabajos *hardware* que se están realizando.

Una vez definida e implementada la arquitectura de Servicios Web, propondremos la implementación de un primer Servicio Web que consistirá en la realización de una aplicación basada en la localización de personas con algún tipo de enfermedad o minusvalía y con la necesidad de tenerlas siempre localizadas. en entornos abiertos a través de una Red de Sensores Inalámbrica basada en *ZigBee*.

Para poder tener localizadas a estas personas, dispondremos de algún dispositivo (tipo pulsera o caja), colocándose de tal manera que no pueda desprenderse de él fácilmente y así poder realizar el seguimiento por la zona en la que ubiquemos la Red de Sensores Inalámbrica. Esta red podría también utilizarse con animales u objetos en movimiento.

Una vez que se consiga el correcto funcionamiento de este Servicio Web, se procederá a añadir nuevos servicios a nuestro sistema, Servicios Web que serán definidos posteriormente y se propondrán como futuro trabajo asociado a la Tesis Doctoral.

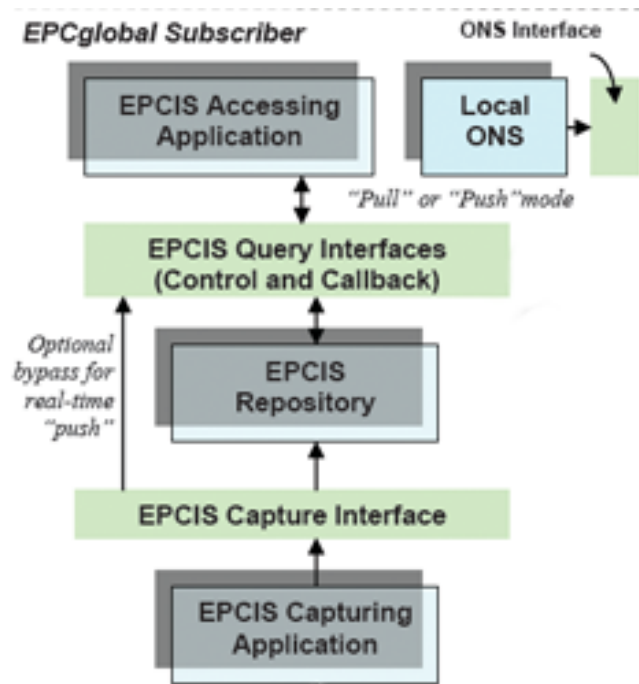


Figura 4.2: Capas intermedias del estándar EPCIS

4.3. Trabajo en desarrollo

El trabajo realizado hasta la fecha, en primer lugar, ha consistido en la selección de un tema novedoso, en el que podamos realizar nuestras investigaciones.

Esta línea, como se ha indicado anteriormente, consistirá en una propuesta de realización de una arquitectura de Servicios Web basada en el estándar *EPCIS* que será ampliamente desarrollado en la futura Tesis Doctoral.

Nuestro trabajo se centra, sobre todo, en las capas más altas del estándar *EPCIS*, sin descuidar las capas más bajas, prestando atención a los dispositivos físicos y a la parte de comunicación entre ellos, aunque no sean la parte central de nuestro trabajo.

En la actualidad, se está trabajando en el estudio del funcionamiento de las capas intermedias del estándar *EPCIS*, capas que mostramos en la figura 4.2, y cómo se comunican entre ellas. Estas capas son: *EPCIS Capturing Application*, *EPCIS Repository* y *EPCIS Accessing Application*.

Para ello, utilizaremos una distribución libre del estándar *EPCIS* llamada *Fosstrak EPCIS Project*, que se explicará detalladamente en el punto 4.3.1 de esta memoria, con la que podremos entender el funcionamiento de estas capas y cómo se comunican entre ellas para, posteriormente, poder realizar nuestras modificaciones para la construcción y desarrollo de la arquitectura de Servicios Web.

Una vez entendidas estas capas intermedias y modificadas para nuestros propósitos, se comenzará con el desarrollo de aplicaciones de nuestra arquitectura.

Una primera aplicación consistirá en el desarrollo de un Servicio Web de localización de personas a través de una Red de Sensores basada en *ZigBee*. Para ello también utilizaremos elementos *hardware* reales, algunos de ellos desarrollados por el Grupo Autolog. Posteriormente, se añadirán otra serie de servicios que se dejarán como trabajo futuro a desarrollar.

Por otro lado, otra parte de nuestro trabajo está consistiendo en el estudio de un estado del arte y la recopilación de información de las distintas tecnologías relacionada con nuestra línea de investigación.

Estas tecnologías, descritas ampliamente en el capítulo 3 de este trabajo, son las siguientes: *ZigBee* (punto 3.2), Redes de Sensores Inalámbricas (punto 3.1), Servicios Web (punto 3.3), *EPCIS* (punto 3.4), Internet de las Cosas (punto 3.5) y Localización-Geolocalización (punto 3.6).

4.3.1. Fosstrak EPCIS Project

Para poder realizar nuestra arquitectura de Servicios Web, se está utilizando **Fosstrak EPCIS Project** (*Free and Open Source Software for Track and Trace*) [48], que es una implementación en código abierto de *EPCIS* [49] que nos proporciona tanto el *EPCIS Repository* certificado por *EPCglobal*, así como el cliente *Capture* y el cliente *Query*, programados en *Java*.

Además de estos elementos estándar, también existe un *Webadapter* con el que conseguiremos un acceso sencillo a *EPCIS* a través de protocolos Web.

Algunas de las características que *Fosstrak EPCIS Project* nos permite, son las que citamos a continuación:

- Desarrollar de forma sencilla un *EPCIS Repository*.
- Realizar consultas (*Querys*) sobre un *EPCIS Repository* existente a través del interfaz gráfico que posee la herramienta.
- Añadir *EPC data* a un *EPCIS Repository* existente a través del interfaz gráfico que posee la herramienta.

Organización

Fosstrak EPCIS Project, como se muestra en la figura 4.3, está formada por tres módulos y en ella se muestran los escenarios de interacción para cada uno de estos tres módulos:

- Una implementación del repositorio *EPCIS Repository*.
- Una aplicación interactiva de *EPCIS Capture*.
- Una aplicación interactiva de *EPCIS Query*.

La arquitectura que utiliza, es del tipo **cliente-servidor**¹. Un **cliente** puede ser o una aplicación *EPCIS Capture*, una aplicación *EPCIS Query* o ambas a la vez. Un **servidor** es un *EPCIS Repository*, que proporciona las interfaces con las cuales los clientes pueden conectarse.

¹La arquitectura **cliente-servidor** consiste, básicamente, en un cliente que realiza peticiones a otro programa (servidor) que le da respuesta.

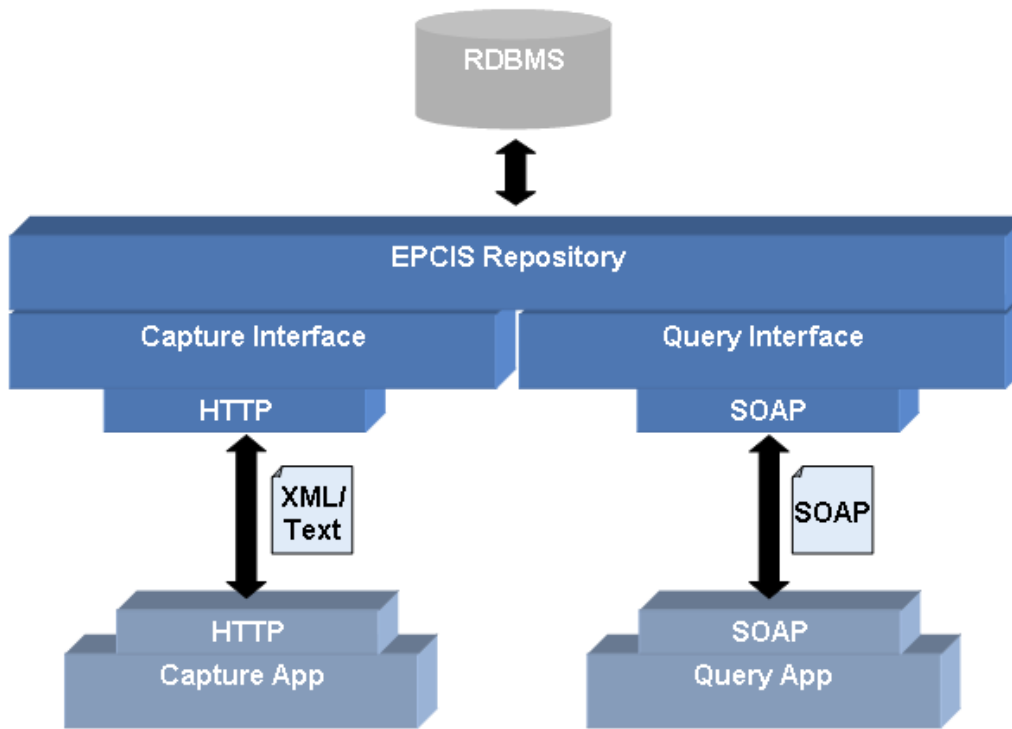


Figura 4.3: Implementación Fosstrak EPCIS

El repositorio analiza las peticiones de los clientes y las procesa, de acuerdo a las reglas definidas en la especificación. Los protocolos de transporte utilizados por las aplicaciones cliente son: *XML* sobre *HTTP*, para *EPCIS Capture*, y *SOAP* sobre *HTTP*, para *EPCIS Query*.

EPCIS Repository

Fosstrak EPCIS Repository implementa los siguientes enlaces, definidos por el estándar *EPCIS*:

- Enlaces *XML* para la capa de definición de datos (*data definition layer*).
- Enlaces *HTTP* para *Capture Interface*.
- Enlaces *SOAP/HTTP* para *Query Control Interface*.
- Enlaces *HTTP* para *Query Callback Interface*.
- Enlaces *HTTPS* para *Query Callback Interface*.

A continuación, vamos a indicar cómo tendríamos que implementar cada uno de los enlaces que hemos indicado.

- **Nivel de acceso.** El nivel de acceso es el punto de entrada del *EPCIS Repository* y nos proporciona los interfaces que utilizarán las aplicaciones cliente para poder acceder a dicho repositorio.

Capture Interface. El enlace *HTTP* para el *Capture Interface* está implementado por un *Java Servlet* ² el cual es registrado por el contenedor de *servlets* (en nuestro caso, utilizaremos *Apache Tomcat* [50] como contenedor de *servlets*).

²Los *servlets* son objetos que se ejecutan en un servidor o contenedor *Java Enterprise Edition (JEE)*, especialmente diseñado para ofrecer contenido dinámico desde un servidor Web, generalmente *HTTP*.

El *servlet* recibe solicitudes de *Capture* desde una aplicación *EPCIS Capture*. Estas solicitudes deben incluir eventos *EPCIS* en serie, en un documento *XML* y se debe dar en la carga de una solicitud *HTTP POST*. El *servlet* valida el documento *XML* entrante contra el esquema *EPCIS* y lo valida en el *CaptureOperationsModule* en el **Nivel de negocio**.

Query Control Interface. Con el fin de implementar los enlaces *SOAP/HTTP* para el *Query Control Interface*, utilizaremos *Apache CXF Web Service Framework*³[51] en combinación con *JAX-WS API* [52] y *JAXB data binding* [53] para mapear objetos *Java* a *XML* y viceversa.

CXF mapea el contenido de las solicitudes *SOAP* entrantes a objetos *Java* y los entrega al *QueryOperationsModule* en el **Nivel de negocio** para su procesamiento.

- **Nivel de negocio.** Este nivel, recibe entradas del **Nivel de acceso**. Está formado por dos módulos: *CaptureOperationsModule* y *QueryOperationsModule*.

Capture Operations Module. Este módulo lo que hace es coger eventos *EPCIS* y guarda su contenido en el *RDBMS* (Sistema de Gestión de Bases de Datos Relacionales). Utilizaremos *Hibernate* [54] para mapear los objetos del modelo de dominio a la base de datos relacional y generar las llamadas *SQL* requeridas.

Query Operations Module. Este módulo genera consultas *SQL* dinámicamente y las envía al **Nivel de recursos** para su procesamiento. El conjunto de resultados devueltos por el **Nivel de recursos** son mapeados al modelo de dominio de eventos de *EPC* y son entregados al **Nivel de acceso**, el cual devuelve los resultados a la aplicación cliente.

Query Callback Interface. Este interfaz no es accesible directamente para las aplicaciones cliente, sino que la aplicación cliente *suscribe* una consulta vía *Query Callback Interface*, con lo que el repositorio devuelve el resultado de la consulta a través de dicho interfaz.

El componente que implementa este interfaz es *QuerySubscription*, que representa consultas que ya están lanzadas o están programadas para ser lanzadas. Una *QuerySubscriptionScheduled* representa una suscripción programada y una consulta. Por otro lado, una *QuerySubscriptionTriggered* representa a una suscripción asociada a un *URI* lanzado y a una consulta.

- **Nivel de recursos.** Este nivel está formado por un *RDBMS*, en el cual se almacenan todos los datos de **eventos**, **vocabularios** y **suscripciones**.

En la figura 4.4, podemos ver un ejemplo del esquema del evento *ObjectEvents*. Indicar también que los esquemas de los eventos *Transaction-*, *Quantity-* y *AggregationEvents* son similares a *ObjectEvents*.

³*Apache CXF* es un *framework* de servicios en código abierto. *CXF* facilita el desarrollo y la construcción de servicios utilizando un *frontend* de programación de *APIs*. Estos servicios pueden utilizar una gran variedad de protocolos como *SOAP*, *XML/HTTP*, *RESTful HTTP* o *CORBA*, pudiendo trabajar con gran variedad de protocolos de transporte como *HTTP*, *JMS* o *JBI*.

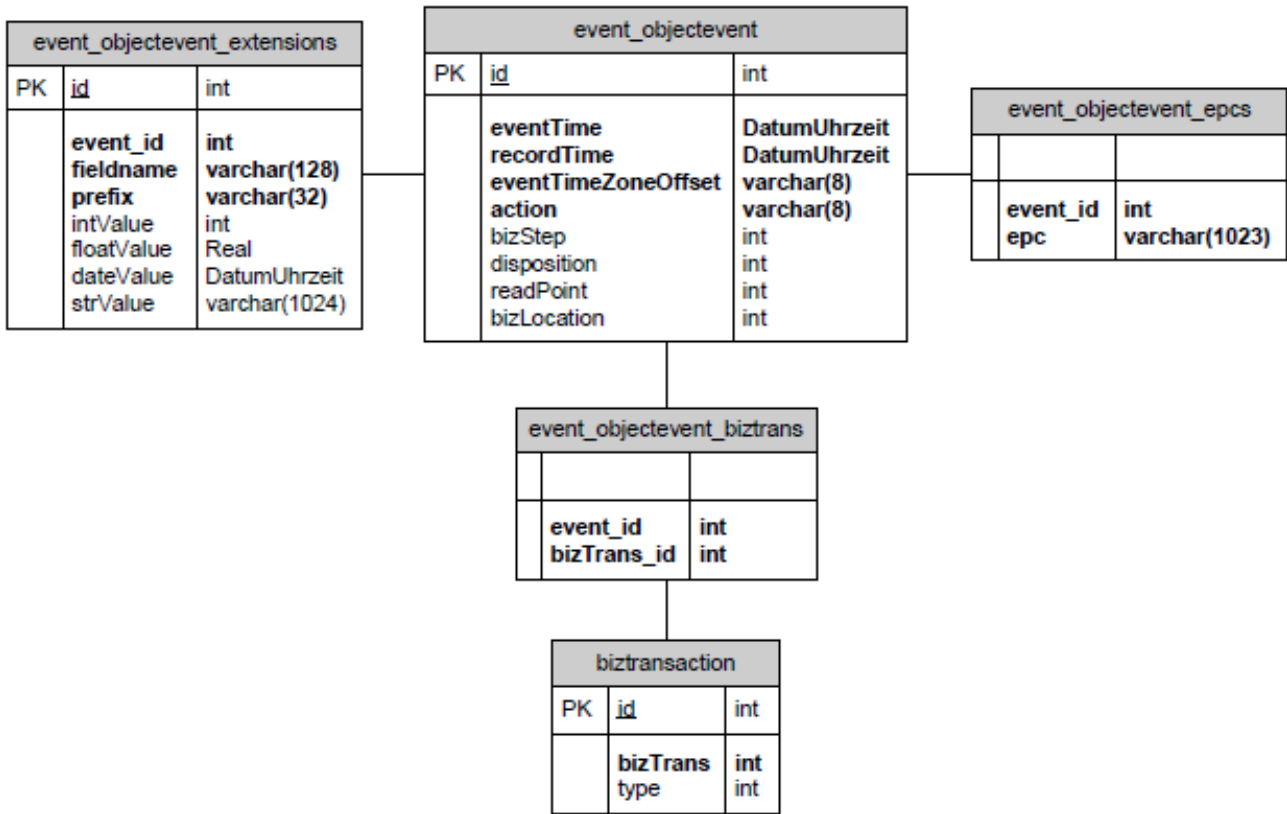


Figura 4.4: Esquema del evento ObjectEvents

Cada **vocabulario estándar** que poseamos, tiene que tener su propia tabla en la base de datos. En la figura 4.5, mostramos los esquemas para los vocabularios estándar *Business*, *Transaction*, *Disposition* y *ReadPoint*. Los **vocabularios no estándar** son almacenados en la tabla de la base de datos *voc_any*.

También, como indicamos anteriormente, las suscripciones también se almacenan en el *RDBMS*. En la figura 4.6 podemos ver el esquema correspondiente a las suscripciones.

EPCIS Capture Application

Una aplicación *Capture EPCIS* se utiliza para capturar eventos en el repositorio. Debido a la implementación de los enlaces *HTTP/POST*, una aplicación de captura debe enviar los eventos *XML* en la carga de una solicitud *HTTP/POST* al repositorio. Utilizaremos los enlaces de datos *JAXB* para mapear el modelo de domino de eventos *EPCIS* a *XML* y viceversa.

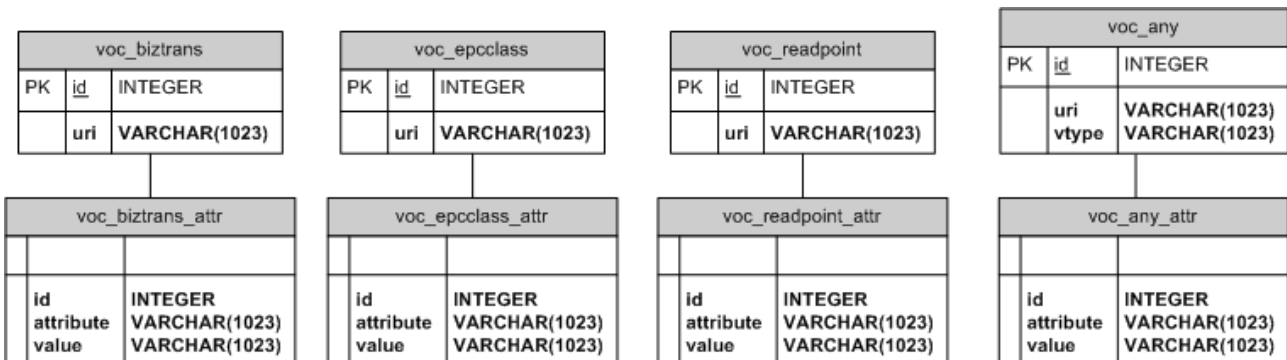


Figura 4.5: Esquemas de los vocabularios Business, Transaction, Diposition y ReadPoint

subscription		
PK	<u>subscriptionid</u>	varchar(767)
	params	Bild
	dest	varchar(1023)
	sched	Bild
	trigg	varchar(1023)
	initialrecordingtime	DatumUhrzeit
	exportifempty	smallint
	queryname	varchar(1023)
	lastexecuted	DatumUhrzeit

Figura 4.6: Esquema de las suscripciones

La aplicación de captura *Fosstrak EPCIS* proporciona tanto la *API* de desarrollo, como el interfaz de usuario. La *API* puede ser utilizada por otras aplicaciones para enviar las consultas (*Capture requests*) al repositorio y la interfaz puede utilizarse, de forma manual, para enviar eventos *EPCIS* dentro del repositorio.

EPCIS Query Application

Una aplicación *Querys EPCIS*, envía las consultas al *EPCIS Repository* utilizando solicitudes *SOAP*. La aplicación de consultas también utiliza el *framework* del *CXF Web Service* para generar las solicitudes de *SOAP* y utiliza los enlaces de datos de *JAXB* para mapear los objetos *Java* a *XML*.

Similar a la aplicación *Capture*, la aplicación de *Query* proporciona tanto la *API* de desarrollo, como el interfaz de usuario.

4.3.2. Elementos hardware

Para comenzar con el desarrollo de la Red de Sensores Inalámbrica basada en *ZigBee*, el Grupo Autolog comenzó utilizando una serie de elementos *hardware* comerciales de la empresa **Libelium** [55] para realizar las primeras pruebas.

Como nodos, se han utilizado las placas *WaspMote* de esta empresa. Estas placas están basadas en una arquitectura modular, con la idea de integrar únicamente los módulos que se necesiten en cada dispositivo, pudiendo cambiarlos y ampliarlos según las necesidades. Poseen puertos de entrada/salida para poder comunicarse con otros dispositivos externos (sensores, componentes o módulos electrónicos) siempre que sean compatibles entre sí.

A estas placas se les puede añadir diferentes tipos de sensores, según los parámetros que queramos controlar. Existen sensores de gases, eventos, para agricultura, detección de zonas para aparcar, acústicos, personalizables, etc.

También hay disponibles una gran cantidad de módulos para añadirle a las placas: *ZigBee*, *Bluetooth*, *GSM/GPRS*, *GPS* y *WaspMote Gateway*.

Una vez realizadas las primeras pruebas con dichos módulos comerciales, se detectaron una serie de problemas que necesitaban la modificación de las placas. Como esto no era posible, se llegó a la conclusión de que era necesario el desarrollo de unas placas propias, para poder solucionar esos problemas que surgieron y reducir el consumo, cosa que con las placas comerciales no podíamos obtener.

A fecha de entrega de este trabajo, el desarrollo de las placas del Grupo Autolog está finalizado y también la fase de pruebas. Se está comenzando con la fase de desarrollo del *middleware* para completar su funcionamiento.

Capítulo 5

Conclusiones

El objetivo principal de este trabajo es el de marcar las bases para la futura Tesis Doctoral y la búsqueda de una línea de investigación a seguir.

Se ha seleccionado un tema novedoso, para poder realizar nuestras investigaciones. Esta línea consistirá en una propuesta de realización de una arquitectura de Servicios Web basada en el estándar *EPCIS* y que será ampliamente desarrollado en la futura Tesis Doctoral.

Se ha realizado un acercamiento y un estudio de las tecnologías relacionadas con nuestra línea de investigación, sobre todo en las tecnologías que no nos eran conocidas a priori, realizando un capítulo de este trabajo dedicado al estado del arte de todas las tecnologías relacionadas con nuestras investigaciones.

Para completar nuestro trabajo con las capas inferiores de *EPCIS*, se ha comenzado a trabajar con las capas superiores, con el objetivo de poder desarrollar aplicaciones con dicho estándar.

También se están desarrollando placas para nodos *ZibBee* de construcción propia, una vez que se han utilizado y evaluado soluciones existentes en el mercado, y que no se adaptaban totalmente a nuestros requerimientos, sobre todo en temas de eficiencia energética.

Finalizado el desarrollo de estas placas y terminada también la fase de pruebas, nos encontramos en la fase de desarrollo del *middleware* para completar su funcionamiento.

En la fecha de entrega de este trabajo, se ha procedido a la instalación de *Fosstrak EPCIS Project* y se ha comenzado a realizar las primeras pruebas y modificaciones de código para poder continuar con nuestras investigaciones, entender completamente el programa y así poder obtener los objetivos propuestos.

Capítulo 6

Futuras líneas y trabajo futuro

Como trabajo futuro y nuevas líneas de investigación, se proponen los siguientes puntos:

- Desarrollo completo de la arquitectura de Servicios Web basada en el estándar *EPCIS*, utilizando *Fosstrak EPCIS Project*.
- Implementación del Servicio Web de localización de personas propuesto, utilizando los elementos *hardware* que hemos indicado en el punto [4.3.2](#).
- Desarrollo de un protocolo de comunicación para volcar los datos que nos envíen los nodos *ZigBee* a un PC o un Servidor Web.
- Desarrollo de una aplicación para posicionar exactamente a las personas dentro de nuestra Red de Sensores Inalámbricos. Para ello se ha pensado en realizar triangulación con *ZigBee*.
- Añadir nuevos Servicios Web a nuestra arquitectura:
 - Monitorización de constantes vitales.
 - Instalación de un acelerómetro en el dispositivo que lleve la persona para poder controlar si sufre desmayos, caídas, etc.
 - Controlar a las personas para que no se salgan del recinto de la Red de Sensores.
 - Podómetro.
 - Desarrollo de un sitio Web para poder saber dónde se encuentra la persona en todo momento. Posibilidad de consulta de este sitio también por familiares.
- Desarrollo de una aplicación para realizar Minería de Datos con los datos que obtengamos de los distintos Servicios Web y poder obtener conclusiones.

Capítulo 7

Currículum Vitae

7.1. Datos personales

- **Apellidos y nombre:** San José Vieco, José Iván.
- **Fecha y lugar de nacimiento:** 25 de Febrero de 1979, Cuenca.
- **Categoría actual:** Contratado con cargo a proyecto I+D en el Instituto de Tecnologías Audiovisuales de la Universidad de Castilla La Mancha (Cuenca).
- **Contacto:** JoseIvan.SanJose@uclm.es

7.2. Formación Académica

- **Graduado en Ingeniería Informática** por la UCLM (2011).
- **Ingeniero Informático** por la UCLM (2010).
- **Título de Máster en Edificación y Hogar Digital (III Edición)** por la UCLM (2010).
- **Ingeniero Técnico en Informática de Gestión** por la UCLM (2002).

7.3. Participación en proyectos I+D

- **Título del proyecto:** Red de Turismo Inteligente (RedTur).
Duración: Julio 2010 - Julio 2012.
Investigador principal: Miguel Ángel López Guerrero.
- **Título del proyecto:** Nuevos materiales para la construcción de fachadas de GRC fotocatalíticas, autolimpiables y de máxima sostenibilidad ambiental.
Duración: Julio 2010 - Diciembre 2012.
Investigador principal: José Manuel Pastor García.
- **Título del proyecto:** Aplicación de Arquitecturas de Sistemas Multiagente complementados con RFID a Entornos Industriales Dinámicos No Deterministas.
Duración: Abril 2009 - Marzo 2012.
Investigador principal: José Manuel Pastor García.

- **Título del proyecto:** Entorno interactivo para el control y simulación de Almacenes Automáticos. (2008-2009).
Duración: Marzo 2007 - Diciembre 2009.
Investigador principal: José Manuel Pastor García.
- **Título del proyecto:** Sistema de Trazabilidad Combinado del ciclo de vida de los paneles de hormigón arquitectónicos. (2008).
Duración: Marzo 2007 - Diciembre 2008.
Investigador principal: José Manuel Pastor García.

7.4. Asociaciones Científico-Técnicas

- Miembro del **Grupo Autolog** [56] de la Universidad de Castilla La Mancha, grupo para la automatización de la identificación y el seguimiento de productos en actividades de fabricación y logística.
- Miembro de la **Sociedad Española de Trazabilidad (SETra)** [57]. También realizo funciones de diseño y mantenimiento de su página Web.

7.5. Líneas de Investigación

Palabras clave: *RFID* (Identificación por Radiofrecuencia), Sistemas Multiagente, Trazabilidad, Redes de Sensores, *ZigBee*, Servicios Web (*Web Services*), *EPCglobal*, Internet de las Cosas.

7.6. Publicaciones

Pastor, J.M., García, A. y San José, J.I.: “El Futuro de la Identificación de Objetos: RFID”. Revista Universitaria Bartolomé Cossío nº13 [58], edición digital, p. 4-11. Mayo de 2011. D.L. CU - 136 - 1996.

7.7. Idiomas

Nivel B1 de Inglés. Certificado de Aptitud “Aprende inglés en tu campus” por la Universidad de Castilla La Mancha.

7.8. Otros méritos

- VI Encuentro de Intercambio de Experiencias de Innovación Docente: Implantación de los Nuevos Grados y sus Implicaciones.
- VI Encuentro de Intercambio de Experiencias de Innovación Docente: Evaluación de Competencias en los Nuevos Grados.
- Asistencia a la I Jornada de Acogida para Profesorado Universitario de Nueva Incorporación celebrada en el Campus de Cuenca.
- Participación en Evaluaciones del 4º Concurso Universitario de Software Libre de Castilla La Mancha.

- Colaborador de la X Semana de la Ciencia en Castilla La Mancha.
- Mantenimiento y actualización de la Página Web de la Escuela Politécnica de Cuenca, desarrollando un simulador de convalidaciones de asignaturas del plan de estudios de Arquitectura Técnica al Grado en Ingeniería de Edificación para su uso en la Web de la Escuela.
- Participación en Evaluaciones del 3º Concurso Universitario de Software Libre de Castilla La Mancha.
- Participación y asistencia en el Día de las Telecomunicaciones de CAstilla La Mancha.
- Participación en Evaluaciones del 2º Concurso Universitario de Software Libre de Castilla La Mancha.
- Colaborador en el programa de actividades de la Residencia Universitaria Bartolomé Cossío de Cuenca relacionadas con el XX Aniversario de la misma.
- Participación en la IX Edición de la Semana de la Ciencia de Castilla La Mancha.
- Miembro del Comité Organizador de las Segundas Jornadas Científicas sobre RFID desarrolladas en Cuenca en Noviembre de 2008.

Bibliografía

- [1] Google scholar. [Online]. Available: <http://scholar.google.es/> 2.1.1
- [2] Scopus. [Online]. Available: <http://www.scopus.com/> 2.1.1
- [3] Isi wos. [Online]. Available: <http://www.accesowok.fecyt.es/login/> 2.1.1
- [4] Lenguaje r. [Online]. Available: <http://lib.stat.cmu.edu/R/CRAN/> 2.1.1
- [5] Latex. [Online]. Available: <http://www.latex-project.org/> 2.1.1
- [6] Beamer. [Online]. Available: <http://www.antoniomtz.org/manuales/presentacion-latex-beamer.pdf> 2.1.1
- [7] Unified parallel c (upc). [Online]. Available: <http://upc.lbl.gov> 2.1.2
- [8] Wikipedia. Ansi c. [Online]. Available: http://es.wikipedia.org/wiki/ANSI_C 2.1.2
- [9] Lenguaje c. [Online]. Available: <http://es.wikipedia.org/C> 2.1.2
- [10] Grupo louise. [Online]. Available: <http://www.i3a.uclm.es/louise/louise/> 2.1.3
- [11] E. Foundation. (2011) Eclipse ide. [Online]. Available: <http://www.eclipse.org> 2.2.3
- [12] ——. (2011) Eclipse modeling framework project. [Online]. Available: <http://www.eclipse.org/modeling/emf/> 2.2.3
- [13] D. Zone. (2009) ¿qué es una red de sensores inalámbrica (wsn)? [Online]. Available: <http://zone.ni.com/devzone/cda/tut/p/id/9507> 3.1.1
- [14] A. Pérez, G. Núñez, A. Rodríguez, R. Salazar, and I. Zilio. (2010) Topologías de red wsn. [Online]. Available: <http://wirelessnetworkproyecto.blogspot.com/2010/07/topologias-de-red-wsn.html> 3.1.3
- [15] F. Beltrán, “Wireless sensor networks,” *Bit*, no. 165, pp. 61–64, 2007. 3.1.4
- [16] Wikipedia. Red de sensores. [Online]. Available: http://es.wikipedia.org/wiki/Red_de_sensores 3.1.4
- [17] J. Valverde. (2007) El estándar inalámbrico zibee. [Online]. Available: <http://seccperu.org/files/ZigBee.pdf> 3.2.1
- [18] Wikipedia. Zigbee. [Online]. Available: <http://es.wikipedia.org/wiki/ZigBee> 3.2.1
- [19] Z. Alliance. Zigbee alliance. [Online]. Available: <http://www.zigbee.org/> 3.2.1
- [20] Wikipedia. Ieee 802.15.4. [Online]. Available: http://es.wikipedia.org/wiki/IEEE_802.15.4 3.2.1

- [21] Zigbee.es. [Online]. Available: <http://www.zigbee.es> 3.2.5
- [22] M. Papazoglou, *Web Services: Principles and Technology*, P. Hall, Ed. Addison-Wesley, 2008. 3.3.1
- [23] O. Cabrera, M. Oriol, and L. López. Wessqos: Un sistema soa para la selección de servicios web según su calidad. [Online]. Available: <http://www.lsi.upc.edu/~llopez/publications/09JSWEB.pdf> 3.3.1
- [24] Wikipedia. Servicio web. [Online]. Available: http://es.wikipedia.org/wiki/Servicio_web 3.3.1
- [25] WS-I. Web services interoperability organization (ws-i). [Online]. Available: <http://www.ws-i.org/> 3.3.1
- [26] D. Margaix. (2007) Conceptos de web 2.0 y biblioteca 2.0: origen, definiciones y retos para las bibliotecas actuales. [Online]. Available: <http://eprints.rclis.org/bitstream/10760/9521/1/kx5j65q110j51203.pdf> 3.3.4
- [27] Wikipedia. Web 2.0. [Online]. Available: http://es.wikipedia.org/wiki/Web_2.0 3.3.4
- [28] EPCglobal. Epc information services (epcis) version 1.0.1 specification. [Online]. Available: http://www.gs1.org/gsm/kc/epcglobal/epcis/epcis_1.0.1-standard-20070921.pdf 3.4.1
- [29] ——. (2011) Epcis. [Online]. Available: <http://www.gs1.org/gsm/kc/epcglobal/epcis> 3.4.1
- [30] Wikipedia. (2011) Epcis. [Online]. Available: <http://en.wikipedia.org/wiki/EPCglobal> 3.4.1
- [31] EPCGlobal. Epcglobal. [Online]. Available: <http://www.gs1.org/epcglobal> 3.4.1
- [32] ——. (2011) Epcglobal usa. [Online]. Available: <http://www.epcglobalus.org> 3.4.1
- [33] Wikipedia. (2011) Internet de las cosas. [Online]. Available: http://es.wikipedia.org/wiki/Internet_de_las_Cosas 3.5.1
- [34] D. Uckelmann, M. Harrison, and F. Michahelles, “An architectural approach towards the future internet of things,” *Springer*, 2011. 3.5.1
- [35] A.-I. Labs. Auto-id center. [Online]. Available: <http://www.autoidlabs.org/> 3.5.1
- [36] D. Casado, M. Vega, and M. López, “Infraestructuras inteligentes en el internet del futuro,” 2010. 3.5.3
- [37] A. de Saint-Exupery, *Internet Of Things: Strategic, Research, Roadmap*. CERP-IoT, September 2009. 3.5.4
- [38] Wikipedia. 6lowpan. [Online]. Available: <http://en.wikipedia.org/wiki/6LoWPAN> 3.5.5
- [39] D. Guinard and V. Trifa. (2011) Web of things. [Online]. Available: <http://www.webofthings.com> 3.5.6
- [40] Wikipedia. (2011) Web of things. [Online]. Available: http://en.wikipedia.org/wiki/Web_of_Things 3.5.6

- [41] D. Guinard, V. Trifa, and E. Wilde, “A resource oriented architecture for the web of things,” 2010. 3.5.6
- [42] Wikipedia. (2011) Representational state transfer. [Online]. Available: http://es.wikipedia.org/wiki/Representational_State_Transfer 3.5.6
- [43] ——. (2011) Sistema de información geográfica. [Online]. Available: http://es.wikipedia.org/wiki/Sistema_de_Informacion_Geografica 3.6.1
- [44] Google. Google maps. [Online]. Available: <http://maps.google.es> 3.6.4
- [45] Microsoft. Bing maps. [Online]. Available: <http://www.bing.com/maps/> 3.6.4
- [46] A. O. España. Open street maps. [Online]. Available: <http://www.openstreetmap.es/> 3.6.4
- [47] W. Staff. W3c semantic web activity. [Online]. Available: <http://www.w3.org/2001/sw/> 3.6.4
- [48] C. Floerkeeier, C. Roduner, and M. Lampe. (2011) Fosstrak. [Online]. Available: <http://www.fosstrak.org/> 4.3.1
- [49] Fosstrak. (2011) Fosstrak download. [Online]. Available: <http://www.fosstrak.org/epcis/download.html> 4.3.1
- [50] A. S. Foundation. (2011) Apache tomcat. [Online]. Available: <http://tomcat.apache.org/> 4.3.1
- [51] ——. (2011) Apache cxf. [Online]. Available: <http://cxf.apache.org/> 4.3.1
- [52] J. Proyect. (2011) Jax-ws. [Online]. Available: <http://jax-ws.java.net/> 4.3.1
- [53] ——. (2011) Project jaxb. [Online]. Available: <http://jaxb.java.net/> 4.3.1
- [54] RedHat. Hibernate. [Online]. Available: <http://www.hibernate.org/> 4.3.1
- [55] L. Comunicaciones. (2011) Libelium: Wireless innovation. [Online]. Available: <http://www.libelium.com/> 4.3.2
- [56] Grupo autolog. [Online]. Available: <http://autolog.uclm.es> 7.4
- [57] Sociedad española de trazabiliad (setra). [Online]. Available: <http://www.rfidspain.org> 7.4
- [58] J. Pastor, A. García, and J. San José. El futuro de la identificación de objetos: Rfid. [Online]. Available: [http://edu.jccm.es/ru/rubcossio/paginas/publicaciones/revistas\(rar\)/revista13.rar](http://edu.jccm.es/ru/rubcossio/paginas/publicaciones/revistas(rar)/revista13.rar) 7.6
- [59] N. Samama, *Global Positioning. Technologies and Performance*. Wiley-Interscience, 2008.