



Sistemas Informáticos

Curso 2005-06

Diseño e Implementación del Proceso de
Validación de Datos de Red Eléctrica
Española con Certificación de Resultados

Javier Lucas González
Enrique Martorano Rincón
Javier Soriano Díaz

Dirigido por:
Prof. Luis Javier García Villalba
Dpto. Sistemas Informáticos y Programación
Grupo de Análisis, Seguridad y Sistemas (GASS)

Facultad de Informática
Universidad Complutense de Madrid



Los abajo firmantes: Javier Lucas González, Enrique Martorano Rincón, y Javier Soriano Díaz, autorizan a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales, y, mencionando expresamente a sus autores, tanto la presente memoria, como el código, la documentación, y/o el prototipo desarrollado.

Javier Lucas González

Enrique Martorano Rincón

Javier Soriano Díaz





ÍNDICE

Prefacio	7
Resumen.....	8
English Summary	9

INTRODUCCIÓN..... 10

<i>El sector eléctrico en España y sus Servicios.....</i>	10
<i>La percepción del cliente</i>	10
<i>Los servicios como capacidad distintiva</i>	11
<i>El nuevo servicio</i>	11
<i>Posibilidades del Nuevo servicio.....</i>	11
<i>Motivación del proyecto</i>	11
<i>Duración</i>	12
<i>Terminología.....</i>	12

SECCIÓN I: LECTURA DE CONTADORES. 15

Parte I: fundamentos técnicos.....	15
1- Interfaz de comunicaciones RS-232.....	16
2.- Comandos Hayes (Comandos AT)	16
Parte II: Análisis y diseño.....	17
1- Protocolo de comunicaciones entre registradores y concentradores de medidas o terminales portátiles de lectura	17
1.1- Nivel Físico.....	17
1.2- Nivel de enlace.....	17
1.2.1- Formato de las tramas.....	18
1.3- Nivel de aplicación.....	22
1.3.1- Unidad Identificadora de Datos	22
1.3.2- Objetos de Información.....	27
1.3.3- Ejemplos de ASDUs.....	27
2- Sistema de automatización de lecturas remotas.....	28
2.1- Lecturas remotas.....	28
2.2- Lecturas concurrentes.....	29
Parte III: Especificación y requisitos.	29
1- Protocolo de comunicaciones entre registradores y concentradores de medidas o terminales portátiles de lectura	29
1.1- Especificación del protocolo.....	29
2- Sistema de automatización de lecturas remotas.....	29
2.1- Especificación del sistema.....	29
3- Requisitos de la Sección II.....	30
Parte IV: Implementación.....	30
1.- El lenguaje y el entorno	30
2.- Descripción general.....	31
3.- Protocolo de comunicaciones entre registradores y concentradores de medidas o terminales portátiles de lectura	31
3.1. Diagrama de secuencia para el inicio de sesión.....	31
4.- Sistema de automatización de lecturas remotas.....	33



SECCIÓN II: CPCSCC 35

Parte I: fundamentos técnicos.....	35
1.- XML	35
2.- Servicios Web.....	36
Arquitectura de los servicios web.....	37
UDDI, el descubrimiento del servicio.....	38
WSDL, la descripción del servicio web.....	38
SOAP.....	39
HTTP, la capa de transporte.....	39
3- DIME	40
4.- WS-Attachments	40
Parte II. Análisis y Diseño	40
1.- CPCSCC	40
2.- Diagrama de Frontera del protocolo.....	41
Diagramas UML.....	42
Diagrama de clases.....	42
Diagramas de Secuencia.....	44
Confirmar fichero.....	44
Enviar Fichero.....	44
Información disponible	45
Petición Fecha hora.....	45
Recoger fichero	45
Parte III: especificación y requisitos	46
1.- Especificación concreta de los servicios Web a implementar.....	46
2.- Requisitos del proyecto	46
Parte IV: Implementación.....	47
1.- El lenguaje y el entorno	47
2.- Tipos construidos	48
3.- Descripción de los Servicios Web	48
Petición de Fecha y Hora	48
Información Disponible.....	50
Recoger Fichero.....	53
Confirmación de Recepción de Fichero	55
Enviar Fichero	58
4.- Detalles de Implementación.....	60
Gestor de Archivos	60
Gestor de certificados	61
Gestor de Logs.....	61
Web Services	61
5.- Descripción de los servicios Web. Archivo cpcsc.wSDL.....	62
CONCLUSIONES.....	66
BIBLIOGRAFÍA.....	67

ANEXO A. RIESGOS DEL PROYECTO..... 68

Riesgos de Software	68
Tipos de riesgos	68
Análisis de riesgos	69
Estrategias de gestión de riesgos.....	70

ANEXO B. PALABRAS CLAVE. 71



ÍNDICE DE FIGURAS

<i>Figura 1 - Contadores eléctricos.....</i>	<i>13</i>
<i>Figura 2 - Esquema global del proyecto.....</i>	<i>14</i>
<i>Figura 3 - Formato de una trama de longitud fija.....</i>	<i>18</i>
<i>Figura 4 - Formato de una trama de longitud variable.....</i>	<i>19</i>
<i>Figura 5 - Campo de control. Comunicación servidor-> contador.....</i>	<i>20</i>
<i>Figura 6 - Campo de control. Comunicación contador->servidor.....</i>	<i>21</i>
<i>Figura 7 - Formato de los la Unidad Identificadora de Datos.....</i>	<i>22</i>
<i>Figura 8 - Formato de la causa de transmisión.....</i>	<i>24</i>
<i>Figura 9 - ASDU Tipo 141.....</i>	<i>27</i>
<i>Figura 10 - ASDU Tipo 142.....</i>	<i>28</i>
<i>Figura 11 - Diagrama de secuencia para el inicio de sesión.....</i>	<i>32</i>
<i>Figura 12- Diagrama de secuencia para una transmisión.....</i>	<i>33</i>
<i>Figura 13 - Interactuación entre los roles de un servicio web.....</i>	<i>37</i>
<i>Figura 14 - La pila del protocolo Web-Service.....</i>	<i>38</i>
<i>Figura 15 - Diagrama de frontera del protocolo CPCSCC.....</i>	<i>41</i>
<i>Figura 16 - Diagrama de clases del protocolo CPCSCC.....</i>	<i>43</i>
<i>Figura 17 - Diagrama de secuencia para Confirmar Fichero.....</i>	<i>44</i>
<i>Figura 18 - Diagrama de secuencia para Enviar Fichero.....</i>	<i>44</i>
<i>Figura 19 - Diagrama de secuencia para Información Disponible.....</i>	<i>45</i>
<i>Figura 20 - Diagrama de secuencia para Petición Fecha y Hora.....</i>	<i>45</i>
<i>Figura 21 - Diagrama de secuencia para Recoger Fichero.....</i>	<i>46</i>

ÍNDICE DE TABLAS

<i>Tabla 1 - Tipos de ASDU.....</i>	<i>23</i>
<i>Tabla 2 - Causas de transmisión.....</i>	<i>25</i>
<i>Tabla 3 - Direcciones de registro.....</i>	<i>26</i>
<i>Tabla 4 - Tipos de fichero y descripción.....</i>	<i>50</i>
<i>Tabla 5 - Riesgos Software.....</i>	<i>68</i>
<i>Tabla 6 - Tipos de Riesgos.....</i>	<i>68</i>
<i>Tabla 7 - Análisis de riesgos.....</i>	<i>69</i>
<i>Tabla 8 - Estrategias de Gestión de Riesgos.....</i>	<i>70</i>



Prefacio

Este documento es la memoria del Proyecto de Sistemas Informáticos para el curso 2005-2006 “Proceso de Validación de Datos de Red Eléctrica Española con Certificación de Resultados.” dirigido por el Profesor D. Luis Javier García Villalba y desarrollado por los alumnos:

- Javier Lucas González
- Enrique Martorano Rincón
- Javier Soriano Díaz

A lo largo de estas páginas, se explicará el proyecto en sí, características técnicas, motivaciones, metodología empleada, etc.



Resumen

Este proyecto ha hecho posible lectura de datos de consumo eléctrico e incidencias en la red y permite la transmisión de estos datos, desde los contadores eléctricos hasta los equipos de almacenamiento de datos.

Para alcanzar los objetivos propuestos, hemos tenido que implementar dos protocolos, uno de extracción de información y otro de comunicación. La primera sección de este documento define el 'Protocolo de comunicaciones entre registradores y concentradores de medidas o terminales portátiles de lectura', un protocolo de bajo nivel que toma la información de los registradores de medida eléctricos a través de puertos serie o por una conexión por módem. La segunda, define el 'Protocolo de Comunicaciones entre Concentradores de Medida. Solución basada en "Web Services"', que permite una transmisión segura de estos datos entre concentradores secundarios o entre un concentrador secundario y otro primario.



English Summary

This project makes possible reading of electrical consumption data and allows transmission of these information from electrical meters to data storage systems.

In order to reach suggested goals, we have implemented two protocols, one for data extraction and another one for communications. First section of this document defines 'Communication protocol between measurement recorders and concentrators or portable reading terminals', a low level protocol which gets information from electrical measurement recorders through serial ports or a modem connection. The second one, defines 'Meter Concentrator communications protocol. "Web Services" based solution', which makes possible a secure transmission of these data between secondary concentrators or a secondary and a primary concentrator.



Introducción

El sector eléctrico en España y sus Servicios

El 1 de Enero de 2006 entró en vigor el Decreto-Ley que terminaba de liberalizar el mercado eléctrico español. La principal novedad de la ley desde el punto de vista del consumidor, es que otorgaba plena capacidad de decisión a este sobre que compañía de electricidad contratar, fuera cual fuese su situación geográfica. Convertía al sistema de cableado en un ente único, al que las empresas y particulares se conectaban, y al que las compañías Eléctricas vendían electricidad.

Se produce por tanto un proceso de liberalización, como el ocurrido con otros sectores estratégicos a finales del siglo pasado, como el petróleo o el transporte aéreo. No es el objetivo de este documento discutir factores macroeconómicos, pero sí cabe destacar una diferencia importante con respecto a las otras liberalizaciones: *la percepción del cliente*

La percepción del cliente

Tras comprar un producto, un cliente pretende de manera inconsciente examinarlo y comprobar sus características físicas. Cuanto más complejo es el producto, más difícil es establecer ese contacto entre el producto y el cliente. Así por ejemplo, en los productos de tecnología, el comprador solo aprecia una carcasa de plástico coloreada que tiene muy poco que ver con el contenido útil y real del producto en sí. A mayor *distancia física* entre producto y cliente, menos se sentirá identificado éste con el primero, y más difícil será la comercialización. **La identificación del producto con la compañía es un objetivo fundamental de ésta.**

En el caso de las compañías eléctricas, lo anterior supone un inconveniente extremo. Podemos enumerar algunas razones

1. el cliente no puede apreciar directamente (ver ni tocar), el producto que adquiere
2. No establece una relación comprador-vendedor pues no tiene que desplazarse para adquirir el producto. Directamente lo obtiene de su punto de acceso particular
3. el producto como tal, es absolutamente indistinguible del que obtendría de haberlo comprado en otra compañía

Debido a que el producto en sí no influye en la percepción del cliente, las compañías eléctricas deben ofrecer algo más. Dar valor añadido a sus productos: **Ofrecer servicios.**



Los servicios como capacidad distintiva

Ante productos intangibles, la necesidad de diferenciar el producto es suplida por las compañías con la oferta de servicios a los clientes. En las Eléctricas, dichos servicios han sido tradicionalmente *Atención al cliente*, y *mantenimiento*. Con el presente proyecto, es posible ofertar un nuevo servicio, permitiendo a quien lo implante diferenciar su producto y ganar cuota de mercado.

El nuevo servicio

Los protocolos del proyecto permite ofrecer información personalizada, segura, fiable e instantánea sobre consumos. Puede ser usada por el cliente para conocer los detalles pormenorizados de su consumo, o por la propia compañía para mejorar su servicio. Esta información es

- Segura: toda la comunicación de datos se realiza bajo un protocolo seguro, SSL. La información está de esta manera protegida contra terceras personas “a la escucha” de los datos. Además, tiene integrado un sistema de **firma digital**, con el que la identificación del suministrador de datos y el receptor, queda asegurada.
- Fiable: el protocolo provee de sistemas de control de error para descartar falsos envíos de datos o lecturas erróneas.
- Instantánea: si la comunicación se realiza a través de Internet, es posible tomar decisiones rápidas en función del análisis de los datos.

Posibilidades del Nuevo servicio

Como servicio diferenciador, es de esperar que tenga una buena acogida entre las compañías eléctricas, o entre compañías de servicios eléctricos. Con la liberalización antes comentada, el mercado debería de acoger bien estas iniciativas. Creemos que el potencial del servicio y su escalabilidad para conseguir nueva funcionalidad lo hacen bastante atractivo.

Motivación del proyecto

A lo largo de los cinco años de carrera, creemos que no habíamos visto con suficiente profundidad el nivel de capa de enlace. En ninguna asignatura se han realizado practicas sobre ello, y teníamos un interés particular en ese nivel de abstracción.

Por otro lado, tampoco habíamos estudiado tecnologías actuales como Web services, soap, xml, y otros sistemas de comunicación basados en



arquitecturas actuales. Al elegir este proyecto, pretendimos formarnos en esos áreas, a la vez que explorábamos un sector nuevo en expansión, el eléctrico.

Partíamos además con la ayuda en el verano de 2005 con la ayuda de dos profesores visitantes de la Universidad de Brasilia, Fabio Mesquita Buiati y Robson da Oliveira. Ambos ofrecieron conocimientos y experiencia en el tema durante los primeros momentos del diseño y desarrollo del proyecto.

Queríamos además, realizar un proyecto con posibilidades reales de salir al mercado, y no desarrollar una aplicación meramente teórica o de ámbito estrictamente universitario. Tener tanto la posibilidad de continuar, (en el sentido de ampliar) el proyecto tras terminarlo en la facultad, como la alternativa de entrar en plantilla de una empresa del sector, fueron dos motivaciones clave para elegir este proyecto

Pero por encima de los motivos anteriores, ha estado la voluntad de investigar un nuevo protocolo e innovar. Llevar a la práctica un proyecto que previamente solo estaba esbozado en un marco teórico ha sido un reto y una satisfacción.

Duración

La extracción de información de los contadores eléctricos se prolongó desde octubre de 2005 hasta marzo de 2006. A finales de ese mes, cuando ya contábamos con información útil a enviar, implementamos el protocolo de comunicación entre contador y concentrador. Las últimas pruebas satisfactorias del sistema fueron el treinta de junio de 2006, lo que da al proyecto una duración aproximada de nueve meses

Terminología

Antes de continuar debemos de aclarar en este punto qué significan algunos conceptos reiteradamente usados en el documento: concentradores secundarios y principales, puntos de acceso, redes de acceso, registradores de medidas, etc. Junto a cada definición aportamos cuando es posible una fotografía o esquema que facilita la comprensión.

Concentradores principales

Un concentrador principal, o CP es una entidad lógica que esconde un complejo sistema software detrás. Cada compañía eléctrica tiene un solo CP, que recoge información de todos los concentradores secundarios asociados que puedan tener sus clientes. El CP actúa como centralizador último de información, y juega un rol principal en la gestión de datos de consumos. Sus datos son importantes en la toma de decisiones sobre qué medidas tomar en la ampliación o reducción del suministro en un determinado momento.

Red de acceso



Se trata de una infraestructura de comunicaciones que incluye desde el módem del registrador hasta la entrada al servidor de comunicaciones del concentrador secundario al que se conecta, y las comunicaciones entre concentradores secundarios.

Concentradores secundarios

Un concentrador secundario, o CS es una entidad lógica a menor nivel que los CP. Si el número de aquellos era muy limitado (solo uno por compañía eléctrica), el número de CS es mucho mayor. Cada cliente industrial puede dar de alta (registrar) un concentrador secundario particular con cualquier Eléctrica. Los concentradores secundarios recogen información de los registradores de medida y se comunican con el CP, bien de forma manual mediante un operario, bien mediante Internet. También pueden comunicarse con otros concentradores secundarios.

Por tanto, un CS es un sistema software que, desde el punto de vista técnico, actúa como interfaz entre las tramas de byte que ofrecen los registradores de medida y el nivel de aplicación que es necesario para establecer comunicación con el CP

Registrador de medidas

Un registrador de medidas es un componente hardware suministrado por fabricantes especializados en electricidad, tales como Orbis o Siemens. Situado en un punto de acceso, es capaz de registrar cada vatio que circula desde la Red eléctrica hasta el usuario. Junto a la información eléctrica, almacena meta datos como fechas, picos de tensión, etc.



Figura 1- Contadores eléctricos



Punto de acceso

La situación física donde se sitúa un registrador de medidas se denomina punto de acceso. Por tanto, dentro de una empresa pueden existir numerosos registradores de medidas y puntos de acceso, un solo concentrador secundario.

En la siguiente figura puede apreciarse la estructura de los diferentes objetos descritos anteriormente

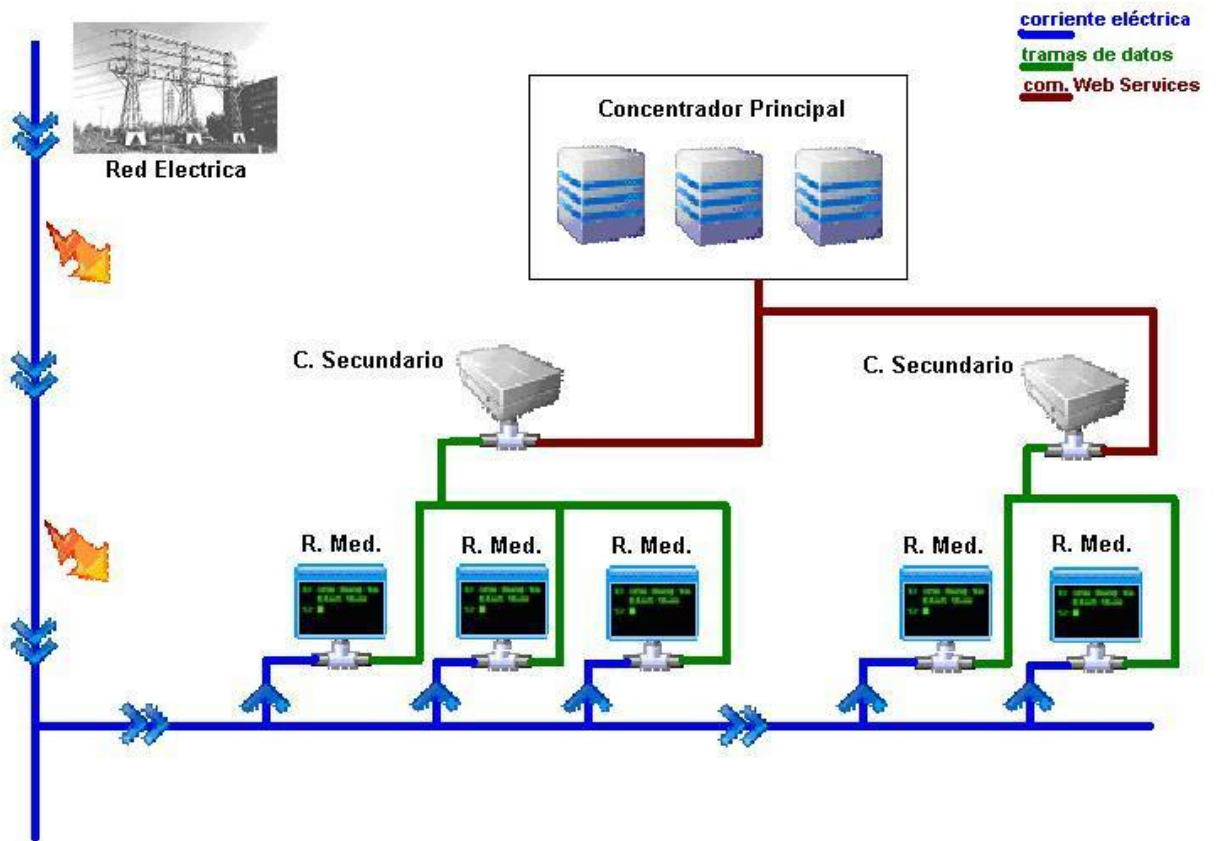


Figura 2 - Esquema global del proyecto



Sección I: Lectura de Contadores

Esta parte de nuestro proyecto se encargó de la implementación del 'Protocolo de comunicaciones entre registradores y concentradores de medidas o terminales portátiles de lectura' de Red Eléctrica de España, disponible en:

http://www.ree.es/cap03/pdf/simel/protoc_RMCM10042002.pdf

Este protocolo sigue la norma IEC 870-5-102 de la International Electrotechnical Commission, que presenta un estándar internacional para la transmisión de totales integrados en sistemas de energía eléctrica, que a su vez hace referencia a los siguientes estándares de la misma comisión:

- Norwegian IEC 870-5-101: personalización para el mercado noruego del estándar de comunicación IEC 870-5-101
- IEC 870-5-1: que describe los formatos de las tramas transmitidas en la capa de enlace.
- IEC 870-5-2: que describe los procedimientos de transmisión en la capa de enlace.
- IEC 870-5-3: que describe las unidades de datos de servicio de aplicación (Application Service Data Units, ASDU).
- IEC 870-5-4: que describe los elementos de información para la capa de aplicación.
- IEC 870-5-5: que describe las funciones posibles a realizar sobre este protocolo.

Por otra parte, dentro de esta sección se implementó una aplicación para la lectura automatizada de contadores eléctricos remotos. Estos terminales poseen un módem GSM integrado y las lecturas se efectúan realizando una llamada a dichos módems y, a continuación, estableciendo una comunicación con el protocolo para la lectura de contadores.

Tanto para la lectura de contadores locales como remotos la conexión se establecía por el puerto serie, directamente conectado a un contador, en el primer caso, o conectado a un módem GSM en el segundo.

Parte I: fundamentos técnicos.

Para la realización de esta parte del proyecto utilizamos distintas tecnologías y herramientas, utilizando Java como lenguaje de programación en el que implementamos el conjunto.

Las tecnologías que trataremos en este apartado son las siguientes:

- Interfaz de comunicaciones RS-232, para la comunicación entre el servidor y los terminales externos, ya fueran contadores o módems GSM.



- El conjunto de comandos Hayes para la configuración de módems (en nuestro caso módems GSM), el establecimiento de llamadas de datos y su gestión y finalización.

En cuanto a herramientas utilizadas podemos destacar la librería javax.comm para el manejo de puertos serie y paralelo siguiendo las especificaciones del interfaz RS-232.

1- Interfaz de comunicaciones RS-232.

El interfaz RS-232 define una norma para el intercambio de datos binarios entre dos equipos, un Equipo terminal de datos (DTE) y un Equipo de terminación del circuito de datos (DCE). La comunicación se efectúa por ráfagas de bits de manera tanto asíncrona como síncrona y con distintas codificaciones, que pueden diferir en número de bits, bits de paridad (paridad par, impar o ninguna), velocidad de transmisión (entre 19200 bps y 600 bps) y existencia o no de bit de stop.

En nuestro proyecto, se utilizó este estándar para la comunicación entre el servidor y los contadores eléctricos, en caso de de obtención directa de las lecturas y para la comunicación entre el servidor y los módems GSM, en caso de obtención remota de las lecturas.

2.- Comandos Hayes (Comandos AT)

El conjunto de comando Hayes (también conocido como conjunto de comandos AT), consiste en una serie de instrucciones para la configuración de módems y el establecimiento y gestión de llamadas. Este conjunto de instrucciones permite operar al módem en dos modos:

- Modo de comandos: en este modo los datos que recibe el módem son interpretado como directivas a ejecutar en el propio módem, para la configuración del mismo, establecimiento de llamadas, etc.

- Modo de datos: en este modo los datos recibidos son enviados a un módem remoto, con el que previamente se iniciado una llamada.

Este conjunto de comandos es muy amplio, y para la realización de nuestro proyecto utilizamos sólo un pequeño subconjunto. Pasamos a detallar los comandos utilizados:

- +++ : este comando es utilizado para alternar entre modo comando y modo de datos.
- AT&V: comprueba la configuración actual del módem.
- ATZ: resetea el módem.
- ATSr=n: cambia el valor de configuración del registro r, utilizando como nuevo valor n.



- AT&F0: cambia la configuración del módem a la configuración de fábrica por defecto.
- ATDTn: este comando inicia una llamada por tonos al número n. En caso de poder iniciar la llamada, el módem que inicio la misma recibirá como respuesta
- ATH: Finaliza una llamada.

Parte II: Análisis y diseño.

1- Protocolo de comunicaciones entre registradores y concentradores de medidas o terminales portátiles de lectura

El protocolo sigue la norma IEC 870-5-102. Así pues la estructura del protocolo sigue los niveles especificados en dicho documento:

- Nivel físico: Basado en recomendaciones ITU-T.
- Nivel de enlace: procedimientos de transmisión de enlace y tramas que soportan los mensajes de aplicación.
- Nivel de aplicación: funciones de aplicación que implican la transmisión de ASDUs (Unidades de Datos de Servicios de Aplicación) entre origen y destino.

1.1- Nivel Físico.

La comunicación usa como canal la comunicación serie según las recomendaciones ITU-T. Más concretamente, para modo remoto se soportarán las siguientes normas:

- ITU-T V.24/V.28 ([3]-5.1.1) para el interfaz de datos entre los ETD y los ETCD, con velocidades de hasta 38.400 bit/s.
- ITU-T V.32, V.32 bis y V.34 para intercambio de datos entre los ETCD, con velocidades de transmisión de hasta 28.800 bit/s.

En modo local se admite cualquier otro medio de conexión, siempre que se atenga a estándares internacionales y siendo las velocidades empleadas una de las siguientes: 200 bit/s 300 bit/s 600 bit/s 1200 bit/s 2400 bit/s 4800 bit/s 9600 bit/s 19200 bit/s 38400 bit/s 57600 bit/s 115200 bit/s.

1.2- Nivel de enlace.

El nivel de enlace sigue los estándares especificados en los documentos IEC 870-5-1 e IEC 870-5-2.



Las consideraciones más importantes a tener en cuenta son las siguientes:

1.2.1- Formato de las tramas.

El protocolo define dos tipos de tramas, tramas de longitud fija y tramas de longitud variable. Pasamos a definir los formatos de ambas para luego especificar cuáles son los datos agrupados en cada campo de ellas. Para ellos utilizaremos indistintamente codificación binaria o codificación hexadecimal (en este último caso denotada como xxH).

Longitud fija:

Se trata de tramas de 6 bytes, con un primer byte de START que delimita el inicio de la trama, un byte que contiene la información de control, dos bytes que codifican la dirección de enlace del terminal, un byte de Checksum y un byte de finalización de trama.

Tramas de longitud fija

START (10H)
C
A
A
CHECKSUM
END (16H)

Figura 3 - Formato de una trama de longitud fija

Longitud variable:

Se trata de tramas de longitud mayor de 9 bytes, con un primer byte de START que delimita el inicio de la trama, dos bytes que contienen información sobre la longitud de la zona de Datos de usuario (parte de la trama de longitud variable), otro byte de START, un byte que contiene la información de control, dos bytes que codifican la dirección de enlace del terminal, a continuación un conjunto de byte de longitud variable que contiene la información de usuario, una trama de Checksum y un byte de finalización de



trama. En la zona de Datos de usuario irá incluida la información del ASDU (Unidad de Datos de Servicios de Aplicación), es decir la información correspondiente al nivel de aplicación, con un formato variable que presentaremos en el punto correspondiente.

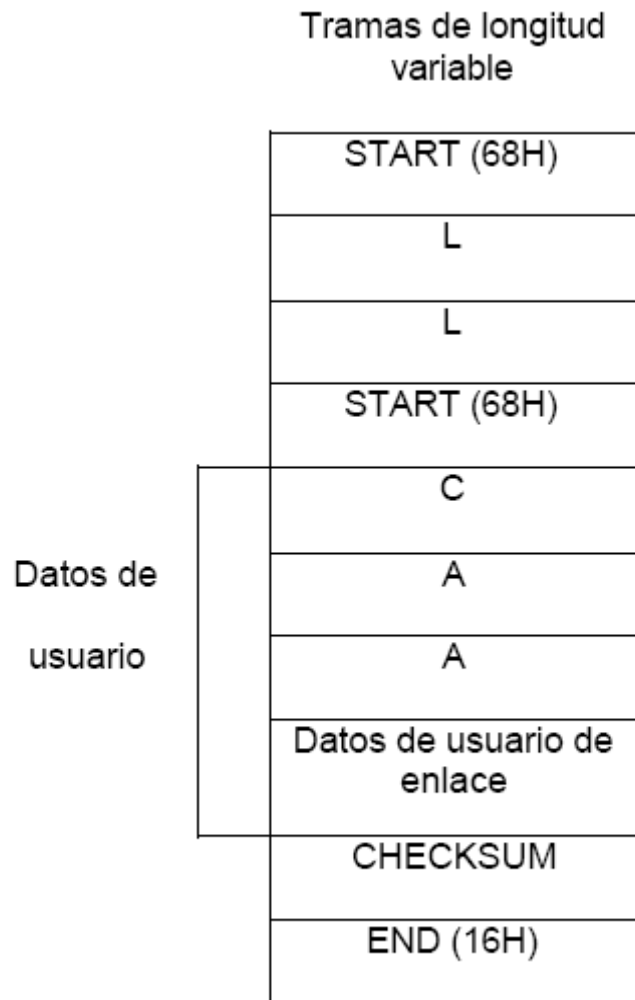


Figura 4 - Formato de una trama de longitud variable

Para la distinción entre tramas de longitud fija y variable en el momento de la recepción de una ráfaga, se especifica que los campos de START de cada tipos de trama tenga un código distinto, 10H en caso de trama de longitud fija y 68H en caso de tramas de longitud variable.

El campo etiquetado como A en las figuras (de dos bytes) codifica la dirección de enlace del terminal. Cada contador tiene una única dirección de enlace, y conexiones con un determinado terminal que incluyan una dirección incorrecta no obtendrán ninguna respuesta por parte del mismo.

El campo etiquetado como L en la figura 2, es el campo de longitud de la zona de datos de usuario, y su valor debe estar entre 0 y 255, siendo por tanto 261 bytes la longitud máxima de una trama.



El campo etiquetado como C en las figuras, es el campo de control. Este campo codifica la misma información en ambos sentidos de la información (del servidor al contador o viceversa), excepto para los bits 5 y 6.

La información que incluyen los campos comunes es la siguiente:

- El bit RES tiene un valor fijo de '0'.
- El bit PRM indica el sentido de la información, '1' en caso de ir hacia el contador y '0' en caso contrario.
- El Código de función informa al receptor de qué tipo de información se está transmitiendo. Se utiliza para detallar si es una transmisión SEND/CONFIRM, SEND/NO REPLY, REQUEST/RESPOND y para comunicar la aceptación o no de las tramas enviadas. Al final de este punto detallaremos los valores válidos para este campo en función del sentido de la comunicación.

Si la comunicación es del servidor al contador el resto de campos son los codificados de según la figura 3:

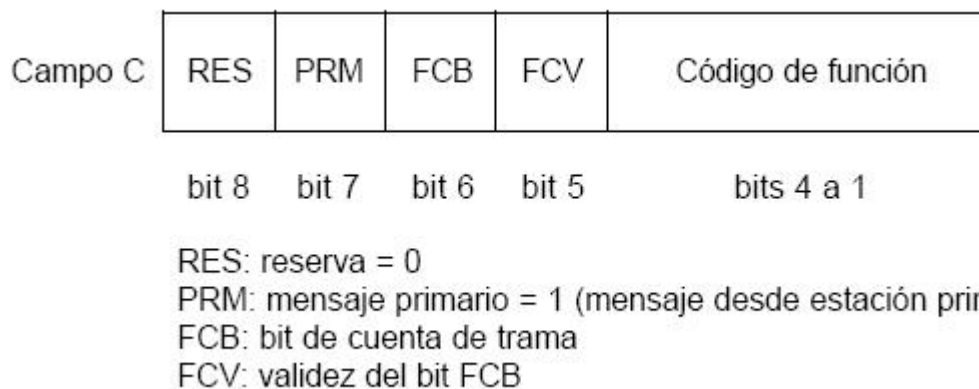


Figura 5 - Campo de control. Comunicación servidor-> contador

El bit FBC es un bit cuyo valor cambia en tramas consecutivas, como medida de control para evitar pérdida o repetición de tramas.

El bit FCV controla el fin de una transmisión. Si el valor es '1' se considera válido el valor anterior del bit FBC ya que se trata de una misma comunicación con varias tramas consecutivas. Si su valor es cero, se obvia el valor de FCB, ya que empieza una comunicación nueva con esta trama.

Si la comunicación es del contador al servidor los campos son los codificados de siguiente figura:



RES: reserva = 0

PRM: mensaje primario = 0 (mensaje desde estación secundaria)

ACD: demanda de acceso

DFC: control de flujo de datos

Figura 6 - Campo de control. Comunicación contador->servidor

El bit ACD tiene un valor de '0' si no se permite el acceso a datos para las siguientes solicitudes, y un valor de '1' en caso contrario.

El bit DFC tiene un valor de '0' si el envío de sucesivos mensajes será tratado por el contador y un valor de '1' si este envío pudiera causar overflow en el buffer de almacenamiento de mensajes.

Valores válidos para el Código de Función:

Los valores válidos para este campo en caso de tratarse de una comunicación hacia el contador (campo PRM del byte de control a '1'), son los siguientes. Añadimos a su descripción la situación en la que deben ser utilizados:

- 0: reposición del enlace remoto. A utilizar en comunicaciones tipo *SEND/CONFIRM*, con FCV = 0 (inicio de la comunicación).
- 3: envío de datos de usuario con confirmación. A utilizar en comunicaciones tipo *SEND/CONFIRM*, con FCV = 1 (resto de la comunicación).
- 9: solicitud de estado del enlace. A utilizar en comunicaciones tipo *REQUEST/RESPOND*, con FCV = 0 (inicio de la comunicación).
- 11: solicitud de datos de clase 2. A utilizar en comunicaciones tipo *REQUEST/RESPOND*, con FCV = 1 (resto de la comunicación).

En caso de tratarse de una comunicación desde el contador (campo PRM del byte de control a '0'), los valores aceptados son los siguientes:

- 0: ACK: reconocimiento positivo (tipo *CONFIRM*).
- 1: NACK: mensaje no aceptado, línea ocupada (tipo *CONFIRM*).
- 8: datos de usuario (tipo *RESPOND*).
- 9: NACK: datos solicitados no disponibles (tipo *RESPOND*).
- 11: estado del enlace o demanda de acceso (tipo *RESPOND*).



El resto de valores no usados, tanto para comunicación hacia el contador o recibiendo datos del mismo, están especificados en el estándar IEC 870-5-2, pero no son utilizados y, por tanto, no son válidos en el protocolo propuesto por Red Eléctrica de España.

1.3- Nivel de aplicación.

En el nivel de aplicación define todos los servicios disponibles para el usuario. Estos servicios se codifican en unas bloques llamados Unidad de Datos de Servicios de Aplicación (Application Service Data Units, ASDUs). Los ASDU están formados por una Unidad Identificadora de Datos seguida de uno o más objetos de información, dependiendo del tipo de ASDU concreto.

1.3.1- Unidad Identificadora de Datos

La Unidad Identificadora de Datos, es la parte del ASDU en la que se incluyen los datos necesarios para el control de la comunicación.

El formato de los ASDU es fijo y tiene la siguiente forma:

Tipo	
0	Cualificador de estructura variable
Causa de transmisión	
Dirección del punto de medida	
Dirección de registro	

Figura 7 - Formato de los la Unidad Identificadora de Datos

Cada campo de la Unidad Identificadora de Datos tiene tamaño de un byte, excepto la dirección del punto de medida cuya longitud es dos bytes.

Descripción de los campos de la Unidad Identificadora de Datos.

- Tipo: Este campo indica el tipo de trama que sigue. Cada tipo de trama tiene un Objeto de Información distinto asociado, así que el protocolo realiza distintas acciones para cada valor distinto de este campo. Los identificadores de tipo permitidos son los siguientes:



Tabla 1 - Tipos de ASDU

Id.	Uso	Mnemónico
1	Información de evento (<i>single-point</i>) con etiqueta de tiempo. Se empleará en la transmisión de incidencias	M_SP_TA_2
8	Totales integrados operacionales, 4 octetos (lecturas de contadores absolutos, en kWh o kVARh)	M_IT_TG_2
11	Totales integrados operacionales repuestos periódicamente, 4 octetos (incrementos de energía, en kWh o kVARh)	M_IT_TK_2
71	Identificador de fabricante y equipo. En lugar de un código de producto se enviará un identificador de equipo	P_MP_NA_2
72	Fecha y hora actuales	M_TI_TA_2
100	Leer identificador de fabricante y equipo	C_RD_NA_2
102	Leer registro de información de evento (<i>single-point</i>) por intervalo de tiempo	C_SP_NB_2
103	Leer fecha y hora actuales	C_TI_NA_2
122	Leer totales integrados operacionales por intervalo de tiempo y rango de direcciones	C_CI_NT_2
123	Leer totales integrados operacionales repuestos periódicamente por intervalo de tiempo y rango de direcciones	C_CI_NU_2
128	Firma electrónica de los totales integrados (lecturas)	M_DS_TA_2
129	Parámetros del punto de medida	P_ME_NA_2
130	Firma electrónica de los totales integrados repuestos periódicamente (incrementos de energía)	M_DS_TB_2
131	Fechas y horas de cambio de horario oficial	M_CH_TA_2
132	Carga de Clave Privada de Firma	C_PK_2
133	Leer Información de Tarificación (Valores en Curso)	C_TA_VC_2
134	Leer Información de Tarificación (Valores Memorizados)	C_TA_VM_2
135	Información de Tarificación (Valores en Curso)	M_TA_VC_2
136	Información de Tarificación (Valores Memorizados)	M_TA_VM_2
137	Cerrar Período de Facturación	C_TA_CP_2
138	Reservado para versiones futuras del protocolo RM-CM	
139	Bloques de totales integrados operacionales (lecturas de contadores absolutos, en kWh o kVARh)	M_IB_TG_2
140	Bloques de totales integrados operacionales repuestos periódicamente (incrementos de energía, en kWh o kVARh)	M_IB_TK_2
141	Leer la configuración del equipo RM.	C_RM_NA_2
142	Envío de la configuración del equipo RM.	M_RM_NA_2
143	Modificación de la configuración de los puertos de comunicaciones.	C_MR_NA_2
144	Lectura de potencias de contrato.	C_PC_NA_2
145	Envío de potencias de contrato.	M_PC_NA_2
146	Modificación de potencias de contrato.	C_MC_NA_2
147	Lecturas de días festivos.	C_DF_NA_2
148	Envío de días festivos.	M_DF_NA_2
149	Modificación de días festivos.	C_MF_NA_2
150-179	Reservados para versiones futuras del protocolo RM-CM	



180	Leer firma electrónica de los totales integrados por intervalo de tiempo (lecturas)	C_DS_TA_2
181	Cambiar fecha y hora	C_CS_TA_2
182	Leer los parámetros del punto de medida	C_PI_NA_2
183	Iniciar sesión y enviar clave de acceso	C_AC_NA_2
184	Leer firma electrónica de los totales integrados repuestos periódicamente, por intervalo de tiempo (incrementos de energía)	C_DS_TB_2
185	Leer fechas y horas de cambio de horario oficial	C_CH_TA_2
186	Modificar fechas y horas de cambio de horario oficial	C_MH_TA_2
187	Finalizar sesión	C_FS_NA_2
188	Reservado para versiones futuras del protocolo RM-CM	
189	Leer bloques de totales integrados operacionales por intervalo de tiempo y dirección	C_CB_NT_2
190	Leer bloques de totales integrados operacionales repuestos periódicamente por intervalo de tiempo y dirección	C_CB_NU_2
191–199	Reservados para versiones futuras del protocolo RM–CM	
200–255	Uso libre para cada fabricante	

- **Cualificador de estructura variable:** Este campo tiene como primer bit siempre el valor 0, seguido de un número que indica cuántos Objetos de Información le siguen.
- **Causa de transmisión:** este campo tiene la siguiente forma:

T	P/N	Causa (6 bits)
---	-----	----------------

Figura 8 - Formato de la causa de transmisión

Siendo T un bit que toma valor '1' si se trata de una trama de test, P/N un bit que toma valor '1' si se la trata es una confirmación positiva o '0' si se trata de una confirmación negativa o no se trata de un mensaje de confirmación. El resto del byte indica la causa de transmisión de la trama, de entre las indicadas en la Tabla 2.



Tabla 2 - Causas de transmisión.

Causa	Significado de la causa de transmisión
4	Inicializada
5	Petición o solicitada
6	Activación
7	Confirmación de activación
8	Desactivación.
9	Desactivación confirmada
10	Finalización de la activación
13	Registro de datos solicitado no disponible
14	Tipo de ASDU solicitado no disponible
15	Número de registro en el ASDU enviado por CM desconocido
16	Especificación de dirección en el ASDU enviado por CM desconocida
17	Objeto de información no disponible
18	Período de integración no disponible
48–52	Reservados para versiones futuras del protocolo RM–CM
53–63	Uso libre para cada fabricante

- Dirección del punto de Medida: Es una dirección fija para cada contador para la correcta toma de datos.
- Dirección de registro: es la dirección usada para acceder a las distintas informaciones disponibles en los contadores. Las permitidas en este protocolo son las siguientes:



Tabla 3 - Direcciones de registro

Dirección de registro	Uso
0	Dirección de defecto
11	Totales integrados con período de integración 1 (curva de carga)
12	RESERVA. [Posible uso futuro para Totales integrados con período de integración 2 (curva de carga, habitualmente cuartohoraria)].
13	RESERVA. [Posible uso futuro para Totales integrados con período de integración 3 (curva de carga)]
21	Totales integrados (valores diarios) con período de integración 1 (resumen diario)
22	RESERVA. [Posible uso futuro para Totales integrados (valores diarios) con período de integración 2 (resumen diario)]
23	RESERVA. [Posible uso futuro para Totales integrados (valores diarios) con período de integración 3 (resumen diario)]
52	Información de evento (<i>single-point</i>), sección 1: incidencias de arranques y tensión bajo límites
53	Información de evento (<i>single-point</i>), sección 2: incidencias de sincronización y cambio de hora
54	Información de evento (<i>single-point</i>), sección 3: incidencias de cambio de parámetros
55	Información de evento (<i>single-point</i>), sección 4: errores internos
128	Información de evento (<i>single-point</i>), sección 5: incidencias de intrusismo
129	Información de evento (<i>single-point</i>), sección 6: incidencias de comunicaciones
130	Información de evento (<i>single-point</i>), sección 7: incidencias de clave privada
131	Información de evento (<i>single-point</i>), sección 8: incidencias de Contrato I
132	Información de evento (<i>single-point</i>), sección 9: incidencias de Contrato II
133	Información de evento (<i>single-point</i>), sección 10: incidencias de Contrato III
134	Información de Tarificación relativa al Contrato I
135	Información de Tarificación relativa al Contrato II
136	Información de Tarificación relativa al Contrato III
137	Información de Tarificación relativa al Contrato Latente I
138	Información de Tarificación relativa al Contrato Latente II
139	Información de Tarificación relativa al Contrato Latente III
140-199	Reservados para versiones futuras del protocolo RM-CM
200-255	Uso libre para cada fabricante



1.3.2- Objetos de Información.

Los objetos o elementos de información son la parte de la trama con una información útil para el usuario. En esta parte del ASDU se incluye la información de las lecturas, sobre las fechas, incidencias, etc. que componen las lecturas.

Cada tipo de ASDU tiene puede tener un tipo de elemento de información distinto, si bien, varios tipos de ASDUs distintos comparten objetos de información con el mismo formato.

A continuación expondremos algunos ejemplos de ASDU, incluyendo su Unidad Identificadora de Datos y sus Elementos de Información y su modo de transmisión. Para más información acerca del formato de los ASDU

1.3.3- Ejemplos de ASDUs

ASDU Tipo 141: C RM NA 2:

Este ASDU es el empleado para leer la información de configuración del equipo. Su Unidad Identificadora de datos tiene la siguiente forma:

Tipo = 141	
0	0
Causa de transmisión	
Dirección del punto de medida	
Dirección de registro = 0	

Figura 9 - ASDU Tipo 141

En este ASDU la causa de transmisión debe ser obligatoriamente 5 (petición) en caso de ser una trama enviada hacia el contador, o 14 (tipo de ASDU solicitado no disponible) si es una trama enviada desde el contador, indicando un error en el envío (la dirección de medida era incorrecta, por ejemplo).

Si la petición enviada al contador es correcta, el contador respondería con el siguiente tipo de trama:

ASDU Tipo 142: M RM NA 2:

Este ASDU contiene la información de configuración del contador, su formato es el siguiente:



Tipo = 142	
0	1
Causa de transmisión	
Dirección del punto de medida	
Dirección de registro = 0	
Información de configuración (246 octetos)	

Figura 10 - ASDU Tipo 142

La causa de transmisión de este ASDU será siempre 5 (solicitada) y con el bit de P/N de ese byte con valor '1', ya que es una confirmación positiva.

2- Sistema de automatización de lecturas remotas.

Para este módulo del proyecto no se siguió ninguna especificación fijada en un protocolo, se diseñó pensando en las necesidades básicas que debería implementar un sistema de este tipo.

El diseño propuesto quedó de la siguiente manera:

- Ampliación de las clases elaboradas en la anterior parte del proyecto para que gestionen la comunicación a través de módems.
- Creación de un sistema para gestionar múltiples lecturas concurrentes, sobre diferentes puertos serie (conectados a un módem cada uno), realizando llamadas en paralelo.

En este módulo contamos con la colaboración de Fabio Mesquita Buiati que añadió a la funcionalidad que acabamos de describir los siguientes puntos:

- Almacenamiento de las lecturas en una base de datos.
- Automatización de las lecturas para que se repitan cada cierto periodo de tiempo

2.1- Lecturas remotas.

Las lecturas se deben realizar a través de módems GSM conectados al servidor por los puertos serie. Para realizar esta tarea tenemos dos objetivos



que cumplir, por una parte, existe una necesidad de gestionar las llamadas y, por otra parte, hay que leer los datos requeridos del contador al que llamamos.

Para el primer objetivo, se modificarán las clases que establecían comunicación por el puerto serie al contador, añadiendo unos métodos para la inicialización de los módems y para la gestión de la llamada. Tanto la inicialización de los módems como la gestión de la llamada se realizan a través de comandos AT. Más adelante se explicará en detalle los pasos seguidos para el establecimiento de las llamadas, el intercambio de datos y la finalización de las llamadas.

Para el segundo objetivo, se utilizaron las herramientas ya creadas para la lectura de terminales locales, ya que una vez iniciada la conexión de datos entre el módem conectado a nuestro servidor y el módem conectado al contador, los detalles relacionados con el canal de comunicación se vuelven transparentes para ambas partes.

2.2- Lecturas concurrentes.

El siguiente objetivo fijado es la posibilidad de realizar varias lecturas por un número no fijo de módems, siendo tampoco fijo el número total de lecturas a realizar. Hay un número máximo de reintentos para cada lectura, en caso de sobrepasar este límite, la lectura no se realiza.

Parte III: Especificación y requisitos.

1- Protocolo de comunicaciones entre registradores y concentradores de medidas o terminales portátiles de lectura

1.1- Especificación del protocolo.

La especificación concreta de este protocolo puede consultarse en el documento que define el mismo. Como resumen podemos indicar a que es un protocolo utilizado para obtener datos de configuración y lecturas de contadores eléctricos para su posterior tratamiento. Las informaciones que se pueden obtener son las expuestas en el la Tabla 1.

2- Sistema de automatización de lecturas remotas.

2.1- Especificación del sistema.

No existe ninguna especificación fijada externamente por ninguna organización ni comité. Así pues la especificación utilizada es la mencionada en el apartado de diseño del sistema.



3- Requisitos de la Sección II.

A) Requisitos funcionales

A.1) Conseguir plena funcionalidad del protocolo de comunicaciones entre registradores y concentradores de medidas o terminales portátiles de lectura.

B) Requisitos de Implementación

B.1) Utilizar un lenguaje de implementación que incluya librerías de comunicación a través del puerto serie. En su defecto, encontrar librerías externas de comunicación serie.

B.2) Utilizar el mismo formato de elementos de información en el código que el usado por los contadores

C) Requisitos Temporales

C.1) Debido a la dependencia con la segunda parte del proyecto, (implementación de *CPCSCD*), esta parte del proyecto debe de estar terminada en a comienzos del primer trimestre de 2006.

C.2) Deberán respetarse los plazos de entregas temporales que el profesor de proyecto tenga a bien fijar.

D) Requisitos Software/Hardware

En cuanto a requisitos software sólo son necesarios un equipo con el entorno de desarrollo Java JDK1.5 y la librería extra javax.comm para la comunicación a través de puertos serie.

En cuanto a requisitos hardware para el desarrollo del proyecto fue necesario tener acceso a:

- Contadores eléctricos para las pruebas de lecturas.
- PCs o servidores con algún puerto serie, o en caso de no disponer de ninguno, adaptadores de puerto USB a puerto serie.

Parte IV: Implementación

1.- El lenguaje y el entorno



Para esta parte del proyecto se utilizó como lenguaje de programación Java, y como entorno de desarrollo Eclipse, todo sobre Red Hat linux 9.

También fue necesario el uso de la librería javax.comm. Esta librería fue la herramienta que utilizamos para el manejo de los puertos serie del servidor. Este API permite un total control de la comunicación tanto por puertos serie como por puertos paralelo, ya que permite trabajar con todo tipo de configuraciones (distinta velocidad de transmisión, número de bits por ráfaga, paridad, etc.), así como control sobre las líneas físicas a través de eventos que se activan con la activación de las señales de control del puerto.

2.- Descripción general

En primer lugar describiremos qué funciones tienen las principales clases que implementan el sistema:

- Contador: esta clase contiene la información de los contadores que necesita el sistema, como puede ser la dirección del punto de medida, clave (enviada en la trama de inicio de sesión), o número de teléfono y número de reintentos para la comunicación de contadores remotos.
- Trama: proporciona algunos métodos para el manejo de las tramas.
- ControladorCanal: esta clase proporciona la funcionalidad que permite el manejo de las capas física y de enlace. Permite la creación del canal de comunicación por el puerto serie y proporciona métodos para su manejo. También se encarga iniciar la sesión y extraer los datos de las tramas de la capa de enlace.
- Inicializador: esta clase tiene la funcionalidad de la capa de aplicación. Permite la recepción y envío de ASDUs. Además permite diferenciar entre comunicación a un contador local o a un contador remoto, en cuyo caso debe almacenar la información obtenida en la base de datos.
- ProgramadorTareas: esta clase es la que implementa la posibilidad de acceso concurrente a distintos contadores utilizando simultáneamente diferentes puertos serie.

3.- Protocolo de comunicaciones entre registradores y concentradores de medidas o terminales portátiles de lectura

Para la correcta implementación del protocolo es necesario que las comunicaciones se efectúen de la manera que indican los siguientes diagramas de secuencia.

3.1. Diagrama de secuencia para el inicio de sesión

Para iniciar una sesión en este protocolo es necesario que seguir los pasos expuestos en la siguiente figura:

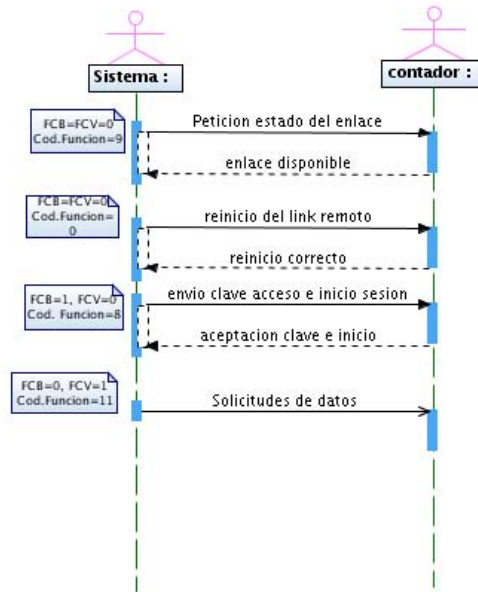


Figura 11 - Diagrama de secuencia para el inicio de sesión

3.2. Diagrama de secuencia para las transmisiones.

El siguiente diagrama presenta el modo de envío de información en la frontera entre el nivel de enlace y el de aplicación. Primero se envía una trama pidiendo información acerca de un tipo de información. Esta trama es de tamaño variable con tipo de ASDU que especifique la información solicitada en concreto. Esta información pertenece a la capa de aplicación, la información de la capa de enlace está en el campo de control, que indica que la trama se envía con solicitando un tipo de información (SEND). Si el contador contesta de manera positiva (CONFIRM), enviaremos una trama solicitando los datos pedidos (REQUEST) y el contador nos devolverá una trama de longitud variable con un ASDU del tipo correspondiente a la respuesta de la ASDU solicitado por el otro extremo. La información de la capa de enlace vuelve a estar en el código de función del campo de control (RESPOND).

En el diagrama mostramos la información referente a la capa de aplicación en el exterior del mismo y la referente a la capa de enlace en el interior.

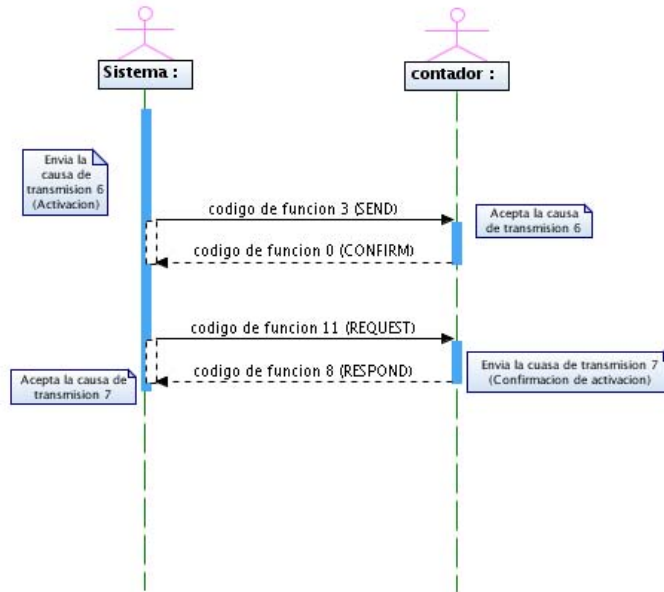


Figura 12- Diagrama de secuencia para una transmisión

4.- Sistema de automatización de lecturas remotas.

Para la permitir la posibilidad de acceder a un número indeterminado de contadores de manera simultánea usando todos los recursos posibles del sistema, utilizamos el siguiente modelo de comportamiento para nuestro diseño:

1. Obtener el número de puertos serie utilizables.
2. Encolar todos los contadores que se tiene que leer.
3. Mientras la cola no esté vacía y haya puertos libres para realizar una lectura se genera una nueva lectura concurrente. Si no hay contadores que leer se espera a que se añada un nuevo contador a la cola y en caso de que no haya puertos disponibles se espera a la liberación de alguno.
4. Al generar una nueva lectura se le asigna al contador a leer un puerto disponible y éste se elimina temporalmente de la lista de puertos utilizables.
5. Se intenta realizar la llamada al contador remoto, en caso de fallo se pasa al paso 6 de la secuencia alternativa 1. En caso de éxito se sigue en el paso 6 de esta secuencia.
6. Una vez establecida la llamada, se pasa a leer los datos del contador. En caso de error se pasa al paso 7 de la secuencia alternativa 2. En caso de éxito en la lectura se sigue con el paso 7 de esta secuencia.
7. Se almacenan los valores leídos en la base de datos.
8. Se comunica al módem que debe pasar de modo de datos a modo de comandos y se finaliza la llamada.



9. Se libera el puerto ocupado por esta lectura, pasando a estar de nuevo disponible.

Secuencia alternativa 1

6. Se para la lectura concurrente en cuestión, se libera el puerto, pasando a estar disponible de nuevo.
7. Se refleja en la clase que representa al contador el intento fallido.
8. Si el número máximo de reintentos ha sido alcanzado, se elimina de la cola de contadores a leer el contador afectado, y en otro caso se vuelve a añadir al final de la cola.

Secuencia alternativa 2

7. Se comunica al módem que debe pasar de modo de datos a modo de comandos y se finaliza la llamada.
8. Se libera el puerto ocupado por esta lectura, pasando a estar de nuevo disponible.
9. Se refleja en la clase que representa al contador el intento fallido.
10. Si el número máximo de reintentos ha sido alcanzado, se elimina de la cola de contadores a leer el contador afectado, y en otro caso se vuelve a añadir al final de la cola.



Sección II: CPCSCC

Esta parte de nuestro proyecto se encarga de la implementación de los Web services necesarios para el Proceso de Validación de Datos de Red Eléctrica Española con Certificación de Resultados.

Dicha implementación, ha tenido como referencia fundamental el documento "Protocolo de Comunicaciones entre Concentradores de Medida. Solución basada en "Web Services" Especificación Funcional de 20 de febrero de 2004 desarrollado por Red Eléctrica Española.

Parte I: fundamentos técnicos

En este proyecto hemos usado tecnologías que no han sido estudiadas a lo largo de la carrera en su mayor parte. Para comprender el alcance del proyecto y su funcionalidad, primero comenzaremos con una introducción teórica a las tecnologías empleadas.

Omitiremos los estándares o arquitecturas ampliamente conocidas en este punto, como JAVA o http. Nos centraremos en las tecnologías clave del proyecto que nos eran desconocidas a principio de curso. XML, XML-RCP, Web Services, SOAP, WSDL, UDDI

1.- XML

XML (eXtensible Markup Language, 'lenguaje de marcado extensible'), es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). No es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. Más adelante comentaremos algunas de esas tecnologías y estándares.

El uso principal dado al XML es el paso de mensajes entre programas. Cuando los desarrolladores deciden implementar un sistema de comunicación basado en Web services, (como en este proyecto), XML es la elección natural. Fundamentalmente existen dos protocolos de comunicación bajo XML: XML-RPC y SOAP.

XML-RPC es un protocolo muy simple que usa mensajes XML para realizar RPCs. Las peticiones son codificadas en XML y enviadas vía http-POST. La respuesta XML está empotrada en el cuerpo de la respuesta http.



Como XML-RPC es una plataforma independiente, permite comunicar aplicaciones muy diversas entre sí. A pesar de ser un buen punto de partida para implementar web-services, XML-RPC no tiene un servicio de descripción gramatical, algo fundamental para permitir conexiones *just-in-time*.

2.- Servicios Web

Un **Servicio Web** es una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes y ejecutadas sobre cualquier plataforma pueden utilizar los servicios Web para intercambiar datos en Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

La principal razón para usar servicios Web es que se basan en HTTP sobre TCP en el puerto 80. Los servicios Web se realizan por este puerto y ello los hace muy convenientes. Además, son muy prácticos porque tienen un débil acoplamiento entre una aplicación que usa el servicio Web y el propio servicio. De esta forma los cambios que cada uno realice con el tiempo no deben afectar al otro.

Un servicio Web o *Web Service* se basa en los siguientes estándares:

- Web Services Protocol Stack: Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- XML: Es el formato estándar para los datos que se vayan a intercambiar. Ya ha sido comentado previamente
- SOAP o XML-RPC: Protocolos sobre los que se establece el intercambio.
- Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP, FTP, o SMTP.
- WSDL: Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- UDDI: Protocolo para publicar la información de los servicios Web. Permite a las aplicaciones comprobar qué servicios Web están disponibles.
- WS-Security: Protocolo de seguridad aceptado como estándar por OASIS. Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.
- WS-Attachments: es un estándar propuesto de servicio Web que aprovecha las ventajas del protocolo DIME de encapsulación de



mensajes diseñado para facilitar el envío de archivos adjuntos con los mensajes SOAP

Arquitectura de los servicios web

Hay dos puntos de vista en al arquitectura de los servicios Web. El primero son los roles que juegan cada uno de los actores y el segundo es la pila del protocolo

A) Roles en un servicio web

Existen tres roles principales

1. El proveedor de servicios, *Service Provider*. este es el proveedor del servicio Web. El Service provider implementa el servicio y lo hace disponible en Internet
2. El peticionario del servicio, o *consumidor*. Utiliza un servicio Web existente abriendo una conexión y enviando una petición XML
3. El registro de servicio. Es un directorio centralizado lógico de servicios. El registro ofrece un lugar donde los desarrolladores pueden publicar nuevos servicios o encontrar los ya existentes.

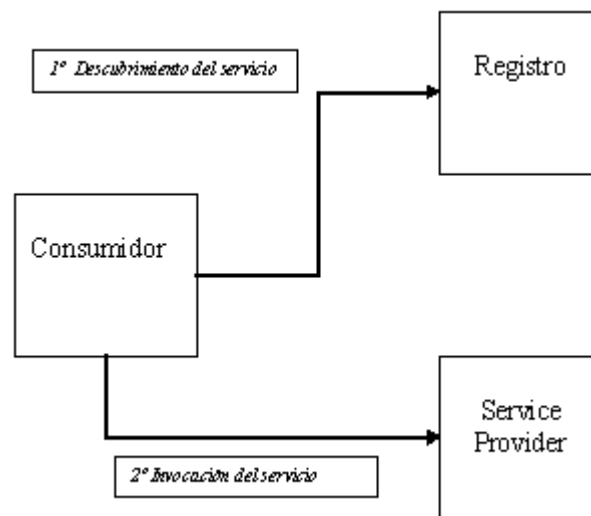


Figura 13 - Interactuación entre los roles de un servicio web

Las anteriores tecnologías y características pueden organizarse en una pila tal como se muestra en la siguiente figura.

B) La pila de protocolo Web Service

La pila tiene cuatro capas que se organizan como muestra la siguiente tabla. A continuación explicamos brevemente cada una de las capas



Capa del protocolo	Tecnología
Descubrimiento	UDDI
Descripción	WSDL
Mensajería XML	XML-RPC, SOAP
Transporte	http

Figura 14 - La pila del protocolo Web-Service

UDDI, el descubrimiento del servicio

UDDI es la capa de descubrimiento en la figura 2. Incluye dos conceptos,

1. UDDI es una especificación para construir directorios distribuidos de Web services. Los datos son almacenados en un formato XML específico. La especificación UDDI incluye detalles en un API para búsquedas de datos existentes y publicación de nuevos datos
2. el UDDI Business Registry es una implementación completamente operacional de la especificación UDDI. Desde que fue puesta en funcionamiento por IBM y Microsoft en Mayo de 2001, el registro UDDI permite cualquier búsqueda de datos UDDI. También incorpora un método para que cualquier compañía pueda registrarse a sí misma o a sus servicios.

Los datos en UDDI

Los datos en UDDI se divide en tres categorías principales

- Páginas blancas: esta categoría incluye información general sobre una compañía específica, como su nombre, la descripción de su negocio y la dirección
- Páginas amarillas: esta categoría incluye una clasificación general de los datos para cada uno de los servicios ofrecidos. Por ejemplo, puede incluir información sobre identificadores de códigos de barras para determinados productos
- Páginas verdes: esta categoría incluye información técnica sobre un Web service. En realidad, una página verde es un puntero a una especificación externa y una dirección para invocar el Web service.

WSDL, la descripción del servicio web

WSDL representa la capa de descripción del servicio Web en el punto 5 de la parte IV (Implementación). Básicamente, WSDL es código XML que especifica la funcionalidad del servicio Web, y le provee de una interfaz pública, que otros servicios Web (o aplicaciones), pueden entender. Dicha interfaz puede incluir información sobre los métodos o funciones disponibles, información sobre los tipos de datos con los que trabajan los mensajes XML o información de direccionamiento específica.

No es estrictamente necesario incluir un servicio WSDL en un protocolo de mensajería XML, pero SOAP lo incorpora y hace un uso intensivo de él.



Al usar WDSL, un cliente puede localizar un Web Service e invocar cualquiera de sus funciones públicas. Existen numerosas herramientas automáticas que automatizan este proceso, permitiendo una fácil integración.

Nuestro protocolo también usa WSDL para describir los servicios Web que implementa.

SOAP

SOAP (siglas de *Simple Object Access Protocol*) es un protocolo estándar creado por Microsoft, IBM y otros, que está actualmente bajo el auspicio de la W3C y define cómo dos objetos en diferentes procesos pueden comunicarse intercambiando datos XML.

A diferencia de DCOM y CORBA, que son binarios, SOAP usa el código fuente en XML, que facilita la eliminación de errores, pero es menos efectivo. El término *Object* en el nombre significa que se adhiere al paradigma de la programación orientada a objetos.

SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí.

SOAP funciona sobre cualquier protocolo de Internet, generalmente http. Tiene como base XML, con un diseño que cumple el patrón Cabecera-Desarrollo de diseño de software. La cabecera *Header* es opcional y contiene meta datos sobre enrutamiento (*routing*), seguridad o transacciones. El desarrollo *Body* contiene la información principal, que se conoce como carga útil (*payload*). La carga útil se acoge a un *XML Schema* propio.

Como vemos, SOAP es más potente como protocolo que XML-RPC, pero también considerablemente más complejo. Los esquemas aumentan la complejidad, así como los espacios de nombres. El tamaño de un mensaje SOAP es también mayor al de un mensaje XML-RPC. En la implementación del protocolo, hemos utilizado SOAP como protocolo de mensajería, debido precisamente a la posibilidad de usar esquemas en los mensajes.

HTTP, la capa de transporte

No comentaremos la capa HTTP por no tener relevancia específica para este proyecto.



3- DIME

Se trata de un protocolo diseñado para encapsular un mensaje SOAP y sus archivos adjuntos relacionados. Al igual que con SOAP, los mensajes DIME, (DIME, Direct Internet Message Encapsulation), se pueden enviar mediante protocolos de transporte estándar, como HTTP, TCP y UDP. DIME admite datos de secuencia e incluso se puede utilizar sin SOAP, aunque en este caso la capacidad de DIME de describir el contenido de los mensajes es limitada. El objetivo de DIME es transportar de manera más eficiente archivos adjuntos junto con los mensajes SOAP, lo que es especialmente útil en el caso de servicios Web que necesitan incluir archivos binarios de gran tamaño, como archivos multimedia o archivos de datos binarios.

También es posible enviar datos en un mensaje SOAP sin DIME, aunque este proceso aumenta desproporcionadamente el tamaño a enviar si los archivos adjuntos son grandes y puede ser incluso especialmente difícil si están firmados digitalmente.

DIME ha sido optimizado para su uso con mensajes SOAP. Al contar con los meta datos que ya existen en un mensaje SOAP, un analizador DIME no tiene que realizar la pesada tarea de leer todos los meta datos que hay en el propio mensaje DIME, lo que aumenta la rapidez y la eficacia con la que se analizan los mensajes DIME

4.- WS-Attachments

WS-Attachments es un estándar propuesto de servicio Web que aprovecha las ventajas del protocolo DIME de encapsulación de mensajes diseñado para facilitar el envío de archivos adjuntos con los mensajes SOAP.

Utiliza la encapsulación directa de mensajes de Internet para enviar y recibir mensajes SOAP con archivos adjuntos adicionales, como por ejemplo archivos binarios, fragmentos de XML o incluso otros mensajes SOAP

Parte II. Análisis y Diseño

1.- CPCSCC

El protocolo a implementar recibe el nombre *“Elegibilidad 2003, Protocolo de Comunicaciones entre concentradores de Medida. Solución basada en Web Service”*. (De ahora en adelante, CPCSCC. Ha sido diseñado por Red Eléctrica de España (<http://www.ree.es>) en base a distintos protocolos, documentos y convenciones que debemos mostrar en este punto.

CPCSCC es un protocolo que permite comunicar concentradores secundarios con puntos de medida eléctricos. Los concentradores secundarios almacenan información (*concentran información*) y pautas sobre



consumos. Los puntos de medida calculan el consumo producido, y sus datos asociados (hora del mismo, picos de consumo, identificación, etc.).

El nivel de abstracción en *CPCSCC* es de capa de aplicación. El diseño completo del protocolo no comprende todas las capas de datos de la pila OSI. Cada una de las capas inferiores es definida por una normativa internacional, detallada en los documentos 870-5-X, (con X en el rango [1..5]).

CPCSCC es la evolución del protocolo CP-CS, (protocolo de comunicación entre **Concentradores Primarios** y **Concentradores Secundarios**), diseñado por la empresa Indra. CP-CS era una especificación que estaba basada en las transmisiones de tramas de información obtenidas directamente de los contadores, encapsulándolas en objetos propios y proporcionando software a nivel de capa de aplicación que solucionase su envío.

Como hemos comentado en el punto anterior, este escenario era el ideal para un desarrollo basado en Web Services, y *CPCSCC* es precisamente lo que aporta.

2.- Diagrama de Frontera del protocolo

A continuación se muestra el diagrama de frontera del protocolo.

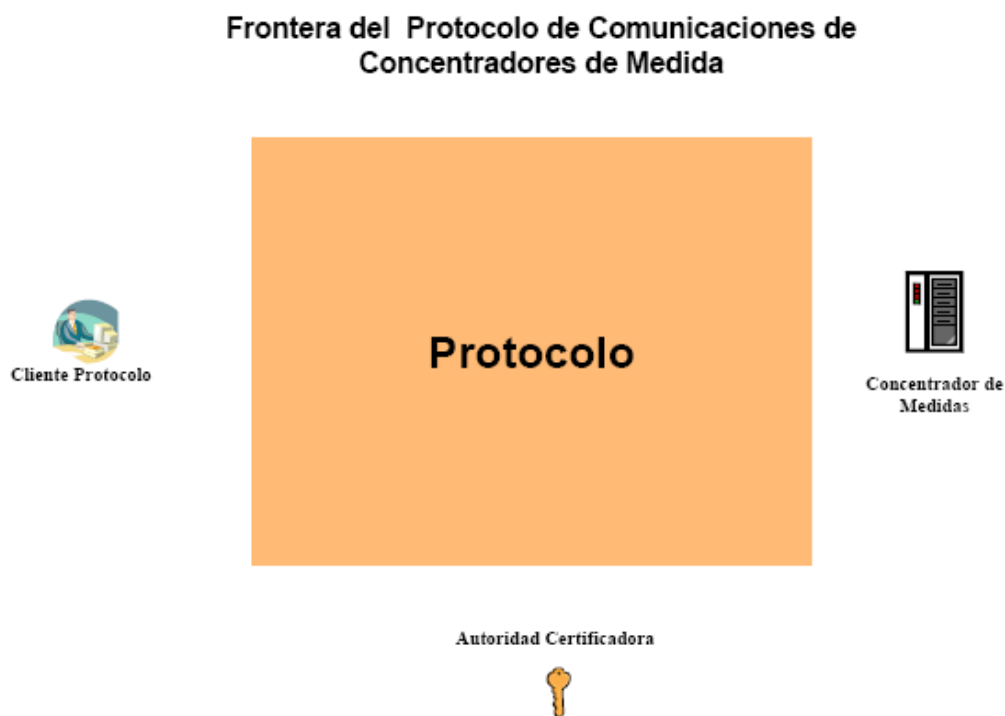


Figura 15 - Diagrama de frontera del protocolo CPCSCC



Tal y como se ve en el diagrama de frontera se han detectado los siguientes actores externos:

- Cliente Protocolo. Son aplicaciones que utilizan la funcionalidad aportada por el protocolo para establecer comunicaciones con otros Concentradores de medidas. Estos actores serán aplicaciones que utilizarán las librerías de invocación de servicios del prototipo.
- Autoridad Certificadora. Será la encargada de proporcionar los certificados necesarios para realizar la comunicación https con los Concentradores de Medidas.
- Concentrador de Medidas. Son los sistemas externos con los que el protocolo debe soportar la comunicación.

Los objetivos que se pretende cubrir con el protocolo de comunicaciones de Concentradores de Medida son los siguientes:

- Proporcionar una forma de comunicación entre Concentradores de Medida basada en estándares.
- Proporcionar una forma de comunicación que sea aplicable a todas las comunicaciones entre los distintos tipos de Concentradores de Medida del mercado eléctrico.
- Proporcionar una especificación del protocolo que permita su implementación por terceros que deseen comunicar con los Concentradores de Medida del mercado eléctrico.
- Se pretende que el protocolo no este basado en productos propietarios.

El protocolo dispondrá de la siguiente funcionalidad básica:

- Consulta de información disponible en concentradores remotos.
- Recogida de información disponible en concentradores remotos.
- Confirmación de recogida de ficheros a concentradores remotos.
- Envío de información Disponible a concentradores remotos.

Hasta este momento hemos destacado los aspectos más relevantes del análisis y diseño del proyecto. Complementando a lo anterior, adjuntamos en el apéndice 1 el Documento *CPCSCC*, donde termina de detallarse el diseño y análisis

Diagramas UML

En este apartado mostramos los diagramas UML de esta sección. Se detallan los diagramas de clases, y los diagramas de secuencia de los servicios Web.

Diagrama de clases

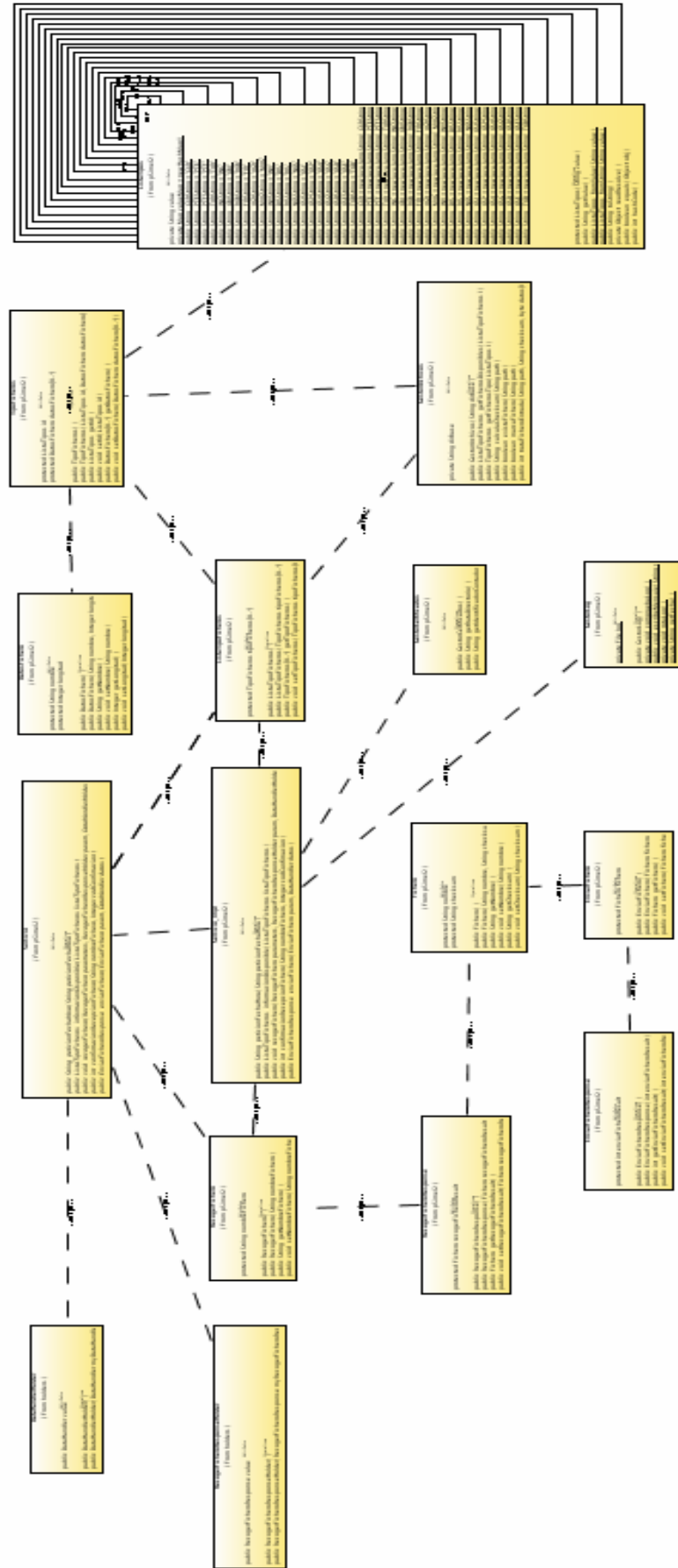


Figura 16 - Diagrama de clases del protocolo CPCSCC



Diagramas de Secuencia

Confirmar fichero

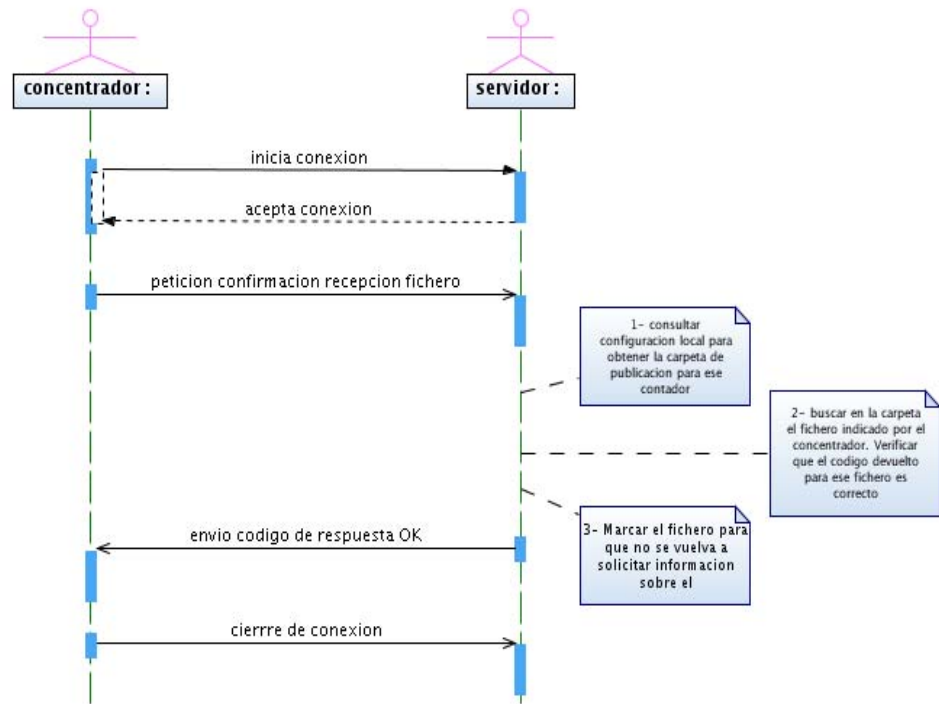


Figura 17 - Diagrama de secuencia para Confirmar Fichero

Enviar Fichero

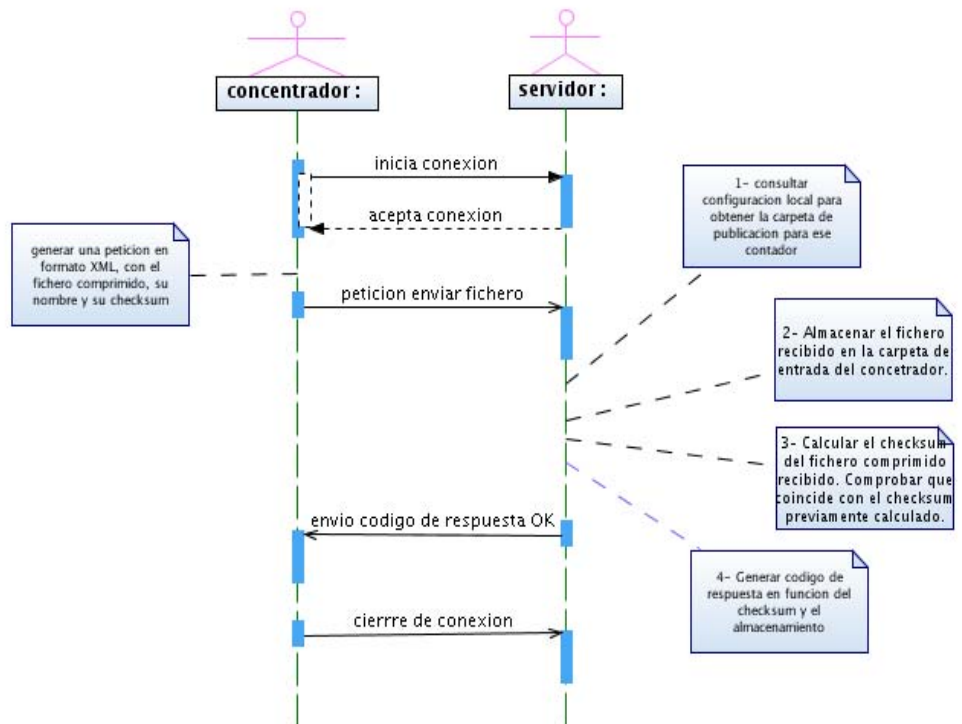


Figura 18 - Diagrama de secuencia para Enviar Fichero



Información disponible

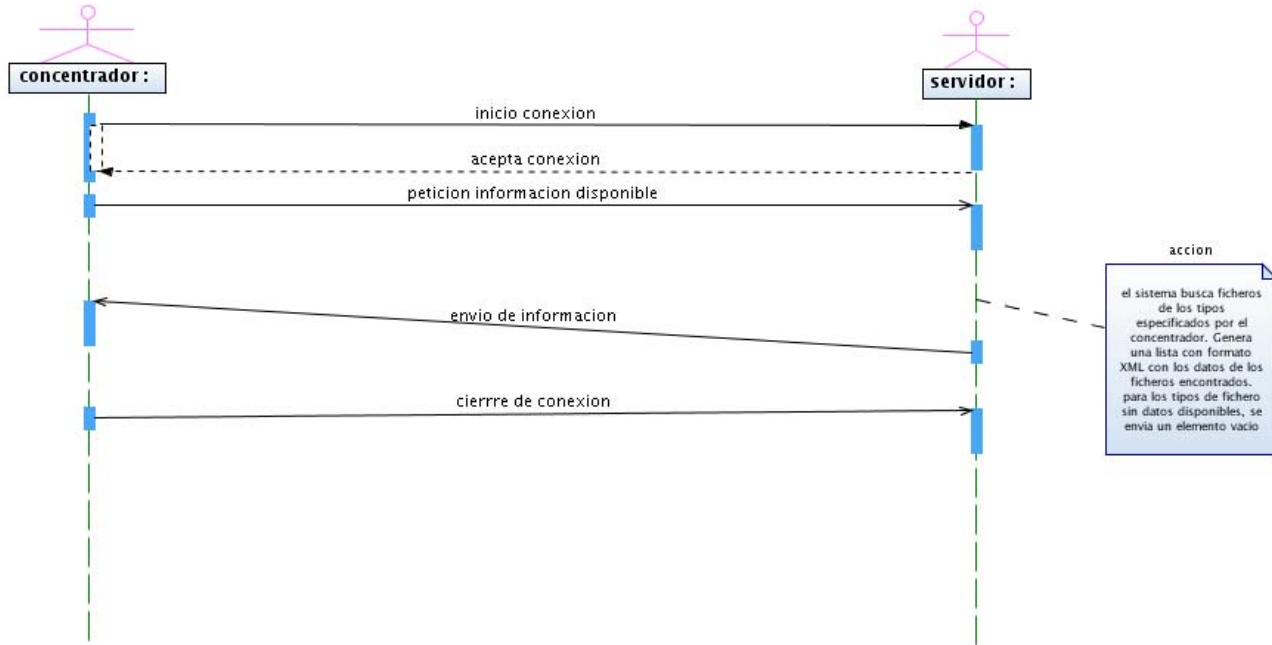


Figura 19 - Diagrama de secuencia para Información Disponible

Petición Fecha hora

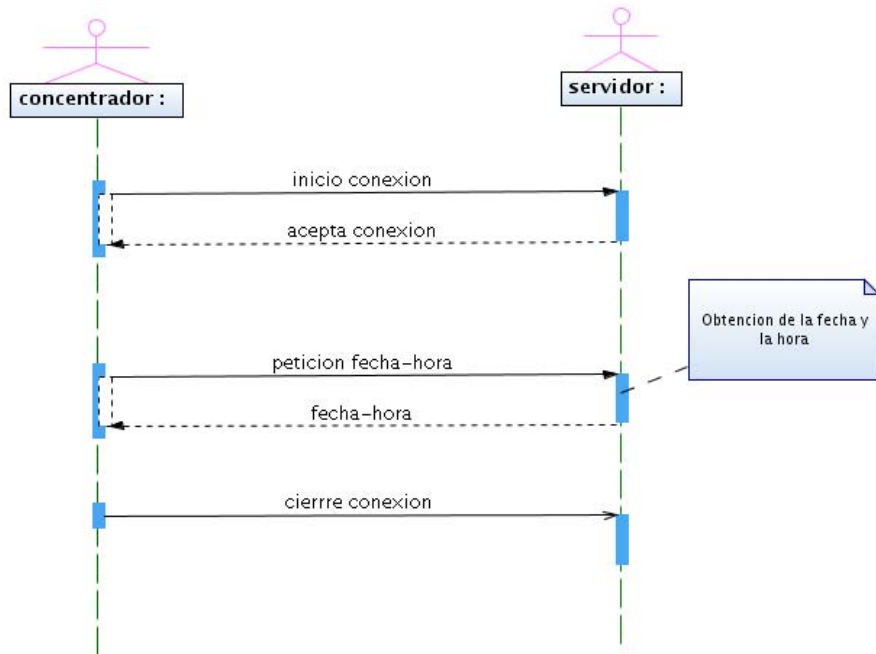


Figura 20 - Diagrama de secuencia para Petición Fecha y Hora

Recoger fichero

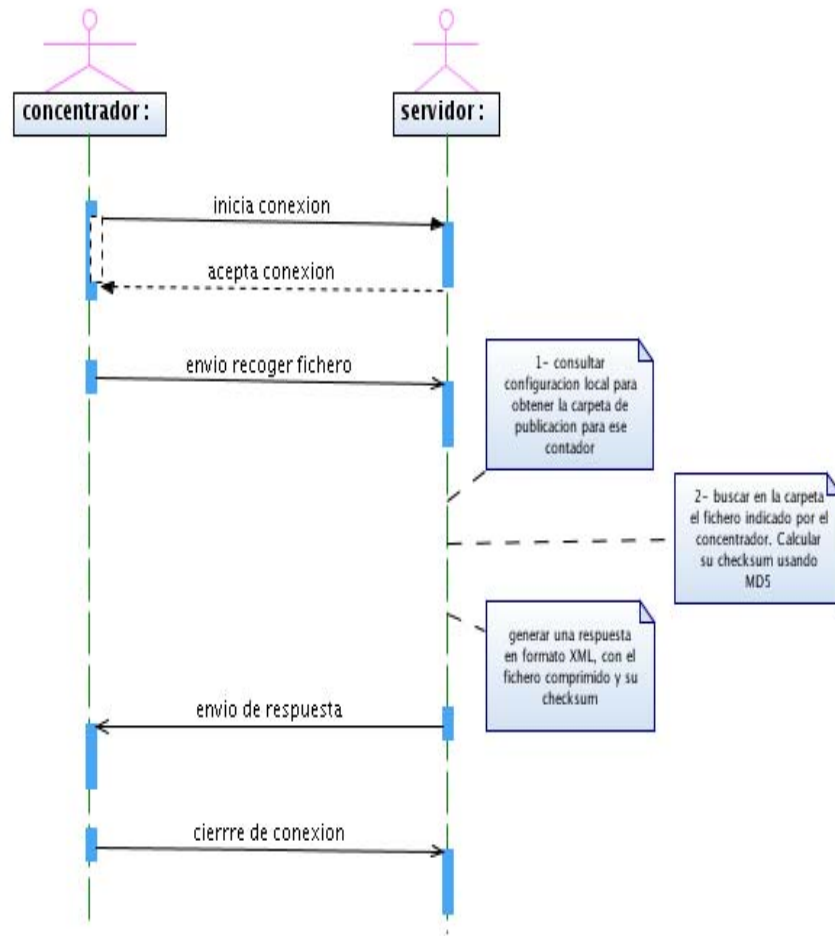


Figura 21 - Diagrama de secuencia para Recoger Fichero

Parte III: especificación y requisitos

1.- Especificación concreta de los servicios Web a implementar

Tal y como se detalló en la introducción teórica a los Web Services, la descripción de los mismos se detalla en su WSDL. Dicha especificación técnica se encuentra en la página 41 del Apéndice 1.

2.- Requisitos del proyecto

Sin perder de vista que este es un proyecto de sistemas Informáticos, podemos clasificar los requisitos de nuestro proyecto siguiendo un orden típico de Ingeniería del Software

A) Requisitos funcionales



A.1) Conseguir plena funcionalidad del protocolo CPCSCC: la implementación deberá poder funcionar individualmente a cualquier otro módulo de protocolos 870-5-X implementados previamente

A.2) Satisfacer las pruebas que realiza Red Eléctrica de España. Para más información, consultar el apartado "*Pruebas del sistema*"

B) Requisitos de Implementación

B.1) Conseguir un desarrollo basado en la plataforma de Sun Microsystems, J2SE: Este requisito era necesario para tener compatibilidad en los datos de prueba con una aplicación desarrollada en el año 04 por alumnos de la FDI. En dicha aplicación existían datos y tablas de concentradores secundarios y puntos de medida.

B.2) Requisitos de interfaces: NA.

C) Requisitos Temporales

El proyecto debe de estar terminado en la primera semana de Julio de 2006, para entrega en la secretaría de Alumnos y evaluación del profesor. Además, deberán cumplirse los plazos de las entregas temporales que el tutor de proyecto establezca.

D) Requisitos Software/Hardware

No hay ninguna restricción en cuanto a limitaciones en la adquisición de licencias propietarias, ni en la utilización de máquinas servidoras. El único requisito es la necesidad de instalar *Apache Tomcat* o *Sun Application Server* para que el Service Provider pueda ofrecer el servicio Web a cualquier cliente que lo desee.

Parte IV: Implementación

1.- El lenguaje y el entorno

Tal y como estaba detallado en el apartado de requisitos, la implementación ha sido desarrollada íntegramente en Java. El entorno utilizado ha sido NetBeans IDE 5.5 bajo Red Hat linux 9 y utilizando como servidor de aplicaciones el Sun Java System Application Server 8.

Gracias a las herramientas incluidas con NetBeans 5.5, hemos tenido más fácil la implementación de los servicios Web especificados en el documento CPCSCC. Como se ha comentado anteriormente, este documento incluye el archivo WSDL con la descripción de los servicios. NetBeans permite la creación de un conjunto de Web services a partir de un fichero con formato .wsdl, generando automáticamente la estructura de clases



.java necesarias para su funcionamiento. Sin embargo, hemos tenido que redefinir algunos de los tipos de entrada y de salida de los Web services. Estos cambios, han sido debidos a la necesidad de transferir ficheros envueltos con SOAP (WS-Attachments) entre el cliente y el servidor del protocolo. En el apartado 5.- Descripción de los servicios Web. Archivo cpcsc.wSDL se da el nuevo listado WSDL y a continuación se muestra el funcionamiento de los servicios Web implementados, definiendo primero los diferentes tipos construidos de los que se harán uso:

2.- Tipos construidos

- TipoFichero. Contiene los ficheros asociados a un tipo.

```
struct TipoFichero{
    enum id = {"CUR", "PTE", "PTF", "TAR", "INC", "OBJ", "ROB", "FIR", "AGR", "MAG", "INV",
              "DIS", "RIS", "NOS", "OSI", "OSP", "OSA", "OSG", "OSE", "OSD", "TOD"};
    DatosFichero datos;
}
```

- DatosFichero. Contiene un atributo nombre con el nombre del fichero, y otro longitud con su tamaño en kilobytes.

```
struct DatosFichero{
    string nombre;
    int longitud;
}
```

- Fichero. Esta estructura esta formada por un elemento Nombre que contiene un string con el nombre del fichero, un elemento Checksum que es un entero con el checksum obtenido al aplicar el algoritmo MD5 a los datos comprimidos bzip2 del fichero. El checksum consistirá en una cadena de texto que representará los bytes de la cadena hexadecimal obtenida. El checksum se codificará como una cadena hexadecimal escrita utilizando los caracteres "0..9" "a..f", es decir las letras se codificarán con minúsculas.

```
struct Fichero{
    string Nombre;
    byte Checksum;
}
```

3.- Descripción de los Servicios Web

Petición de Fecha y Hora

Las peticiones de fecha y hora las atenderá el servicio Web *peticionFechaHora* con urn: *peticionFechaHora*

Argumentos: no tiene

Retorno: devuelve una cadena de caracteres (xsi:string) con el formato: "DD/MM/AAAA hh:mm:ss".



Ejemplo

La llamada a la función debe responder al siguiente esquema:

```
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:msg="http://simel.ree.es/cpcsc/2003/Mensajes/"
xmlns:ref="http://schmeas.xmlsoap.org/ws/2002/04/">
  <soap-env:Body>
    <msg:peticionFechaHora>
    </msg:peticionFechaHora>
  </soap-env:Body>
</soap-env:Envelope>
```

Este es el retorno que se obtendría al hacer una llamada al servicio *peticionFechaHora*

```
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:msg="http://simel.ree.es/cpcsc/2003/Mensajes/"
xmlns:ref="http://schmeas.xmlsoap.org/ws/2002/04/">
  <soap-env:Body>
    <msg:peticionFechaHoraResponse>
    <msg:peticionFechaHoraResult>02/04/2003 12:00:00</msg:peticionFechaHoraResult>
    </msg:peticionFechaHoraResponse>
  </soap-env:Body>
</soap-env:Envelope>
```

Secuencia de eventos

Acciones sobre el Protocolo	Acciones del Protocolo
1. Un Concentrador de Medidas remoto inicia una conexión https con el servidor del protocolo.	2. El sistema verifica la identidad del Concentrador. 3. El sistema acepta la conexión.
4. El Concentrador de Medidas remoto invoca al servicio <i>peticionFechaHora</i>	5. El sistema obtiene la fecha y hora de la máquina en la que se ejecuta. 6. El sistema envía la fecha y la hora obtenidas al Concentrador de Medidas que la solicitó.
7. El Concentrador de Medidas cierra la conexión https.	
Secuencia alternativa de eventos: No se puede verificar identidad del Concentrador.	
Acciones sobre el Protocolo	Acciones del Protocolo
	3. El sistema rechaza la conexión.
4. El Concentrador de Medidas cierra la conexión https	



Información Disponible

Las peticiones de Información Disponible las atenderá el servicio web *informacionDisponible* con urn: *informacionDisponible*.

Argumento: listaTipos de tipo ListaTiposFichero (esquema *InformacionDisponible.xsd*).

listaTipos: Es una estructura que contiene la lista de tipos de ficheros que se desean consultar.

La estructura *ListaTipoFicheros* contiene una lista de elementos *TipoFichero*.

Cuando esta estructura se utiliza como argumento de entrada, los elementos *TipoFichero* no contendrán listas de elementos *DatosFichero*. Solo tendrán el identificador del tipo de fichero.

La siguiente tabla muestra los tipos de fichero posibles, junto con su descripción y las letras que por las que empiezan los archivos de cada tipo de fichero según el documento "*Ficheros para el intercambio de información de medidas (Versión 11, 1 junio de 2006)*" de REE. Estas letras nos sirven para poder informar al usuario de los tipos de fichero disponibles. Entre paréntesis aparecen los capítulos correspondientes al documento mencionado:

Tabla 4 - Tipos de fichero y descripción

Tipo	Descripción	Distintivo/s
CUR	Curva de Carga Horaria de Punto de Medida(3.1.1), Curva de Carga Horaria de Punto Frontera(3.1.2), Curva de Carga Cuarto Horaria de Punto de Medida (3.1.3).	P1, F1 y P2.
PTE	Medidas de Punto de Horarias Pendientes(3.1.14), , Saldos de Contador Pendientes(3.1.17), Firmas de Puntos de Medida Pendientes(3.1.15),	P1P, PSAL, FIRP
PTF	Medidas de Punto de Frontera Horarias Pendientes(3.1.16), Medidas Agregadas pendientes (3.2.4)	FP, MAGR3
TAR	Cierres de Contrato 1, 2 y 3 de Punto de Medida (3.1.4), Lecturas Instantáneas de Contrato 1, 2 y 3 de Punto de Medida(3.1.5), Saldo de Lectura de Facturación (3.1.6).	CN, SN, SALD
INC	Incidencia en punto de medida (3.1.7), Eventos de Registrador (3.1.8). Anulación de medidas en punto frontera (3.1.18).	INCI, LV, ANF1
OBJ	Objeciones de Medidas Horarias(3.1.9) , Objeciones a Medidas no Horarias(3.1.11)	OBJE, OBJES



ROB	Respuesta de Objeciones Horarias (3.1.10), Respuestas a Objeciones de Medidas no Horarias(3.1.12).	REOB, REOBS
FIR	Firmas de Curvas de Carga Horaria de Punto de Medida(3.1.13).	P1FIR
AGR	Definición de Agregaciones(3.2.1), Baja de Agregaciones (3.2.2)	AGREE, BAJAGREE
MAG	Medidas Agregadas(3.2.3), Medidas agregadas formato de longitud variable (3.2.4)	MAGR2, MAGR3
INV	Inventario de Puntos de Medidas de Clientes(3.3.2), Solicitud de Modificación de Inventario de Puntos de Medidas de Clientes (3.3.3) , Inventario de Equipos de Medida de Clientes (3.3.5), Solicitud de Modificación de Inventario de Equipos de Clientes (3.3.6), Notificación Fecha de Lectura (3.3.9)	PMCLIE, SOLPMCLIE, EQUIPOS, SOLEQUIPOS, LECT
RIS	Respuesta a Solicitud de Modificación de Inventario de Puntos de Medidas de Clientes (3.3.4), Respuesta a Solicitud de Modificación de Inventario de Equipos de Medida(3.3.7)	RESPMCLIE, RESEQUIPOS
NOS	Notificación de Alta de Punto de Suministro de Cliente Tipo 1 ó 2 al Operador del Sistema(3.3.10), Notificación de Modificación Cambio de Comercializador de Clientes Tipo 1 al Operador del Sistema(3.3.11). Notificación modificación clientes a OS (3.3.12), Notificación baja de clientes (3.3.13), Corrección errores clientes (3.3.14)	ALCL, CACL, CACL2, BACL2, RECT
OSI	Datos de Inventario OS. Varios ficheros de acuerdo 3.4.1	CLOS, CLOS2
OSP	Datos Horarios de Energía por Punto Frontera y Magnitud del Participante 1(3.4.2.1), Datos Horarios de Energía por Punto Frontera y Magnitud del Participante 2 (3.4.2.2), Datos Horarios de Energía por Punto Frontera y Magnitud del Participante 2 Correspondientes a Puntos Frontera de un Participante 1 (3.4.2.3.), Datos Horarios de las Pérdidas de Transporte (3.4.2.34).	EPFPF, PER
OSA	Acumulados Mensuales de Energía por Punto Frontera y Magnitud del Participante 1 (3.4.3.1), Acumulados Mensuales de Energía por Punto Frontera y Magnitud del Participante 2 (3.4.3.2), Acumulados Mensuales de Energía por Punto Frontera y Magnitud del Participante 2 Correspondientes a Puntos Frontera de un Participante 1 (3.4.3.3).	acum.
	Inventarios de agregaciones tipo 3 y siguientes (3.4.4.1), Acumulados mensuales de energía por agregación del participante 1 (distribuidora) (3.4.4.2), Acumulados mensuales de energía por agregación del participante 2 (comercializadora) (3.4.4.3), Acumulados mensuales de energía por agregación del participante 1 (distribuidor) correspondientes a agregaciones de un participante 2 (comercializadores) (3.4.4.4), Datos horarios diarios de energía por agregación del participante 1	AGREEOS, ACUMAGREOS,



OSG	(distribuidor)(3.4.4.5), Datos horarios diarios de energía por agregación del participante 2 (comercializadora) (3.4.4.6), Datos horarios diarios de energía por agregación del participante 1 (distribuidor) correspondientes a agregaciones de un participante 2 (comercializadora) (3.4.4.7)	AGREOS2
OSE	Ficheros de error de datos enviados al OS	Extensión .bad2
OSP	Demanda del Sistema (3.5.1.) , Perfiles Finales (3.5.2).	DEMR, PERFF
TOD	No es un tipo de fichero, esta indicación es para poder recibir información de ficheros de todos los tipos descritos anteriormente.	---

Quando se solicite información disponible de tipo "TOD", éste es el único tipo que debe aparecer en el mensaje de petición, ya que no es lógico pedir por ejemplo información disponible de tipo "TOD" y de tipo "CUR" a la vez porque el primer tipo engloba al segundo.

A continuación se muestra un ejemplo de esta estructura:

```
<?xml version="1.0" encoding="UTF-8"?>
<ListaTipoFicheros xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="argInformacionDisponible.xsd">
  <TipoFichero id="CUR"/>
  <TipoFichero id="TAR"/>
  <TipoFichero id="OBJ"/>
  <TipoFichero id="AGR"/>
  <TipoFichero id="INV"/>
  <TipoFichero id="OSP"/>
  <TipoFichero id="OSA"/>
</ListaTipoFicheros>
```

Retorno: Devuelve una estructura del mismo tipo que el argumento de entrada (descrito en el esquema informacionDisponible.xsd).

Los ficheros que se encuentren disponibles para ese concentrador se listarán creando una lista de elementos DatosFichero por cada tipo de fichero. Si se ha solicitado información de tipo "TOD" se responderá con un listado de todos los ficheros de los que disponga discriminando entre tipos. Aquellos tipos de los que no disponga de ficheros se enviarán como elementos vacíos. A continuación se muestra un ejemplo de esta estructura de salida:

```
<?xml version="1.0" encoding="UTF-8"?>
<listaTipoFicheros xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="retInformacionDisponible.xsd">
  <TipoFicheros id="CUR" />
  <TipoFicheros id="TAR" />
  <TipoFicheros id="INC" >
    <DatosFichero nombre="P2_021_20030410.01" longitud="10" />
    <DatosFichero nombre="F2_021_20030410.01" longitud="50" />
  </TipoFicheros>
</listaTipoFicheros>
```

Secuencia de Eventos:

Acciones sobre el Protocolo	Acciones del Protocolo



<p>1. Un Concentrador de Medidas remoto inicia una conexión https con el servidor del protocolo.</p>	<p>2. El sistema verifica la identidad del Concentrador.</p>
<p>4. El Concentrador de Medidas remoto invoca al servicio <i>informacionDisponible</i> con el una estructura XML con los tipos de fichero que desea.</p>	<p>3. El sistema acepta la conexión.</p> <p>5. El sistema consulta su configuración para obtener a dirección de la carpeta de publicación para ese concentrador.</p>
<p>8. El Concentrador de Medidas cierra la conexión https.</p>	<p>6. El sistema busca en esa carpeta ficheros de los tipos especificados por el concentrador. Genera una lista con formato XML con los nombres de los ficheros encontrados, sus tamaños y sus fechas de publicación. Para los tipos de fichero solicitados de los que no dispongan ficheros publicados (en ese momento, o bien no los proporcione nunca) se generará un elemento vacío para cada uno de esos tipos de ficheros.</p> <p>7. El sistema envía la lista al Concentrador de Medidas que lo solicitó.</p>
<p>Secuencia alternativa de eventos: No se puede verificar identidad del Concentrador.</p>	
<p>Acciones sobre el Protocolo</p>	<p>Acciones del Protocolo</p>
<p>4. El Concentrador de Medidas cierra la conexión https</p>	<p>3. El sistema rechaza la conexión.</p>
<p>Secuencia alternativa de eventos: Se solicita el tipo "TOD" más algún otro tipo.</p>	
<p>Acciones sobre el Protocolo</p>	<p>Acciones del Protocolo</p>
<p>8. El Concentrador de Medidas cierra la conexión https.</p>	<p>6. El sistema genera un fallo SOAP de clase Client.TOD.</p> <p>7. El sistema envía el fallo.</p>

Recoger Fichero



Las peticiones de recoger fichero las atenderá el servicio Web recogerFichero con urn: recogerFichero

Argumentos: nombreFichero: Es un string que contiene el nombre del fichero que se desea recoger.

Retorno: Devuelve un tipo Fichero.

El tipo de información que contiene el elemento secundario se identificará con la etiqueta application/x-bzip2 por estar comprimido con formato bzip2. En el caso de que se solicite recoger un fichero que no existe, se retornará como error esta misma estructura con todos los campos vacíos.

Ejemplo

La llamada a la función debe responder al siguiente esquema:

```
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:msg="http:// simel.ree.es/cpcsc/2003/Mensajes"
xmlns:ref="http://schmeas.xmlsoap.org/ws/2002/04/">
  <soap-env:Body>
    <msg:recogerFichero>
      <msg:nombreFichero> F1_20030402.01</msg:nombreFichero >
    </msg:recogerFichero>
  </soap-env:Body>
</soap-env:Envelope>
```

Este es el retorno que se obtendría al hacer una llamada al servicio

```
000001 0 0 0001 00000000000000000000
00101010 0101010 0101011110100101010
101001 010101 01010101010100101010
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:msg="http:// simel.ree.es/cpcsc/2003/Mensajes"
xmlns:ref="http://schmeas.xmlsoap.org/ws/2002/04/">
  <soap-env:Body>
    <msg:recogerFicheroResponse>
      <msg:recogerFicheroResult>
        <msg:Nombre>F1_20030401.01</msg:Nombre>
        <msg:Checksum>12434AEFCDFD2334EFDABC2345AE431A</msg:Checksum>
      </msg:recogerFicheroResult>
    </msg:recogerFicheroResponse>
  </soap-env:Body>
</soap-env:Envelope>
```

```
-----
000001 0 0 0001 00000000000000000000
00101010 0101010 0101011110100101010
101001 010101 01010101010100101010
application/x-gzip
<< 1.42 MB de datos binarios para F1_20030401_00001.01.XML >>
```

Secuencia de Eventos:

Acciones sobre el Protocolo	Acciones del Protocolo
1. Un Concentrador de Medidas remoto inicia una conexión https con el servidor del protocolo.	2. El sistema verifica la identidad del Concentrador.
	3. El sistema acepta la conexión.



<p>4. El Concentrador de Medidas remoto invoca al servicio recogerFichero. Le pasa como argumento el nombre del fichero que quiere recoger.</p> <p>11. El Concentrador de Medidas cierra la conexión https.</p>	<p>5. El sistema obtiene el identificador del Concentrador de Medidas que invoca el servicio.</p> <p>6. El sistema consulta su configuración para obtener a dirección de la carpeta de publicación para ese concentrador.</p> <p>7. El sistema busca en esa carpeta el fichero indicado por el concentrador. El fichero ya estará comprimido.</p> <p>8. El sistema calculará el checksum del fichero comprimido utilizando el algoritmo MD5.</p> <p>9. El sistema generará una respuesta en formato XML, en la que se introducirá el checksum del fichero, y el fichero comprimido.</p> <p>10. El sistema envía la respuesta al Concentrador de Medidas que lo solicitó.</p>
<p>Secuencia alternativa de eventos: No se puede verificar identidad del Concentrador.</p>	
<p>Acciones sobre el Protocolo</p> <p>4. El Concentrador de Medidas cierra la conexión https</p>	<p>Acciones del Protocolo</p> <p>3. El sistema rechaza la conexión.</p>
<p>Secuencia alternativa de eventos: El fichero solicitado no existe en el sistema.</p>	
<p>Acciones sobre el Protocolo</p> <p>10. El Concentrador de Medidas cierra la conexión https.</p>	<p>Acciones del Protocolo</p> <p>8. El sistema genera como respuesta un fallo SOAP de clase Client.Fichero.</p> <p>9. El sistema envía la respuesta al Concentrador de Medidas que solicitó el fichero.</p>

Confirmación de Recepción de Fichero



Las peticiones de confirmación de recepción de fichero las atenderá el servicio web confirmacionRecepcionFichero con urn: confirmacionRecepcionFichero.

Argumentos: Recibe los siguientes argumentos:

nombreFichero: Es un string con el nombre del fichero.

codConfirmacion: Es un entero con un código de confirmación. Los posibles códigos son los siguientes:

- 0. El fichero se confirma positivamente.
- 1. Error en la validación del checksum.
- 2. Error genérico.

Retorno: Devuelve uno de los siguientes códigos:

- 0. Confirmación correcta.
- 1. Error genérico al procesar el fichero.

Ejemplo

La llamada a la función debe responder al siguiente esquema:

Llamada a la función cuando la recepción ha sido correcta (código 0)

```
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:msg="http:// simel.ree.es/cpcsc/2003/Mensajes"
xmlns:ref="http://schmeas.xmlsoap.org/ws/2002/04/">
  <soap-env:Body>
    <msg:confirmacionRecepcionFichero>
      <msg:nombreFichero>F1_20030402.01</msg:nombreFichero>
      <msg:codConfirmacion>0</msg:codConfirmacion>
    </msg: confirmacionRecepcionFichero>
  </soap-env:Body>
</soap-env:Envelope>
```

Este es el retorno que se obtendría al hacer una llamada al servicio (en el caso de que todo ha ido bien)

```
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:msg="http:// simel.ree.es/cpcsc/2003/Mensajes"
xmlns:ref="http://schmeas.xmlsoap.org/ws/2002/04/">
  <soap-env:Body>
    <msg:confirmacionRecepcionFicheroResponse>
      <msg:confirmacionRecepcionFicheroResult>0</msg:confirmacionRecepcionFicheroResul
t>
    </msg: confirmacionRecepcionFicheroResponse>
  </soap-env:Body>
</soap-env:Envelope>
```

Secuencia de Eventos:

Acciones sobre el Protocolo	Acciones del Protocolo
1. Un Concentrador de Medidas remoto inicia una conexión https con el servidor del protocolo.	2. El sistema verifica la identidad del Concentrador.
	3. El sistema acepta la conexión.



<p>4. El Concentrador de Medidas remoto invoca al servicio confirmacionRecepcionFichero. Le pasa como argumentos el nombre del fichero que ha recogido y un código de confirmación.</p>	<p>5. El sistema obtiene el identificador del Concentrador de Medidas que invoca el servicio.</p> <p>6. El sistema consulta su configuración para obtener a dirección de la carpeta de publicación para ese concentrador.</p> <p>7. El sistema busca en esa carpeta el fichero indicado por el concentrador.</p> <p>8. El sistema verifica que el código devuelto para ese fichero es de fichero correcto.</p> <p>9. El sistema marca el fichero para que no se vuelva a considerar la próxima vez que ese Concentrador de Medidas le solicite información disponible.</p> <p>10. El sistema envía un código de respuesta indicando que se acepta la confirmación de recepción para ese fichero.</p>
<p>11. El Concentrador de Medidas cierra la conexión https.</p>	
<p>Secuencia alternativa de eventos: No se puede verificar identidad del Concentrador.</p>	
Acciones sobre el Protocolo	Acciones del Protocolo
<p>4. El Concentrador de Medidas cierra la conexión https</p>	<p>3. El sistema rechaza la conexión.</p>
<p>Secuencia alternativa de eventos: El fichero no existe en el sistema.</p>	
Acciones sobre el Protocolo	Acciones del Protocolo
<p>10. El Concentrador de Medidas cierra la conexión https.</p>	<p>8. El sistema genera una respuesta de error genérico (código 2).</p> <p>9. El sistema envía la respuesta al Concentrador de Medidas.</p>
<p>Secuencia alternativa de eventos: El código de confirmación del fichero es de error de checksum.</p>	
Acciones sobre el Protocolo	Acciones del Protocolo
	<p>9. El sistema graba una línea de log para registrar el evento.</p> <p>10. El sistema genera una respuesta de</p>



confirmación correcta (código 0).

11. El sistema envía la respuesta al Concentrador de Medidas.

12. El Concentrador de Medidas cierra la conexión https.

Enviar Fichero

Las peticiones de envío de fichero las atenderá el servicio Web enviarFichero con URN: enviarFichero

Argumentos: Recibe fichero que es un tipo Fichero descrito en la función recogerFichero (corresponde al esquema ficheroTransferido.xsd).
fichero: Es una estructura XML. La estructura es la misma que la empleada en el retorno del servicio recogerFichero así como el uso del encapsulado. En esta estructura se enviará el nombre del fichero, el checksum de sus datos comprimidos calculado con el algoritmo MD5 y los datos comprimidos en formato bzip2. El checksum consistirá en una cadena de texto que representará los bytes de la cadena hexadecimal obtenida.

Retorno: Devuelve un entero con los siguientes valores:

- 0. Envío correcto.
- 1. Error de checksum.
- 2. Error genérico.

Ejemplo

La llamada a la función debe responder al siguiente esquema:

```
000001 0 0 0001 00000000000000000000
00101010 0101010 0101011110100101010
101001 010101 0101010101010100101010
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:msg="http:// simel.ree.es/cpscsc/2003/Mensajes"
xmlns:ref="http://schmeas.xmlsoap.org/ws/2002/04/">
  <soap-env:Body>
    <msg:enviarFichero>
      <msg:fichero>
        <msg:Nombre>P1_20030410.01</msg:Nombre>
        <msg:Checksum>12434AEFCDFD2334EFDABC2345AE431A </msg:Checksum>
      </msg:ficheros>
    </msg:enviarFichero>
  </soap-env:Body>
</soap-env:Envelope>
-----
000001 0 0 0001 00000000000000000000
101001 010101 0101010101010100101010
P1_20030410_00001.01.XML
application/x-bzip2
<< 1.42 MB de datos binarios para P1_20030410_00001.01.XML >>
-----
```



Este es el retorno que se obtendría al hacer una llamada al servicio (cuando el envío ha sido correcto):

```
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:msg="http://simel.ree.es/cpscsc/2003/Mensajes"
xmlns:ref="http://schmeas.xmlsoap.org/ws/2002/04/">
  <soap-env:Body>
    <msg:recogerFicheroResponse>
      <msg:recogerFicheroResult>0</msg:recogerFicheroResult>
    </msg:recogerFicheroResponse>
  </soap-env:Body>
</soap-env:Envelope>
```

Secuencia de Eventos:

Acciones sobre el Protocolo	Acciones del Protocolo
1. Un Concentrador de Medidas remoto inicia una conexión https con el servidor del protocolo.	2. El sistema verifica la identidad del Concentrador. 3. El sistema acepta la conexión.
4. El Concentrador de Medidas remoto invoca al servicio enviarFichero. Le pasa como argumentos el nombre del fichero, el checksum del fichero comprimido, y el fichero comprimido en una estructura con formato XML.	5. El sistema obtiene el identificador del Concentrador de Medidas que invoca el servicio. 6. El sistema consulta su configuración para obtener a dirección de la carpeta de entrada para ese concentrador. 7. El sistema calcula el checksum del fichero comprimido que ha recibido. 8. El sistema verifica que el checksum calculado coincide con el checksum recibido. 9. El sistema almacena el fichero recibido en la carpeta de entrada del concentrador con el nombre que se ha recibido. El fichero se almacena en formato comprimido. 10. El sistema genera un código de respuesta en el que se notifica que el fichero ha sido enviado con éxito. Este código se genera en función de la validación del checksum y del resultado del almacenamiento. 11. El sistema envía la respuesta al Concentrador de Medidas.
12. El Concentrador de Medidas cierra la conexión https.	



Secuencia alternativa de eventos: No se puede verificar identidad del Concentrador.	
Acciones sobre el Protocolo	Acciones del Protocolo
4. El Concentrador de Medidas cierra la conexión https	3. El sistema rechaza la conexión.
Secuencia alternativa de eventos: El checksum calculado no coincide con el recibido.	
Acciones sobre el Protocolo	Acciones del Protocolo
11. El Concentrador de Medidas cierra la conexión https.	9. El sistema genera una respuesta indicando que el checksum no coincide. 10. El sistema envía la respuesta al Concentrador de Medidas.
Secuencia alternativa de eventos: Error genérico al procesar el fichero.	
Acciones sobre el Protocolo	Acciones del Protocolo
12. El Concentrador de Medidas cierra la conexión https.	10. El sistema genera como respuesta un fallo SOAP de clase Server.Generico. 11. El sistema envía la respuesta al Concentrador de Medidas.

4.- Detalles de Implementación

En este punto explicamos con algo más de detalle la implementación de los Web services, sirviéndonos de la descripción hecha en el punto anterior. Los diagramas de clases de los módulos explicados en este punto se encuentran en la Parte II. Análisis y diseño.

Gestor de Archivos

Este módulo de la aplicación se encarga de la gestión del sistema de archivos necesario para el almacenamiento ordenado de los diferentes archivos para cada uno de los concentradores de medidas registrados en el servidor *CPCSCC*.

También es el encargado de efectuar el checksum MD5 de los archivos tratados, haciendo uso de la clase *MessageDigest* incluida en el *jdk 1.5*.



Gestor de certificados

Este módulo se encarga de la correcta identificación de cada uno de los concentradores de medidas que se autentican contra el servidor CPCSCC. Realizada esta identificación, debe devolver a la implementación del servicio, el Distinguished Name (DN) obtenido directamente del certificado digital. Este campo DN deberá estar construido de la siguiente manera:

C: Contendrá el identificador del país al cual pertenece la organización para la cual se emite el certificado. En este caso su valor será siempre “es”.

O: Contendrá el identificador de la organización que emite el certificado:

REE

OU: Indicador de rama del directorio dedicada a Certificados de Concentradores de Medidas:

CN: Identificador de la organización a la que pertenece el Concentrador de Medidas:

CN: Contendrá el identificador del usuario final del certificado. Este identificador podrá ser de los siguientes tipos:

- Servidor: En el caso de que el usuario final del certificado sea un servidor de aplicaciones que implemente los servicios Web del protocolo, este campo contendrá una dirección el “hostname” de su URL.

- Cliente: Los certificados de clientes del protocolo contendrán en este campo un identificador de cliente.

Esta identificación está declarada como OBLIGATORIA y se realizará a través del certificado digital utilizado para establecer la conexión SSL entre el cliente y el servidor.

Gestor de Logs

También hemos dotado a la aplicación de un sistema de gestión de los logs que realiza un rotado de los ficheros en los que se almacenan en caso de llegar a un tamaño máximo fijado.

Web Services

Como hemos comentado a lo largo de este texto, la herramienta utilizada para la implementación del protocolo CPCSCC nos ha permitido la creación de la estructura básica de clases para la puesta en funcionamiento de los servicios Web. Un de las clases autogeneradas por NetBeans (Servicio_Impl.java) alberga el núcleo de la implementación de todos y cada uno de los servicios Web en forma de métodos internos a la clase.



Nuestra parte ha consistido fundamentalmente en dotar de funcionalidad a esos métodos contruidos a partir del archivo .wsdl. Esto lo hemos realizado basándonos en las definiciones de los Web services que hemos expuesto en el punto 3 de esta misma sección.

5.- Descripción de los servicios Web. Archivo cpcsc.wSDL

```
<?xml version="1.0" encoding="utf-8" ?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ree="http://localhost:8102/proyectocpcs"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  targetNamespace="http://localhost:8102/proyectocpcs"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
<types>
  <xsd:schema elementFormDefault="qualified" targetNamespace="http://localhost:8102/proyectocpcs">
    <xsd:element name="peticionFechaHora">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element minOccurs="1" maxOccurs="1" name="peticionFechaHora" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="peticionFechaHoraResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element minOccurs="1" maxOccurs="1" name="peticionFechaHoraResult" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="informacionDisponible">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element minOccurs="1" maxOccurs="1" name="listaTipoFicheros" type="ree:ListaTipoFicheros" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:complexType name="ListaTipoFicheros">
      <xsd:sequence>
        <xsd:element minOccurs="1" maxOccurs="20" name="TipoFicheros" type="ree:TipoFicheros" />
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="TipoFicheros">
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" name="DatosFichero" type="ree:DatosFichero" />
      </xsd:sequence>
      <xsd:attribute name="id" type="ree:ListaTipos" />
    </xsd:complexType>
    <xsd:complexType name="DatosFichero">
      <xsd:attribute minOccurs="1" maxOccurs="1" name="nombre" type="xsd:string" />
      <xsd:attribute minOccurs="1" maxOccurs="1" name="longitud" type="xsd:int" />
    </xsd:complexType>
    <xsd:simpleType name="ListaTipos">
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="CUR" />
        <xsd:enumeration value="PTE" />
        <xsd:enumeration value="PTF" />
        <xsd:enumeration value="TAR" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:schema>
</types>
```



```
<xsd:enumeration value="INC"/>
<xsd:enumeration value="OBJ"/>
<xsd:enumeration value="ROB"/>
<xsd:enumeration value="FIR"/>
<xsd:enumeration value="AGR"/>
<xsd:enumeration value="MAG"/>
<xsd:enumeration value="INV"/>
<xsd:enumeration value="DIS"/>
<xsd:enumeration value="RIS"/>
<xsd:enumeration value="NOS"/>
<xsd:enumeration value="OSI"/>
<xsd:enumeration value="OSP"/>
<xsd:enumeration value="OSA"/>
<xsd:enumeration value="OSG"/>
<xsd:enumeration value="OSD"/>
<xsd:enumeration value="OSE"/>
<xsd:enumeration value="TOD"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:element name="informacionDisponibleResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="1" maxOccurs="1" name="informacionDisponibleResult"
type="ree:ListaTipoFicheros"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="recogerFichero">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="1" maxOccurs="1" name="nombreFichero" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="recogerFicheroResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="1" name="recogerFicheroResult" type="ree:Fichero"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="Fichero">
  <xsd:sequence>
    <xsd:element minOccurs="1" maxOccurs="1" name="Nombre" type="xsd:string"/>
    <xsd:element minOccurs="1" maxOccurs="1" name="Checksum" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="confirmacionRecepcionFichero">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="1" name="nombreFichero" type="xsd:string"/>
      <xsd:element minOccurs="0" maxOccurs="1" name="codConfirmacion" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="confirmacionRecepcionFicheroResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="1" maxOccurs="1" name="confirmacionRecepcionFicheroResult" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="enviarFichero">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="1" name="fichero" type="ree:Fichero"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```



```
</xsd:complexType>
</xsd:element>
<xsd:element name="enviarFicheroResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="1" maxOccurs="1" name="enviarFicheroResult" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
</types>
<message name="peticionFechaHoraSoapIn">
  <part name="parameters" element="ree:peticionFechaHora"/>
</message>
<message name="peticionFechaHoraSoapOut">
  <part name="parameters" element="ree:peticionFechaHoraResponse"/>
</message>
<message name="informacionDisponibleSoapIn">
  <part name="parameters" element="ree:informacionDisponible"/>
</message>
<message name="informacionDisponibleSoapOut">
  <part name="parameters" element="ree:informacionDisponibleResponse"/>
</message>
<message name="recogerFicheroSoapIn">
  <part name="parameters" element="ree:recogerFichero"/>
</message>
<message name="recogerFicheroSoapOut">
  <part name="param" element="ree:recogerFicheroResponse"/>
  <part name="datos" type="xsd:hexBinary"/>
</message>
<message name="confirmacionRecepcionFicheroSoapIn">
  <part name="parameters" element="ree:confirmacionRecepcionFichero"/>
</message>
<message name="confirmacionRecepcionFicheroSoapOut">
  <part name="parameters" element="ree:confirmacionRecepcionFicheroResponse"/>
</message>
<message name="enviarFicheroSoapIn">
  <part name="param" element="ree:enviarFichero"/>
  <part name="datos" type="xsd:hexBinary"/>
</message>
<message name="enviarFicheroSoapOut">
  <part name="parameters" element="ree:enviarFicheroResponse"/>
</message>
<portType name="Servicio">
  <operation name="peticionFechaHora">
    <input message="ree:peticionFechaHoraSoapIn"/>
    <output message="ree:peticionFechaHoraSoapOut"/>
  </operation>
  <operation name="informacionDisponible">
    <input message="ree:informacionDisponibleSoapIn"/>
    <output message="ree:informacionDisponibleSoapOut"/>
  </operation>
  <operation name="recogerFichero">
    <input message="ree:recogerFicheroSoapIn"/>
    <output message="ree:recogerFicheroSoapOut"/>
  </operation>
  <operation name="confirmacionRecepcionFichero">
    <input message="ree:confirmacionRecepcionFicheroSoapIn"/>
    <output message="ree:confirmacionRecepcionFicheroSoapOut"/>
  </operation>
  <operation name="enviarFichero">
    <input message="ree:enviarFicheroSoapIn"/>
    <output message="ree:enviarFicheroSoapOut"/>
  </operation>
</portType>
<binding name="Servicio" type="ree:Servicio">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
</binding>
</wsdl:binding>
</wsdl:service>
</wsdl:definitions>
</wsdl:serviceDefinition>
</wsdl:definitions>
</wsdl:definitions>
```



```
<operation name="peticionFechaHora">
  <soap:operation soapAction="http://localhost:8102/proyectocpcs/peticionFechaHora" style="document"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
<operation name="informacionDisponible">
  <soap:operation soapAction="http://localhost:8102/proyectocpcs/informacionDisponible" style="document"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
<operation name="recogerFichero">
  <soap:operation soapAction="http://localhost:8102/proyectocpcs/recogerFichero" style="document"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <mime:multipartRelated>
      <mime:part>
        <soap:body parts="param" use="literal"/>
      </mime:part>
      <mime:part>
        <mime:content part="datos" type="application/x-bzip2"/>
      </mime:part>
    </mime:multipartRelated>
  </output>
</operation>
<operation name="confirmacionRecepcionFichero">
  <soap:operation soapAction="http://localhost:8102/proyectocpcs/confirmacionRecepcionFichero" style="document"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
<operation name="enviarFichero">
  <soap:operation soapAction="http://localhost:8102/proyectocpcs/enviarFichero" style="document"/>
  <input>
    <mime:multipartRelated>
      <mime:part>
        <soap:body parts="param" use="literal"/>
      </mime:part>
      <mime:part>
        <mime:content part="datos" type="application/x-bzip2"/>
      </mime:part>
    </mime:multipartRelated>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
</binding>
<wsdl:service xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" name="Servidor">
  <wsdl:port name="Servicio" binding="ree:Servicio">
    <soap:address location="http://localhost:8102/proyectocpcs"/>
  </wsdl:port>
</wsdl:service>
</definitions>
```



CONCLUSIONES

Este proyecto nos ha servido para ampliar nuestros conocimientos de diferentes tecnologías no vistas durante la carrera. También hemos fijado conceptos de Ingeniería del Software que ya aplicamos en esa asignatura.

Destaca sobre el resto de conocimientos adquiridos, la posibilidad de adentrarnos en los sistemas de comunicaciones y de control en entornos profesionales. Ejemplos de los conocimientos adquiridos al respecto son el uso de especificaciones técnicas avanzadas, estándares internacionales, y modelos de desarrollo avanzados. Gracias a nuestro trabajo de este proyecto hemos conocido los métodos de trabajo de empresas de gran entidad como Indra, Red Eléctrica de España y hemos tenido acceso a especificaciones de estándares internacionales, sin las cuales no habríamos podido trabajar ya que son de uso restringido.

Además de los puntos mencionados anteriormente, hemos podido aprender a administrar hardware de alto rendimiento para su posterior configuración como servidor de nuestro proyecto.

Como resumen diremos que el proyecto ha sido muy enriquecedor para nuestra formación e interesante en muchos aspectos técnicos. Si bien hubo momentos difíciles, tenemos que reconocer que llevarlo a buen puerto nos supone una gran satisfacción personal y un final para nuestra formación universitaria.



BIBLIOGRAFÍA

Título: *Elegibilidad '03: Protocolo de Comunicaciones entre concentradores de Medida. Solución basada en "Web Services"*.

Autor: Red Eléctrica de España

Fecha: 20-2-2004

Título: *International Standard IEC-870-5-[1..5]*

Autor: International Electrotechnical Comitee

Fecha: 1- 4-1992

Título: *International Standard IEC-870-5-101: User Conventions revision 2.0*

Autor: International Electrotechnical Comitee

Fecha: 24-3-2003

Título: *International Standard IEC-870-5-102. Link Transmission Procedures*

Autor: International Electrotechnical Comitee

Fecha: 1-06-1996

Título: *Sistemas de Información de Medidas eléctricas: Criterios para el establecimiento de los puntos frontera en instalaciones de producción en régimen especial. Versión 1.0*

Autor: Red Electrica de España

Fecha: 30-11-2004

Título: *Understading Web Services: XML, WSDL, SOAP and UDDI*

Autor: Eric Newcomer

Editorial: Addison-Wesley

Fecha: 2005

Título: *Web Services Essentials*

Autor: Ethan Cerami

Editorial: O'Reilly

Fecha: 2002

Título: *Protocolo CP-CS. Sistema de Información de Medidas eléctricas SIMEL.*

Autor: Indra

Fecha: 10-3-1999

Referencias de Internet consultadas:

<http://www.ree.es>

<http://www.microsoft.com>

<http://java.sun.com>

<http://www.netbeans.org>



ANEXO A. Riesgos del proyecto

Este apartado hace referencia a los riesgos que pueden aparecer tanto al principio como a lo largo del desarrollo, identificando cada tipo de riesgo, su probabilidad, efectos que puedan causar en el proyecto, algunas formas de prevenirlos y cómo actuar en cada caso para solucionarlos.

Riesgos de Software

Tabla 5 - Riesgos Software

RIESGO	TIPO	DESCRIPCIÓN
Integridad de equipo.	Proyecto.	.El personal del equipo puede dejar de lado el proyecto por tener otras asignaturas más prioritarias, por falta de interés en el proyecto, o falta de motivación
Salud del equipo.	Proyecto.	El personal del equipo puede dejar momentáneamente el proyecto por problemas de salud.
problemas con el aprendizaje de nuevas tecnologías.	Proyecto.	Algunos o todos los miembros sean incapaces de aprender las nuevas tecnologías que van a ser usadas en el desarrollo del proyecto.
Dificultades para la reunión	Proyecto.	Debido a que todos los miembros del equipo están trabajando en una beca de horario parcial, el tiempo disponible para reuniones de proyecto está limitado a la intersección de horarios libres.
Descoordinación entre los distintos módulos.	Desarrollo.	Dificultad de integración de los distintos módulos una vez estén estos terminados, capaz de producir cambios (sustanciales o no) en uno o más módulos.

Tipos de riesgos

Tabla 6 - Tipos de Riesgos

TIPO DE RIESGO	POSIBLES RIESGOS
Tecnológicos	- problemas con el aprendizaje de nuevas tecnologías. - Integración de las distintas componentes desarrolladas
Plantilla	- Integridad del equipo. - Salud del equipo. - Dificultades para la reunión
Estimación	- Estimaciones incorrectas en tiempo para el aprendizaje y desarrollo de nuevas tecnologías y conceptos



Organización	- problemas software. - Descoordinación entre los distintos módulos
Requisitos	- Cambio en los requisitos de la especificación.

Análisis de riesgos

Tabla 7 - Análisis de riesgos

RIESGO	PROBABILIDAD	EFFECTOS
Integridad de equipo.	Baja	Tolerable
Salud del equipo	Alta	Tolerable
Dificultades para la reunión de todo el grupo.	Muy alta	Tolerable
problemas con el aprendizaje de nuevas tecnologías.	Muy baja	Tolerable
Descoordinación entre los distintos módulos.	Alta	Tolerable



Estrategias de gestión de riesgos

Tabla 8 - Estrategias de Gestión de Riesgos

RIESGO	ESTRATEGIA
Integridad de equipo.	<ul style="list-style-type: none">- Plan de prevención: Motivar al personal para que siga trabajando el proyecto, exponiendo las ventajas, puntos fuertes, eliminando el exceso de presión.- Plan de Solución: Redistribución de las funciones del personal, con su consecuente aumento de trabajo para cada miembro.
Salud del equipo.	<ul style="list-style-type: none">- Plan de prevención: No es un riesgo que se pueda prevenir, pero si se pueden tomar medidas previas para minimizar su efecto, como puede ser el no sobrecargar de trabajo a cada miembro.- Plan de solución: Asignación temporal del trabajo a otro miembro.
Dificultades para la reunión de todo el grupo.	<ul style="list-style-type: none">- Plan de prevención: Fomentar la comunicación por otros medios, como correo electrónico.- Plan de solución: Reducción de las reuniones a únicamente las indispensables y que esas sean de obligada asistencia.
problemas con el aprendizaje de nuevas tecnologías.	<ul style="list-style-type: none">- Plan de prevención: Facilitar los manuales necesarios y tutoriales para el aprendizaje de las nuevas tecnologías de manera fácil y sencilla.- Plan de solución: Traspasar a ese miembro a otra grupo de desarrollo del proyecto y sustituirlo por otro miembro conocedor de la tecnología o capaz de aprenderla.
Descoordinación entre los distintos módulos.	<ul style="list-style-type: none">- Plan de prevención: Esquematizar el desarrollo en términos generales desde un principio, de modo que los desarrolladores tengan información suficiente sobre todos los módulos y de ese modo hacer sus desarrollos compatibles.- Plan de solución: Modificación de los módulos afectados para hacer posible la integración o rediseñar el módulo de conflicto para que se pueda integrar con los demás módulos.



Anexo B. Palabras Clave.

SECCION I

Registrador de medidas.

Concentrador de medidas.

ASDU (Application Service Data Unit)

Puerto Serie.

Módem GSM.

SECCION II

Concentrador Primario.

Concentrador Secundario.

WebService.

SSL.

Attachments.