

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

INGENIERO SUPERIOR DE TELECOMUNICACION



**CAPTURA, ANÁLISIS Y OPTIMIZACIÓN
DEL CONSUMO ELÉCTRICO OBTENIDO
DE CONTADORES DE LA RED ELÉC-
TRICA MEDIANTE MUESTREO CON
MICROPIC.**

PROYECTO FIN DE CARRERA

Marzo – 2015

AUTOR: Antonio Sirvent Santos

DIRECTOR/ES: Miguel Onofre Martínez Rach
Alonso Ivars Nicolás

Dedicado a mi sufridora madre y a mi paciente padre.

VISTO BUENO Y CALIFICACIÓN DEL PROYECTO

Título proyecto:

CAPTURA, ANALISIS Y OPTIMIZACION DEL CONSUMO ELECTRICO OBTENIDO
MEDIANTE MUESTREO CON MICROPIC DE CONTADORES DE LA RED
ELECTRICA.

Proyectante: Antonio Sirvent Santos

Director/es: Miguel Onofre Martínez Rach

Alonso Ivars Nicolás

VºBº director/es del proyecto:

Fdo.:

Fdo.:

Lugar y fecha:

CALIFICACIÓN NUMÉRICA

MATRÍCULA DE HONOR

Méritos justificativos en el caso de conceder Matrícula de Honor:

Conforme presidente:

Fdo.:

Conforme secretario:

Fdo.:

Conforme vocal:

Fdo.:

Lugar y fecha:

Agradecimientos

Agradezco la paciencia de mi familia y tutores por llevar este proyecto a cabo, han sido muchos años de discontinuidad debido a la finalización de mis asignaturas y el comienzo de mi vida laboral que me permitía poco tiempo libre para la finalización de este proyecto y al final ha tenido que ser una operación la que me haya dado la posibilidad de rematarlo, gracias a todos por vuestra paciencia.

Agradecer también la ayuda de José Juan que tanto me ha ayudado con el Java a Alonso con todo el tema del protocolo y a Carlos AC's con la placa y las memorias.

Agradecer también a Miguel Onofre mi tutor toda la paciencia que ha tenido conmigo desde que iniciáramos la primera versión y todo lo que he aprendido con él, gracias por la confianza que he adquirido en todo lo que me has enseñado.

Mis compañeros de universidad, con quienes viví muchos buenos momentos (aprobados, deporte, salidas) y otros no tan buenos (suspensos, jornadas interminables de estudio, malos ratos) pero siempre con buen humor.

A mis amigos, que llevan toda la vida soportándome (y lo que les queda...).

Agradecer a mi novia la paciencia, constancia y ayuda que me ha dado para llevar esto a acabo, han sido momentos muy difíciles, duros y con falta de motivación en muchos momentos y siempre ha estado ahí para darme ánimos y centrarme cuando yo no era capaz.

Resumen

El presente PFC está basado en un requerimiento de la empresa AYF Energía: la necesidad de poder muestrear varias veces a lo largo de un minuto las diferentes variables físicas relevantes en un suministro eléctrico. Éstas son las tensiones e intensidades de fase, el desfase de las mismas y las potencias activas y reactivas.

La finalidad de la empresa es que todos estos datos se vuelquen en una base de datos para posteriormente poder manejarlos y facilitar la toma de decisiones. Todo el proceso ha de estar manejado de una forma sencilla desde una aplicación que, a día de hoy por seguridad y facilidad, se quiere en el host de la empresa pero se espera que en un futuro se pueda llevar desde una aplicación web.

Se pretende pues, mediante un sistema conectado a un contador eléctrico de compañía distribuidora, convertirlo en un analizador de redes para así tener un muestreo continuo de las variables físicas del consumo energético, almacenarlo en un soporte físico para que posteriormente estos datos sean procesados por la aplicación mencionada para poder tomar las conclusiones necesarias y en un futuro aplicarle una lógica de IA que le permita la toma de decisiones.

ÍNDICE

1	INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS	10
1.1	Introducción	10
1.2	Motivación.....	12
1.3	Objetivos	16
1.3.1	Estudio, selección del PIC y memoria apropiados. Fase 0	17
1.3.2	Estudio del protocolo de comunicaciones de red eléctrica. Fase 0.....	17
1.3.3	Extracción de los datos del contador	17
1.3.4	Diseño de la base de datos. Fase 2	19
1.3.5	Desarrollo de la aplicación de gestión de los datos de clientes. Fase 2..	19
2	ESTRUCTURA DEL PROYECTO	21
2.1	introducción y Estructuración	21
2.2	Parte desarrollo hardware	22
2.3	Parte desarrollo software	22
3	FASES DEL DESARROLLO HARDWARE	25
3.1	Estudio comparativo de los diferentes PICs y memorias	25
3.1.1	MICROCONTROLADOR.....	25
3.1.2	MEMORIA	31
3.1.3	PUERTO RS232	34
3.2	Implementación del circuito electrónico y programación del PIC.....	35
3.3	Estudio del protocolo de comunicación que permita el análisis del consumo. 36	
4	FASES DEL DESARROLLO SOFTWARE.....	41
4.1	Diseño de la aplicación Excel de carga de los ficheros.....	41
4.1.1	Introducción al protocolo de comunicaciones ASDU con lecturas instantáneas.....	41

4.1.2	Documentación.....	46
4.1.3	Tipos de Trama.....	47
4.1.4	Parámetros.....	47
4.1.5	El problema del tiempo.....	48
4.1.6	Conversión de los datos del protocolo	49
4.2	Diseño de La base de datos.....	49
4.2.1	Introducción	49
4.2.2	Elección de la base de datos	50
4.2.3	Elección de las variables y los tipos de dato	50
4.2.4	Elección de las tablas siguiendo las formas normales	52
4.2.5	Modelo de la base de datos.....	53
4.2.6	SQL que contiene la BBDD.....	57
4.3	Diseño de La aplicación.....	57
4.3.1	Navegación de la aplicación.....	57
4.3.2	Esquemas jerárquicos de los módulos	67
4.3.3	Clase AplicacionDao.....	68
4.3.4	Clase de generación de Excel	72
4.3.5	Clase NasaWorldWind	75
5	RESULTADOS EN UN CLIENTE.....	80
5.1	Datos Previos del cliente	80
5.2	Aplicación de nuestro sistema	81
5.3	Tarifa de mano de obra	83
5.4	Presupuesto y viabilidad del pfc.....	83
6	CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS.....	85
6.1	Conclusiones	85

6.2	Mejoras de Software.....	85
6.2.1	Mejoras de procesos.....	85
6.2.2	Mejoras de la BBDD.....	86
6.2.3	Mejora de los mapas.....	87
6.3	Mejoras de hardware.....	88
6.3.1	Medidas remotas mediante un modem GSM/4g.....	88
6.3.2	Reconocimiento de periodos tarifarios 0.....	88
7	ANEXO - REFERENCIAS BIBLIOGRÁFICAS.....	90
7.1	Glosario.....	90
7.2	Protocolo de comunicaciones ASDU.....	95
7.2.1	Normativa.....	95
7.2.2	Protocolo de comunicaciones.....	96
7.2.3	Estructura de los datos de aplicación (ASDU).....	102
7.3	Formas normales.....	117
7.4	Definición de la BBDD.....	120
7.5	Datos previos a la ejecución y análisis del cliente.....	124
7.6	Datos posteriores al análisis de los datos registrados.....	129
7.7	Índice y referencia de las figuras.....	133
7.8	Referencias.....	135

1 INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS

1.1 INTRODUCCIÓN

El objetivo de este proyecto fin de carrera “*CAPTURA, ANÁLISIS Y OPTIMIZACIÓN DEL CONSUMO ELÉCTRICO OBTENIDO MEDIANTE MUESTREO CON MICROPIC DE CONTADORES DE LA RED ELÉCTRICA*” es la realización de un sistema de medición y almacenamiento para la empresa AYF Energía, para ello es necesario tomar las medidas eléctricas de los contadores de red de sus clientes para así, tras el posterior análisis de estos datos, poder optimizar su rendimiento eléctrico mediante el registro y análisis de estos datos.

Se ha estructurado en diferentes fases el desarrollo de la aplicación:

El punto de partida para el desarrollo de este PFC consistió en la regulación a nivel nacional del protocolo IEC(La Comisión Electrotécnica Internacional) 870-5-102 para la conexión con el contador eléctrico homologado por las compañías distribuidoras mediante el puerto RS-232 integrado en el mismo. Para ello la empresa valora diferentes alternativas que desea desarrollar,

- Desarrollo Hardware para la extracción y volcado de las tramas a un soporte físico.
- Desarrollo de software de conexión vía pc para la extracción de la información y almacenarla en un fichero plano,

El objetivo por parte de la empresa sería la necesidad de poder garantizar la extracción de estos datos de la manera más rápida posible para poder salir al mercado, ambas opciones se exploraron y desarrollaron en paralelo para garantizar su obtención. Finalmente solo una es la que se cerró completamente en este PFC, la otra continúa actualmente en desarrollo y está expandiendo su alcance una vez probada en clientes reales en los cuales se han podido detectar mejoras/carencias que han de ser tratadas.

El siguiente paso consistió en el desarrollo de una aplicación para gestionar correctamente los datos obtenidos. Esta aplicación estará desarrollada en Java y nos dará la posibilidad de visualizar, organizar e insertar dicha información en un HOST (anfitrión) que tendrá una base de datos MySQL diseñada por nosotros.

Esta BBDD (Base de datos) será la base para futuras líneas de trabajo y optimización de las instalaciones de los clientes. También mediante estimaciones a futuro del consumo, se podrían conseguir importantes rebajas en el consumo de la luz trabajando con el mercado de subasta diario y organizando convenientemente los procesos para poder sacar el máximo partido a dichos datos y poder optimizar los suministros energéticos estudiados.

Básicamente el diseño de los requisitos anteriormente comentados con AYF Energía y que se ampliarán en el transcurso del documento fueron extraídos y analizados de manera que se pudieran separar en fases independientes de diseño con sus propios hitos y posibilidad de tener líneas de trabajo en paralelo, cosa que fue importante ya que nos permitió poder trabajar de manera alternativa en diversos puntos sin vernos impactados por retrasos que se podían dar por razones ajenas a nosotros y nuestro desempeño. Estas fases lógicas se consideraron independientes y son básicamente las que nos determina cualquier modelo de capas,

- Preparación y estudio del nivel físico.
- Tratamiento, gestión y almacenamiento de la información, elección del formato con el soporte adecuado.
- Desarrollo de una aplicación de usuario con la que se pueda trabajar e interactuar fácilmente en un pc convencional para tratar como servidor.

Los niveles en los que se ha estructurado serían los siguientes, obviando la primera fase de análisis de los requisitos:

- **FASE 0:** Esta primera fase consiste en la elección y caracterización del medio físico, es decir, el cableado empleado, los componentes necesarios de hardware y el estudio del protocolo de transmisión.

- **FASE 1:** Trata el desarrollo del firmware de comunicaciones, todo lo necesario para que tanto el maestro como los esclavos puedan generar tramas de comunicación procesables y entendibles por el otro extremo de la comunicación, desarrollo del procesador para el repertorio de instrucciones propio, que permitirá configurar el sistema con las prestaciones y niveles que se desee para luego almacenarlos en el soporte físico y poder comenzar la fase 2.
- **FASE 2:** Desarrollo de la aplicación Java, definición de la BBDD y programación visual de la aplicación. Esta es la parte más completa del desarrollo y en la que ha incidido más el objetivo dentro de este PFC.

1.2 MOTIVACIÓN

La necesidad de ser competitivo e innovador en este cambiante panorama eléctrico nacional y específicamente dentro del actual mercado eléctrico lleva a la empresa AYF a adaptarse a las nuevas medidas adoptadas para el registro de los datos de los clientes de la red eléctrica nacional.

Para ello se toma la decisión de comenzar a trabajar con los contadores para potencias mayores a 15kW e investigar alrededor de su protocolo de comunicaciones junto a la posibilidad de tomar unas medidas de las que normalmente no se disponen y que pueden ser muy útiles y rentables de cara la subasta de la energía eléctrica que se produce diariamente.

Para poder optimizar el proceso de compra de la energía eléctrica, tanto en el mercado diario como en el intradiario, hemos de tener claras las necesidades energéticas diarias y horarias para cada suministro (contadores de cliente) para que esto pueda suponer un beneficio neto importante, y una inestimable ayuda para ello es conocer un detalle de lo ocurrido con anterioridad.

El primer requerimiento es pues, determinar cuáles son los datos que estos contadores nos permiten obtener. Entre la multitud de posibilidades (que se detallarán en sucesivos apartados), la más interesante es conocer los valores instantáneos del suministro, a saber, tensiones e intensidades de fase, sus desfases,

sus potencias activas y reactivas, así como las energías totales acumuladas en cada instante.

Lo que se pretende es tener varias medidas por minuto para poder detectar picos instantáneos de tensión/intensidad y así poder corregir desvíos de ésta al encender máquinas o procesos de un determinado cliente, poder analizar cuando se repiten estos “saltos”, y poder extraer nuevas conclusiones gracias a tener valores desagregados (cada ciertos segundos) en vez de tener los valores agregados (cada 15 ó 60 minutos) como ofrece actualmente el contador promediados según los estándares marcados por red eléctrica.

El propósito principal de este proyecto es facilitar un sistema de registro de estas medidas para poder comenzar a trabajar con ellas una vez organizadas por cliente y punto de suministro. Se presentarán los datos ordenados por fecha y hora de registro y con un registro suficientemente desagregado que permita analizar y extraer conclusiones con un alto grado de certeza.

La gran cantidad de datos almacenados, incluyendo variables típicas del funcionamiento normal del suministro (parámetros de producción supuesto el caso de una industria) o climáticos, hará necesaria la ampliación del sistema desarrollado en el presente proyecto, para poder agilizar y automatizar la toma de decisiones con nuevas pautas marcadas gracias a la experiencia adquirida y conocimiento detallado del consumo eléctrico.

Se puede afirmar pues que éste es el primer paso para sentar las bases de un sistema capaz de predecir los consumos horarios y posibles fluctuaciones de los suministros eléctricos.

Otra motivación importante del presente proyecto es la posibilidad de poder disponer de un analizador de redes eléctricas de bajo coste, alta fiabilidad y fácil acceso.

El bajo coste viene motivado por el hecho que los componentes que incrementan el valor de los analizadores de redes tradicionales son los que miden las variables físicas. Al estar estos componentes integrados en los contadores actuales, sólo se tienen que leer los datos, lo que abarata en gran medida la posibilidad de disponer de

un analizador de redes. Por tanto nuestro proyecto será en si un analizador de redes eléctricas en sí mismo.

Respecto a la fiabilidad, dado que los contadores han de estar homologados y verificados por la compañía distribuidora, puede asegurarse que los valores ofrecidos van estar en consonancia con todos los valores y variables necesarias para una perfecta optimización del suministro y de la compra de su energía.

Por último, el contador siempre se encuentra en puntos de fácil acceso. Se suele dar el caso de que para poder colocar las pinzas amperimétricas se ha de desmontar parte de un cuadro, problema que con esta solución se evita. Por otra parte, en los suministros de alta tensión, parte de los datos que ofrece el contador son el alta tensión, y el resto son fácilmente derivables del resto, por lo que se evita tener que operar en alta tensión.

Los modelos de contadores de Red Eléctrica son variados y diferentes según fabricante, pero en los clientes que tenemos podemos acceder a esta información de con el mismo protocolo que es lo interesante para poder hacer funcionar un software estándar.



Figura 1. Ejemplo de contadores de energía de diferentes fabricantes (de izquierda a derecha: Landis&Gyr, CIRCUTOR, Actaris, ZIV, Orbis) utilizados habitualmente en centrales FV

El futuro diseño que se pretende utilizar para tomar las tele medidas sería el que se en las imágenes siguientes de la figura 2 Modem GSM(Global System for Mobile communications) con convertidor RS232 para lecturas remotas y convertidor USB para lecturas con ordenador local., de esta manera se podría tener la información volcada en tiempo real sin necesidad de acudir al cliente, con ello podríamos tener conclusiones *just in time* que serían una ventaja estratégica para la empresa y un valor añadido para destacar sobre la competencia en este mercado que está en plena ebullición.



Figura 2. Contador de energía con convertidor RS232.



Figura 3. Modem GSM con convertidor RS232 para lecturas remotas y convertidor USB para lecturas con ordenador local.

Los valores básicos que recogemos con nuestro sistema son los básicos de tensión, intensidad y potencia en diferentes fases que registran los contadores. Hemos añadido a los valores básicos unos cuantos valores más para poder registrar un máximo de valores razonable que permita un equilibrio entre la cantidad de información que puede registrarse y la memoria disponible en la placa.

1.3 OBJETIVOS

En los siguientes puntos comentaremos los objetivos básicos que hemos ido fijando según la planificación realizada inicialmente.

Las fases comentadas en la introducción en el punto 1.1 y que corresponderían con la Fase 0, 1 y 2 del modelo, se detallan separadamente en este apartado.

1.3.1 Estudio, selección del PIC y memoria apropiados. Fase 0

Se tiene como objetivo realizar un estudio de los posibles micros disponibles en la industria para realizar lecturas secuenciales sobre el contador de red eléctrica, tomando las lecturas instantáneas de éste y volcándolas en un soporte físico.

1.3.2 Estudio del protocolo de comunicaciones de red eléctrica. Fase 0.

Se debe de estudiar toda la información referente al protocolo común que ofrecerán todos los contadores y puntos de suministro eléctrico. Haremos un estudio para ver cómo conseguir esta información instantánea para su extracción y procesado posterior.

1.3.3 Extracción de los datos del contador

En este punto se consideraron dos vertientes paralelas con el objetivo de conseguir las primeras tramas, La empresa necesita tener dos maneras con las que poder contar para realizar esta labor, esto es primordial para confirmar la importación de tramas y verificar el formato que estas tienen para continuar con el desarrollo.

- **Diseño de la placa de control que incluye el PIC (Peripheral Interface Controller), memoria y el conversor PIC/RS232 . Fase 0 -1**

El requisito sería diseñar un modelo sencillo para la lectura del contador y que veremos en el punto 3.1. Básicamente se ha de conectar el PIC y configurarlo para poder conectarlo con el conversor de señal para que se pueda conectar con el contador, una vez establecida la conexión el código del MicroPic tiene un bucle de consulta al contador de valores instantáneos

Hemos estimado que haciendo lecturas cada 15-25 segundos y la información que almacenamos por trama es de 330 Bytes, hacemos una estimación al alza de 1Kbyte por minuto, 60 Kbyte hora que nos da aproximadamente 1,5 Megabyte al día de volcado de información. Con lo cual con una USB de unas 5 Gigabytes tendríamos espacio suficiente para el volcado de la información. Como hemos comentado este requisito está actualmente en fase de pruebas y no está plenamente optimizado.

- **Extracción de las variables de análisis del consumo para su almacenamiento por tramas en un fichero. Fase 1**

Los objetivos en este punto son estudiar el protocolo y desarrollar un programa Java que nos permita conseguir tramas en hexadecimal como desarrollo alternativo al punto anterior.

Ha de realizarse mediante un código que se pueda ejecutar desde cualquier pc sobremesa común, al cual introduciendo unos determinados parámetros básicos de conexión podamos extraer la información de las lecturas instantáneas en hexadecimal.

1.3.4 Diseño de la base de datos. Fase 2

El objetivo de la base de datos es cubrir la necesidad de volcar de manera fácilmente reconocible los datos que provienen de una lectura instantánea del contador. Estos datos hexadecimales deben ser representados en una Excel y para ello deben de ser almacenados en un formato decimal.

Los datos deben de estar organizados en la tabla para agilizar las futuras consultas que se realicen, se pretende tener un diseño de la BBDD bastante normalizado ya que no vamos a tener problemas de concurrencias.

El diseño de ésta debe recoger los requisitos solicitados por el cliente y necesarios para poder definir un análisis consistente. Hay que definir cuáles deberían ser los datos almacenados y con qué frecuencia, teniendo en cuenta que estos datos se registrarán por cada contador y estos deben de agruparse para un único cliente.

1.3.5 Desarrollo de la aplicación de gestión de los datos de clientes. Fase 2

El objetivo de este apartado sería poder integrar dentro de una aplicación la posibilidad de cargar los datos que hemos leído del contador y que tenemos almacenados en un soporte físico mediante un fichero de texto, desde este formato debemos de ser capaces de poder cargar esta información en nuestra base de datos a través de la aplicación, la cual debe de cumplir ciertos requisitos necesarios para la empresa.

Los objetivos de nuestro desarrollo mediante la aplicación son,

- El de poder insertar las tramas registradas en nuestra BBDD con un formato generado desde la lectura directa del contador de la red y posicionarla dentro del ámbito de un cliente.
- El de poder consultar las tramas que insertamos en nuestra base de datos y volcarla en una hoja de cálculo con un formato concreto.
- Debe de ser posible mostrar la ubicación en un punto concreto del mapa la ubicación del contador, mediante geolocalización.
- Debe de poder registrar la información de distintos clientes con diferentes contadores en diferentes ubicaciones.
- Deben de poder registrarse las medias de consumo, tanto para un cliente con todos sus contadores, como para un contador concreto.
- Repositorio de clientes
- Gestión de la información básica de los clientes
- Consulta de los datos, tanto por contador como por cliente
- Cálculo de medias de consumo de contador por cliente
- Consulta de totales del cliente-contador con filtro entre fechas.
- La extracción y manejo de toda la información ha de ser mediante un formato xls de hoja de cálculo.

Con estos requisitos contemplaríamos de manera resumida el alcance de este PFC a nivel aplicación.

2 ESTRUCTURA DEL PROYECTO

Los siguientes capítulos nos hablarán un poco de los protocolos y requisitos que tenemos con Red Eléctrica para la adquisición de estos datos, haciendo hincapié en las diferentes versiones de la norma “IEC 870-5-102” anteriormente comentada. En el primer Anexo 7.1 podremos consultar el detalle.

Una vez conocidos estos detalles y realizado el análisis, hablaremos de la parte del desarrollo de Hardware con los objetivos definidos por parte de la empresa que hemos podido ver en el capítulo anterior, lo conseguido y los puntos a continuar en líneas futuras de desarrollos que podremos desarrollar en el apartado 6.1 .

La parte final del documento, contendrá toda la información referente al desarrollo de cada una de las fases en las que se ha dividido éste a nivel de Software, contemplando todo el procesado de datos, el posterior almacenamiento de la información en bases de datos, así como el diseño del interfaz de la aplicación. Explicaremos mediante su flujo la navegación y utilidades desde un punto de vista funcional.

2.1 INTRODUCCIÓN Y ESTRUCTURACIÓN

Como hemos visto en los capítulos introductorios de este documento podemos dividir el trabajo realizado en este PFC en tres Fases, la primera que sería la Fase 0 sería una primera fase de análisis del protocolo y de los requisitos que tenemos por parte de la empresa.

Una vez visto el alcance y teniendo claros los módulos que vamos a desarrollar empezamos con el nivel Físico que sería la Fase 1 que hemos comentado en el punto 1.3 anterior en el que se detallan los objetivos que debíamos alcanzar junto con la empresa para poder llevar a cabo la aplicación, una vez finalizadas estas dos

primeras fases y con las tramas ya conseguidas en soporte físico pudimos comenzar la parte de desarrollo Software de la Fase 2.

Este estudio del protocolo y las indicaciones necesarias para conseguir los datos de trama con sus formatos correspondientes pueden resultar un poco pesado, como hemos comentado haremos referencia al primer anexo si se desea hacer un estudio de este tema.

2.2 PARTE DESARROLLO HARDWARE

En este punto comentaremos las partes en las que dividimos nuestro desarrollo hardware para la elaboración de este PFC.

Básicamente nos centraremos en 3 conceptos,

- Estudio de los dispositivos que podremos necesitar y sus características.
- Desarrollo del mismo y su programación.
- Obtención de la trama.

Con estos puntos dejaremos descritos los pasos en los que trabajamos en este PFC y mostraremos el resultado de los Objetivos que comentamos en el punto anterior.

En el anexo (apartado 7) y en las conclusiones para líneas de trabajo futuras (apartado 6) podremos ver más en detalle alguno de los puntos que se exponen en el capítulo que nos habla del desarrollo del Hardware.

2.3 PARTE DESARROLLO SOFTWARE

En las posteriores secciones vamos a comentar las partes en las que dividimos nuestro desarrollo software.

El sistema requerido es una aplicación Java que se utilizara desde un único Pc Host, la idea es que sea escalable con el tiempo para poder convertirse en una aplicación que se pueda utilizar desde internet para que cuando algún trabajador esté en un sitio determinado con los datos pueda cargarlos para que así otros usuarios puedan acceder a esta misma información. Con esto debíamos definir también una BBDD para poder almacenarla y consultarla desde la aplicación, con todo esto se decidió utilizar un proyecto Java con Eclipse y una base de datos de Mysql que tiene buena conexión con el lenguaje Java mediante conectores específicos como el JDBC.

La aplicación ha de ser capaz de atender a todos los requisitos de consultas y cargas de los datos recibidos, aparte de requisitos que han ido surgiendo como la necesidad de que los contadores estén marcados de manera visual en un mapa con vista de satélite, para que contuviera el punto exacto en el que los técnicos lo pudieran ver, ya que ciertas direcciones de fábricas/almacenes no tienen fácil localización. Otro requisito era que no referenciara a google earth ni a ningún otro proveedor externo, con lo cual tuvimos que encontrar unas librerías como las NasaWind que nos permitían hacer lo requerido y con buen rendimiento.

El desarrollo se ha hecho de manera muy escalonada a nivel software, con ello hemos ido estableciendo los requisitos del sistema a realizar. El primer paso se dio con la primera trama conseguida con la parte hardware y ello llevó al desarrollo de una aplicación que con un amplio conocimiento del ASDU (Application Service Data Unit) nos permitiera gracias a las librerías POI la extracción de esos datos y su visualización en una hoja de cálculo con valores perfectamente entendibles.

Una vez realizado el proceso anterior y cercado las variables necesarias se vio la necesidad de guardar toda la información en una BBDD en vez de en una Excel para que fuera mucho más manejable y ayudara al análisis o a la toma de decisiones, esto nos hizo ver que deberíamos guardar la información de varios clientes y que estos a su vez podrían tener varios contadores cada uno, cosa que nos llevó a diseñar esta misma según los requerimientos que consideramos para nuestra aplicación.

Una vez estuvo definido y creado todo lo anterior proseguimos con el tema de la creación de la aplicación, la parte visual y la parte necesaria para la ejecución con la base de datos de inserciones, consultas y listados. Una vez tuvimos todo esto desarrollado se nos comentó la necesidad de poder localizar el contador con una

aplicación que nos mostrara su ubicación exacta en un mapa según unas coordenadas, ya que en algunos casos la posición del contador estaba ubicada en un poste de luz y no se localizaba fácilmente con los mecanismos habituales.

Esta aplicación es un punto de partida para el manejo de información de cliente de la empresa y creemos que va a resultar de gran utilidad para la toma de decisiones en todo lo relativo a mejoras energéticas en un cliente y también para lo relativo a la previsión de consumo del mercado eléctrico, todo esto junto a un análisis profundo en función del cliente pensamos que puede dar un trato diferencial con sus clientes en un futuro a muy corto plazo.

3 FASES DEL DESARROLLO HARDWARE

3.1 ESTUDIO COMPARATIVO DE LOS DIFERENTES PICS Y MEMORIAS

Llegados a este punto del proyecto, y habiendo definido previamente los objetivos a conseguir, llega el momento de seleccionar la tecnología a emplear para llevar a cabo la implementación del dispositivo.

A la hora de esta selección, se tuvieron en cuenta los siguientes criterios de diseño:

- Fiabilidad contrastada de los componentes electrónicos a utilizar
- Bajo tiempo de desarrollo del sistema (reducido “time to market”)
- Bajo coste de los componentes
- Existencia de suficientes puertos de comunicaciones

Dadas las condiciones antes mencionadas, se optó por trabajar con micro controladores frente a otras tecnologías como FPGA's(Field Programmable Gate Arrays) dado que para el proyecto no es necesario disponer de hardware reconfigurable, sino que se realizará una PCB(printed circuit board) a medida.

3.1.1 MICROCONTROLADOR

Dentro de los microcontroladores se estudiaron detenidamente las siguientes familias:

- PIC de Microchip
- PSCoC de Cypress
- Microcontroladores Atmel

- Texas Instruments

En base a los criterios antes mencionados se optó por la tecnología Microchip dado que cumple con los siguientes:

- Tecnología ampliamente contrastada y madura
- Experiencias satisfactorias previas de trabajo con los microcontroladores PIC
- Amplia comunidad de desarrolladores que permitirá ayudar a resolver problemas de diseño tanto Hardware como Firmware, reduciendo el *time to market*
- Bajo coste

Una vez seleccionada la tecnología PIC de Microchip, se procederá a elegir el modelo concreto de microcontrolador.

Tras un examen minucioso del catálogo de microcontroladores PIC, se decide emplear el modelo 18F2550 que dispone de los puertos de comunicaciones necesarios para implementar las comunicaciones requeridas.



Figura 4. Imagen de nuestro micro controlador

El modelo tiene un diagrama de pines acorde al siguiente esquema, extraído del Datasheet del mismo.

En el esquema pueden apreciarse los distintos puertos de entrada/salida así como otros pines de interés para la aplicación:

28-Pin PDIP, SOIC

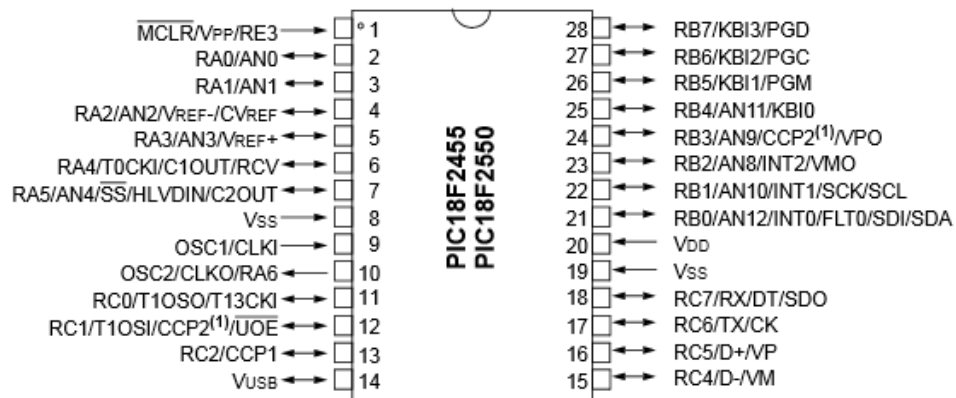


Figura 5. Puertos de nuestro PIC.

A continuación se muestra información de las características técnicas del micro:

Features

- Full Speed USB 2.0 (12Mbit/s) interface
 - 1K byte Dual Port RAM + 1K byte GP RAM
 - Full Speed Transceiver
 - 16 Endpoints (IN/OUT)
 - Internal Pull Up resistors (D+/D-)
 - 48 MHz performance (12 MIPS)
 - Pin-to-pin compatible with PIC16C7X5

Figura 6. Características del micro controlador.

Este modelo es ideal para aplicaciones de bajo consumo y conectividad gracias a disponer de tres puertos serie, FS-USB (12 Mbit/s), I²C, y SPI. Además dispone de un puerto serie asíncrono, (EUSART)

Este tipo de microcontrolador se ajusta perfectamente para el desarrollo de control embebido y aplicaciones de monitorización que requieren conexión periódica con un ordenador.

Parameter Name	Value
Program Memory Type	Flash
Program Memory (KB)	32
CPU Speed (MIPS)	12
RAM Bytes	2,048
Data EEPROM (bytes)	256
Digital Communication Peripherals	1-UART, 1-A/E/USART, 1-SPI, 1-I2C1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	2 CCP
Timers	1 x 8-bit, 3 x 16-bit
ADC	10 ch, 10-bit
Comparators	2
USB (ch, speed, compliance)	1, Full Speed, USB 2.0
Temperature Range (C)	-40 to 85
Operating Voltage Range (V)	2 to 5.5
Pin Count	28

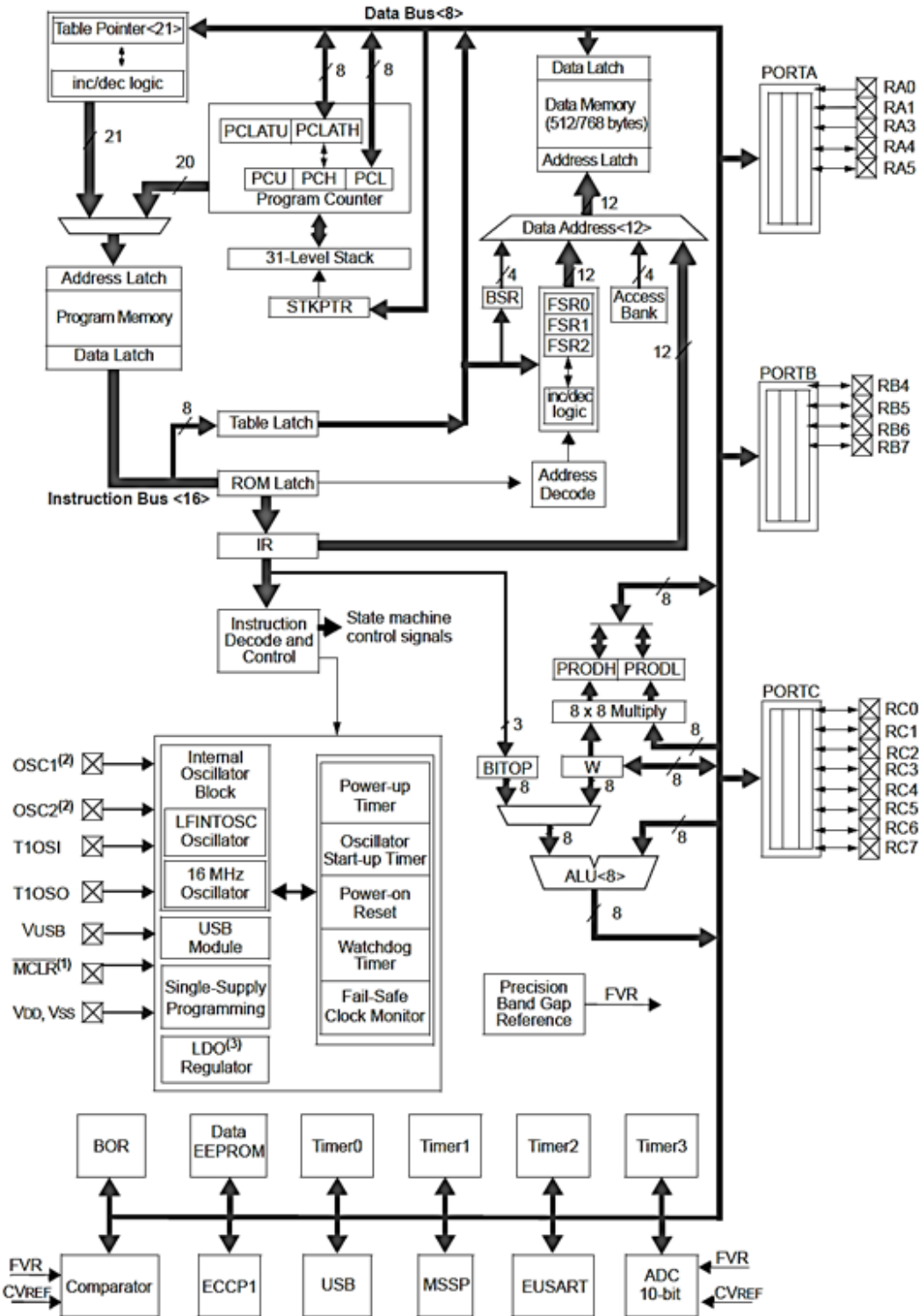
Figura 7. parámetros del microcontrolador.

El voltaje de operación cubre el valor de diseño que se requiere, 5 voltios, además el procesador permite operar hasta una tasa de 48MHz.

Por último, cabe destacar su bajo precio, cercano a los 4 dólares por unidad con posibilidad de escalado para un mayor número de unidades.

Como memorias, el 18F2550 viene con una gran cantidad de memoria RAM (random-access memory) (2048 bytes) y una memoria EEPROM (Electrically Erasable Programmable Read-Only Memory) de 256 bytes.

Por último se muestra el diagrama de bloques del dispositivo extraído también de la hoja de características técnicas facilitado por el propio fabricante Microchip:



Note 1: RA3 is only available when $\overline{\text{MCLR}}$ functionality is disabled.

Note 2: OSC1/CLKIN and OSC2/CLKOUT are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to Section 2.0 "Oscillator Module" for additional information.

Note 3: PIC18F13K50/PIC18F14K50 only.

Figura 8. Diagrama de bloques del dispositivo.

3.1.2 MEMORIA

Además será necesaria una memoria externa al microcontrolador para poder almacenar los datos registrados dado que pueden llegar a ser numerosos.

Esta memoria, por motivos de sencillez de integración se ha elegido también del fabricante Microchip. El modelo seleccionado es el 24LC256.

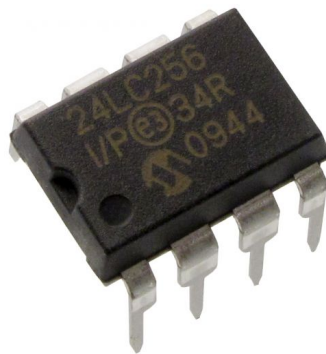


Figura 9. Memoria 24LC256

Esta memoria dispone de una capacidad de 256Kbits, con una frecuencia máxima de reloj de 400 kHz:

Parameter Name	Value
Density	256K bits (x8)
Op. Volt Range (V)	2.5 to 5.5
Max. Clock Freq.	400 kHz
Page Size (bytes)	64
Write Protect	Full Array
Temp Range (°C)	-40°C to +125°C
Endurance	1,000,000
Data Retention (Years)	200

Figura 10. Parámetros de la memoria 24LC256

Algunas de las características más destacadas de la memoria pueden contemplarse en las tablas adjuntas:

Features
<ul style="list-style-type: none">■ Reliable EEPROM Memory<ul style="list-style-type: none">□ 32K x 8 (256Kbit)□ Self-Timed Erase/Write Cycle□ 64-Byte Page Write Buffer□ Page Write Time 5 ms Max.□ Hardware Write-Protect Pin□ Factory Programming Available■ Low Power<ul style="list-style-type: none">□ Operating voltage 2.5V to 5.5V□ Read current 400 uA, max.□ Standby current 1 uA, max.■ 2-Wire Serial Interface, I²C™ Compatible<ul style="list-style-type: none">□ Cascadable up to Eight Devices□ Schmitt Trigger Inputs for Noise Suppression□ Output Slope Control to Eliminate Ground Bounce□ 100 kHz and 400 kHz Clock Compatible■ ESD Protection >4000V■ Pb-Free and RoHS Compliant

Figura 11. Características de nuestra memoria 24LC256

Este modelo de memoria es capaz de ser operada a los 5 voltios a los que se desea trabajar. Está diseñada específicamente para aplicaciones de adquisición de datos o comunicaciones. Su diagrama de bloques es el siguiente:

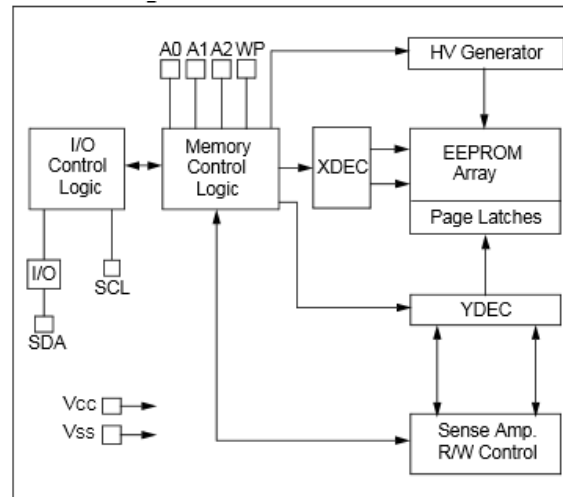


Figura 12. Modelo de la memoria 24LC256 que vamos a utilizar.

El precio es también reducido, costando en torno a 88 céntimos de dólar la unidad.

Los 256Kbit nos permiten guardar la información de unas 4 horas seguidas en ella para luego volcarla a otro soporte como por ejemplo una memoria USB que está conectada al Microcontrolador.

3.1.3 PUERTO RS232

El diseño del circuito para la transmisión de datos por medio del puerto serie RS - 232 requirió diseñar un circuito que pudiera manejar señales eléctricas con puertos serie.

Un 0 lógico está entre +1 y + 5 V.

Un 1 lógico está-a entre -1 y - 5 V.

La región entre -1 y +1 no está definida.

El voltaje en circuito abierto no debe exceder los 5V de referencia a tierra.

La corriente en cortocircuito no debe de exceder los 100 mA.

La comunicación del RS232 es asíncrona, pero las tramas de consulta ya vienen con etiquetas de tiempo y por motivos de sencillez anteriormente mencionados no hemos incluido señal de reloj.

La figura muestra un RS232 con un puerto serie, las señales de entrada al dispositivo vienen directamente del PIC como se puede ver en las imágenes anteriores de la placa.

3.2 IMPLEMENTACIÓN DEL CIRCUITO ELECTRÓNICO Y PROGRAMACIÓN DEL PIC.

La implementación que teníamos preparada sería la siguiente, es un diseño muy simple para conseguir almacenar la información que deseamos en la memoria y así poder gestionarla desde nuestro programa instalado en el servidor,

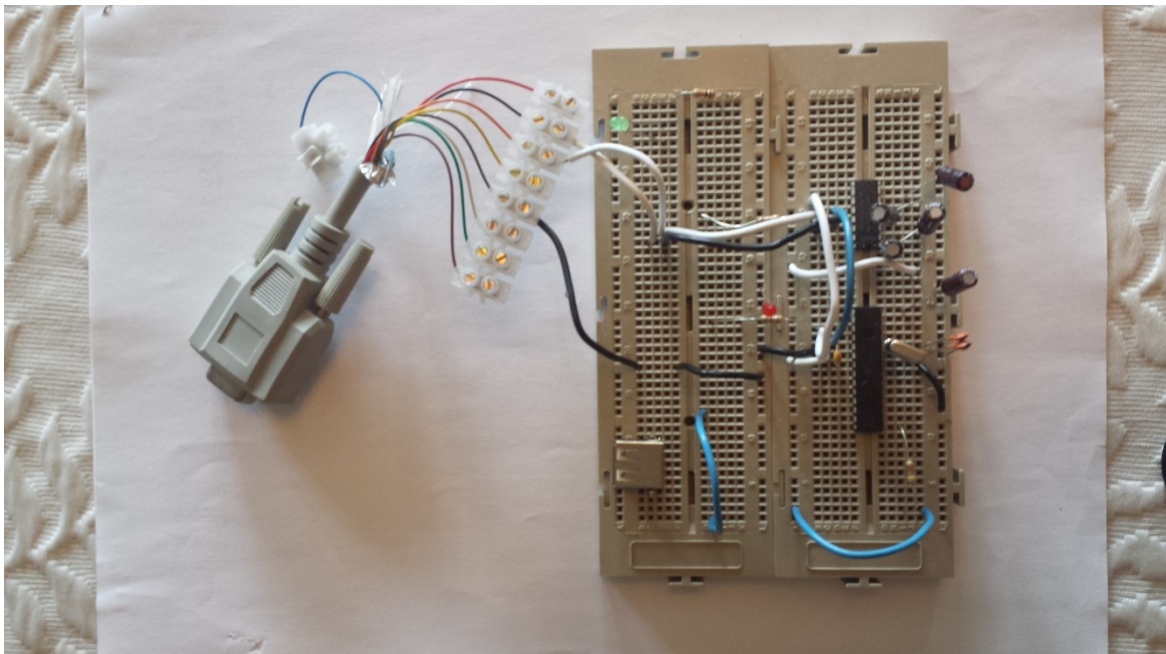


Figura 13. Nuestra placa base y micro controlador preparado para recuperar la información y almacenarla en un dispositivo USB.

Disponemos de la conexión del puerto con el contador, el USB para volcar los datos y el PIC programado para realizar una consulta prefijada que siempre respeta las mismas pautas según el protocolo y cuando el retorno es correcto lo guarda en la memoria.

El código del microcontrolador que tenemos actualmente sería el siguiente a alto nivel:

Bucle de lectura cada 15 segundos con la petición de la trama instantánea.

- Escucha de la trama.
- Evalúa el retorno de la petición
- Envío de la trama a escribir en memoria USB

End-Bucle

Las lecturas que se hacen continuamente sobre el contador para así registrar los datos que va devolviendo este suelen tener una cadencia entre 15-25 segundos que es el tiempo de respuesta que nos suele dar.

En la implementación práctica hemos tenido problemas con los picos de tensión, las desconexiones del aparato y su ubicación dentro del entorno, en ciertos momentos no ha sido fácil de llevar. Tampoco hemos conseguido mantener estable en el tiempo el sistema a más de un día.

Al disponer de un sistema alternativo y como hemos comentado en los objetivos, por necesidades de la empresa este desarrollo se ha dejado en stand by.

3.3 ESTUDIO DEL PROTOCOLO DE COMUNICACIÓN QUE PERMITA EL ANÁLISIS DEL CONSUMO.

El análisis del protocolo fue una de las partes más complejas de nuestro desarrollo ya que las especificaciones de red eléctrica no son fácilmente entendibles y tuvimos varios errores en cuanto a las posiciones hexadecimales de algunos datos concretos que eran necesarios para nuestras tramas, otros datos han de tener ciertos

movimientos de bytes para poder hacer lecturas concretas, en otras ocasiones nos topamos con que las unidades que venían estaban en Kilovatios/hora en vez de vatios como esperábamos mismo caso cuando nos encontrábamos intensidades que venían en mili amperios y otros como por ejemplo el tiempo venían incompletos ya que le faltaban los segundos, siempre venía xx:xx:00 y esto no era posible registrarlo así en base de datos si queríamos guardar varias medidas dentro del mismo minuto y debimos de implementar un código para poder evitar esto.

La Extracción actualmente se realizará mediante un proceso Java desde el PC al puerto COM del contador como hemos comentado anteriormente. Ahora detallaremos un poco su funcionamiento,



Figura 14. Nuestro contador de energía conectado a pc con convertidor RS232 y al PC.

Adjuntamos también una captura del programa con el que realizamos la obtención de los datos del contador, este es el medio mediante el cual extraemos actualmente la información. Podemos ver los parámetros que necesitamos informar para su conexión,

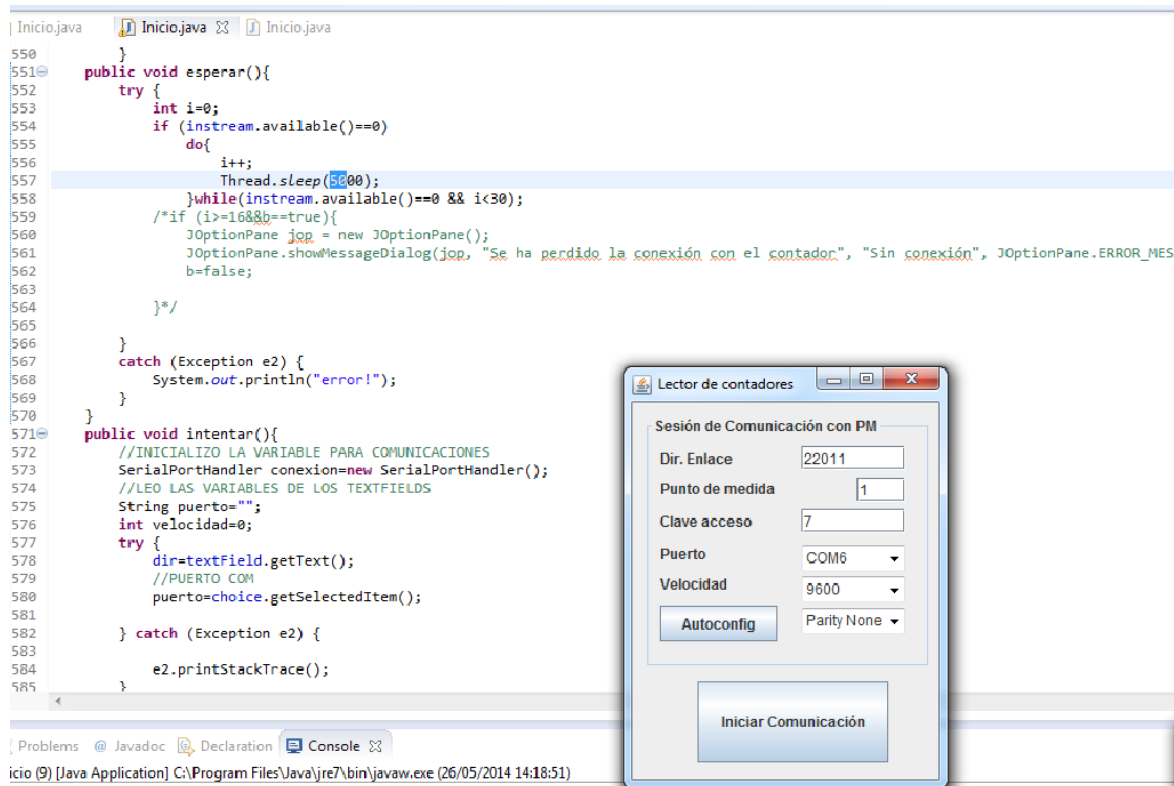


Figura 15. Imagen de conexión de datos instantáneos.

El proceso de captura de datos se centra en la misma pauta que teníamos con el PIC, sería el lanzamiento al contador de una petición de datos instantánea mediante un bucle de lectura que hace el lanzamiento de dichas peticiones.

Las tramas que se reciban con contestación de "Solicitud de valores instantáneos OK" del contador las guardaremos en el fichero de datos.

```

*****INICIO DE CONEXIÓN Y ADQUISICIÓN DE VALORES INSTANTÁNEOS*****
-> 10 49 fb 55 99 16
<- 10 0b fb 55 5b 16
-> 10 40 fb 55 90 16
<- 10 00 fb 55 50 16
-> 10 49 fb 55 99 16
<- 10 0b fb 55 5b 16
clave de acceso:
-> 68 0d 0d 68 73 fb 55 b7 01 06 01 00 00 07 00 00 00 89 16
<- 10 00 fb 55 50 16
CLAVE DE ACCESO ACEPTADA
-> 10 5b fb 55 ab 16
<- 68 0d 0d 68 08 fb 55 b7 01 07 01 00 00 07 00 00 00 1f 16
Solicitud valores instantaneos:
-> 68 0c 0c 68 73 fb 55 a2 03 05 01 00 00 c0 c1 c2 b1 16
<- 10 00 fb 55 50 16
SOLICITUD VALORES INSTANTÁNEOS OK
-> 10 5b fb 55 ab 16
<- 68 68 68 68 08 fb 55 a3 03 05 01 00 00 c0 6e 12 00 00 00 00 00 08 01 00 00 00 0
-> 10 7b fb 55 cb 16
  
```

Figura 16. Imagen de cómo adquirimos los datos instantáneos con petición desde el PC.

Este programa es el que cumpliría el requisito que podemos ver en la parte de objetivos 1.3.3. Podemos ver en texto a modo display el funcionamiento del bucle para tratar el protocolo de red eléctrica. Como se puede ver es un tratamiento bastante simple en el que mandamos las peticiones y cuando tenemos acceso por la clave

Las tramas de datos se guardan en un fichero de texto como podemos ver en la imagen inferior.

Figura 17. Trama de datos extraída del contador.

El desarrollo del software se centró en la necesidad de comunicar un ordenador mediante un programa java con el contador a través del puerto serie. El conocimiento del protocolo de comunicación permite poder extraer la información necesaria de las tramas instantáneas que se van generando en el contador.

Como hemos comentado anteriormente el desarrollo Hardware de este PFC es posible que acabe utilizando una conexión por 3G/GSM/Wifi o similar para enviar esta información directamente al Host en cuestión que los trate o a la base de datos. Consideramos que es la idea óptima desde un punto de vista funcional pero posiblemente mucho más cara que la adoptada en este proyecto inicialmente. Habrá que buscar un compendio mucho más práctico y en algún momento se retomará esta idea para tomar las medidas de las localizaciones fácilmente accesibles. Igualmente esto se verá en los apartados de posibles mejoras en el punto 6 del documento.

4 FASES DEL DESARROLLO SOFTWARE

En los próximos puntos veremos cómo hemos diferenciado la creación, diseño y pruebas del software que se ha construido en este PFC, lo hemos dividido en 3 partes.

La creación y extracción de los datos extraídos del contador, la base de datos y la aplicación.

4.1 DISEÑO DE LA APLICACIÓN EXCEL DE CARGA DE LOS FICHEROS.

En este apartado veremos cómo hemos desarrollado la aplicación para cumplir con los objetivos planteados en el punto 1.3. De este estudio y desarrollo salen ciertas clases y métodos que veremos más o menos en detalle en función de su necesidad.

Comentar también que utilizamos ciertas librerías específicas que nos proporcionaban ciertas necesidades de nuestro proyecto, había varias donde elegir, pero decidimos que utilizaríamos las apache POI para la creación de libros Excel desde Java ya que son ampliamente utilizadas y tienen mucho material de consulta, también han sido ampliamente testeadas, con lo cual son muy eficientes.

4.1.1 Introducción al protocolo de comunicaciones ASDU con lecturas instantáneas

En este punto desarrollaremos cómo funciona el protocolo ASDU para lectura de valores instantáneos y lectura de los datos del contador que se consumen en el momento actual de la consulta, según el tipo 162.

La primera parte antes de construir la aplicación fue la generación de una Excel que fuera capaz de convertirnos los valores recibidos en el soporte físico a valores legibles para poder interpretarlos, para posteriormente aplicar un análisis de los mismos.

4.1.1.1 Tipo 162/Valores Instantáneos. Petición.

Se empleará este ASDU para interrogar al RM sobre la programación/estado de los Elementos del Grupo Instantáneos.

<i>Tipo = 162</i>	
<i>0</i>	<i>n</i>
<i>Causa de transmisión</i>	
<i>Dirección del punto de medida</i>	
<i>Dirección de registro = 0</i>	
<i>Dirección de Objeto 1</i>	
<i>(= grup)</i>	
<i>Dirección de Objeto 2</i>	
<i>(= grup)</i>	
<i>..etc..</i>	
<i>Dirección de Objeto n</i>	
<i>(= grup)</i>	

Figura 18. Consulta del punto de medida

Las causas de transmisión serán:

CM ⇒ RM	5	Petición
RM ⇒ CM	14	El ASDU no está definido

Figura 19. Causas de transmisión para valores instantáneos.

Los Objetos de Información indican qué información se está solicitando al RM. El orden de petición es indiferente, pero en ningún caso se podrá solicitar dos veces el mismo Objeto de Información en un mismo ASDU.

Si el RM está en disposición de enviar la información requerida, se empleará el ASDU tipo 163 para ello.

4.1.1.2 Tipo 163/Valores instantáneos. Respuesta

Se empleará este ASDU para enviar al CM (Concentradores de medida) los valores de los Elementos del Grupo Instantáneos.

<i>Tipo = 163</i>	
<i>0</i>	<i>n</i>
<i>Causa de transmisión</i>	
<i>Dirección del punto de medida</i>	
<i>Dirección de registro = 0</i>	
<i>Dirección de ObjetoSolicitado 1</i> <i>(= grup)</i>	
<i>Info relativa al Objeto Solicitado 1</i>	
<i>Dirección de ObjetoSolicitado 2</i> <i>(= grup)</i>	
<i>Info relativa al Objeto Solicitado 2</i>	
<i>..etc..</i>	
<i>Dirección de ObjetoSolicitado n</i> <i>(= grup)</i>	
<i>Info relativa al Objeto Solicitado n</i>	

Figura 20. Respuesta de valores instantáneos.

Las causas de transmisión serán:

| RM ⇒ CM | 5 Solicitada

Figura 21. Envío de objetos solicitados en el 163 del ASDU.

Se envían los Objetos solicitados.

4.1.1.3 Elem. Info.: Totalizadores de energías (Agrup: V / Dir. Obj.: 192)

Contiene los valores instantáneos de los totalizadores de energías, referidos a valores primarios.

La estructura del elemento es:

```

Tot_Energias:= {
  UI30[1..30]<0, 999999999>:=      KWh - Activa importación
  BS1[31]<0>                        No Usado
  BS1[32] <0, 1>:=                <0>:= Medida Válida
                                     <1>:= Medida Invalida
  UI30[33..62]<0, 999999999>:=    KWh - Activa exportacion
  BS1[63]<0>                        No Usado
  BS1[64] <0, 1>:=                <0>:= Medida Válida
                                     <1>:= Medida Invalida
  UI30[65..94]<0, 999999999>:=    KVARh - Reactiva Q1
  BS1[95]<0>                        No Usado
  BS1[96] <0, 1>:=                <0>:= Medida Válida
                                     <1>:= Medida Invalida
  UI30[97..126]<0, 999999999>:=   KVARh - Reactiva Q2
  BS1[127]<0>                       No Usado
  BS1[128] <0, 1>:=                <0>:= Medida Válida
                                     <1>:= Medida Invalida
  UI30[129..158]<0, 999999999>:=  KVARh - Reactiva Q3
  BS1[159]<0>                       No Usado
  BS1[160] <0, 1>:=                <0>:= Medida Válida
                                     <1>:= Medida Invalida
  UI30[161..190]<0, 999999999>:=  KVARh - Reactiva Q4
  BS1[191]<0>                       No Usado
  BS1[192] <0, 1>:=                <0>:= Medida Válida
                                     <1>:= Medida Invalida
  CP40[193..232]:=                 Etiqueta de tiempo. Hora RM.
}

```

Figura 22. Estructura de los totalizadores de energías.

De esta tabla de valores, sacamos los totales energéticos.

4.1.1.4 Elem. Info.: Potencias activas (Agrup: V / Dir. Obj.: 193)

Contiene los valores instantáneos de las potencias activas, reactivas y Factores de Potencia (total y por fases), referidos a valores primarios.

La estructura del elemento es:

Pot:= {	
UI24[1..24]<0, 9999999>:=	P. Activa Total (KW).
UI24[25..48]<0, 9999999>:=	P. Reactiva Total (KVA _R).
UI10[49..58]<0, 1000>:=	Factor de Potencia Total (en milésimas).
BS1[59]<0, 1>:=	<0>:= P. Activa Total es importada.
	<1>:= P. Activa Total es exportada.
BS1[60]<0, 1>:=	<0>:= P. Reactiva Total es Q1/Q2.
	<1>:= P. Reactiva Total es Q3/Q4.
BS3[61..63]<0>:=	No Usados
BS1[64] <0, 1>:=	<0>:= P. Total. Medidas Válidas
	<1>:= P. Total. Medidas Invalidas
UI24[65..88]<0, 9999999>:=	P. Activa Fase I (KW).
UI24[89..112]<0, 9999999>:=	P. Reactiva Fase I (KVA _R).
UI10[113..122]<0, 1000>:=	Factor de Potencia (cos phi). Fase I (en milésimas).
BS1[123]<0, 1>:=	<0>:= P. Activa Fase I es importada.
	<1>:= P. Activa Fase I es exportada.
BS1[124]<0, 1>:=	<0>:= P. Reactiva Fase I es Q1/Q2.
	<1>:= P. Reactiva Fase I es Q3/Q4.
BS3[125..127]<0>:=	No Usados
BS1[128] <0, 1>:=	<0>:= P. Fase I. Medidas Válidas
	<1>:= P. Fase I. Medidas Invalidas
UI24[129..152]<0, 9999999>:=	P. Activa Fase II (KW).
UI24[153..176]<0, 9999999>:=	P. Reactiva Fase II (KVA _R).
UI10[177..186]<0, 1000>:=	Factor de Potencia (cos phi). Fase II (en milésimas).
BS1[187]<0, 1>:=	<0>:= P. Activa Fase II es importada.
	<1>:= P. Activa Fase II es exportada.
BS1[188] <0, 1>:=	<0>:= P. Reactiva Fase II es Q1/Q2.
	<1>:= P. Reactiva Fase II es Q3/Q4.
BS3[189..191]<0>:=	No Usados
BS1[192] <0, 1>:=	<0>:= P. Fase II. Medidas Válidas
	<1>:= P. Fase II. Medidas Invalidas
UI24[193..216]<0, 9999999>:=	P. Activa Fase III (KW).
UI24[217..240]<0, 9999999>:=	P. Reactiva Fase III (KVA _R).
UI10[241..250]<0, 1000>:=	Factor de Potencia (cos phi). Fase III (en milésimas).
BS1[251]<0, 1>:=	<0>:= P. Activa Fase III es importada.
	<1>:= P. Activa Fase III es exportada.
BS1[252]<0, 1>:=	<0>:= P. Reactiva Fase III es Q1/Q2.
	<1>:= P. Reactiva Fase III es Q3/Q4.
BS3[253..255]<0>:=	No Usados
BS1[256] <0, 1>:=	<0>:= P. Fase III. Medidas Válidas
	<1>:= P. Fase III. Medidas Invalidas
CP40[257..296]:=	Etiqueta de tiempo. Hora RM.
}	

Figura 23. Tabla que nos muestra los totales de potencias activas consultadas.

En estas tablas aparte de poder ver las equivalencias para extraer los valores, también nos indica los bits en los que podemos validar si las medidas son válidas.

También podemos ver las unidades en las que nos van a llegar dichas medidas.

4.1.1.5 Elem. Info.: V_I (Agrup: V / Dir. Obj.: 193)

Contiene los valores instantáneos de las tensiones y corrientes, referidos a valores secundarios.

La estructura del elemento es:

```

V_I:= {
  UI24[1..24]<0, 9999999>:=      Intensidad Fase I (décimas de amperio).
  UI30[25..54]<0, 999999999>:=   Tensión Fase I (décimas de volt).
  BS1[55]<0>:=                  No Usados
  BS1[56] <0, 1>:=              <0>:= Fase I. Medidas Válidas
                                <1>:= Fase I. Medidas Invalidas
  UI24[57..80]<0, 9999999>:=     Intensidad Fase II (décimas de amperio).
  UI30[81..110]<0, 999999999>:=  Tensión Fase II (décimas devolt).
  BS1[111]<0>:=                  No Usados
  BS1[112] <0, 1>:=              <0>:= Fase II. Medidas Válidas
                                <1>:= Fase II. Medidas Invalidas
  UI24[113..136]<0, 9999999>:=   Intensidad Fase III (décimas de amperio).
  UI30[137..166]<0, 999999999>:= Tensión Fase III (décimas devolt).
  BS1[167]<0>:=                  No Usados
  BS1[168] <0, 1>:=              <0>:= Fase III. Medidas Válidas
                                <1>:= Fase III. Medidas Invalidas
  CP40[169..208]:=              Etiqueta de tiempo. Hora RM.
}

```

Figura 24. Valores de diferentes fases de medidas de tensión e intensidad.

4.1.2 Documentación

Del estudio de los datos anteriores extraídos de red eléctrica podemos mostrar la siguiente tabla, en ella vemos la interrelación directa entre nuestras variables y los datos que obtenemos en hexadecimal.

ASDU 162				
ACT	46	45	44	
REACT	49	48	47	
FP	51	50		
V1	88	87	86	85
V2	95	94	93	92

V3	102	101	100	99
I1	84	83	82	
I2	91	90	89	
I3	98	97	96	
ACT_1	54	53	52	
ACT_2	62	61	60	
ACT_3	70	69	68	
REACT_1	57	56	55	
REACT_2	65	64	63	
REACT_3	73	72	71	
FP1	59	58		
FP2	67	66		
FP3	75	74		
ACT_IMPORT	17	16	15	14
ACT_EXPORT	21	20	19	18
REACT_1	25	24	23	22
REACT_2	29	28	27	26
REACT_3	33	32	31	30
REACT_4	37	36	35	34

Figura 25. Tabla con las posiciones de los datos en hexadecimal.

Esta tabla nos proporciona las posiciones de las variables en el interior trama en el caso de una consulta instantánea al mismo.

4.1.3 Tipos de Trama

Dentro del ASDU podemos hacer diferentes tipos de consultas, en el Anexo 7.1 podemos ver diferentes maneras de consultarlo, en nuestro caso concreto queremos un tipo de “consulta instantánea” que viene del estándar 162 como comentamos anteriormente, en este anexo veremos claramente cómo utilizar las tramas tanto de consulta como de recepción de datos, con todos los parámetros necesarios para la implementación del desarrollo.

4.1.4 Parámetros

Los parámetros a extraer actualmente son todos los que nos ofrece el protocolo, son muchos y a día de hoy no se quieren utilizar todos ellos, en la base de datos guardamos los que se han considerado dentro de los objetivos de este proyecto y se consideran necesarios para tener un análisis eficiente.

En estos momentos se ha decidido tenerlos todos para poder ir sacando conclusiones con las variables más características, si de cara a futuro vemos que no hicieran falta tantos, liberaríamos memoria en tablas y sobre todo en la extracción de datos del contador.

4.1.5 El problema del tiempo

En el protocolo actual no aparecen los segundos en los que se hace la petición con la trama, nos da simplemente la hora y minuto de la consulta al contador, este dato forma parte de la clave primaria elegida en el modelo de la base de datos, es un dato relativamente relevante ya que nos interesa saber la media de consumos horarios. A nivel de minutos queremos saber la media de consultas que podemos hacer y el valor que tienen las mismas, tomamos la decisiones de generar un numero aleatorio incremental entre 0 y 60 para cada una de estas medidas, por lo general oscilaran entre 2-4 al minuto.

```
public String returnhour (String[] sCadena, String hora_ant){
    int Segundos_actual = 0;
    String Segundos = "00" ;
    String hora = (ConvHexIntFec(sCadena[104] , 3)+":"+ConvHexIntFec(sCadena[103],2)+" "+Segundos_actual);
    //en caso de que tengamos repetidas varias mediciones en el mismo minuto para esta medicion
    if (hora.subSequence(0, 5).equals(hora_ant.subSequence(0, 5)))
    {
        Segundos_actual = Integer.parseInt(hora_ant.substring(6)) + (int)(Math.random()*9 + 1);}
    //si la hora:segundo es nueva para esta medicion
    else {
        Segundos_actual = (int)(Math.random()*9 + 1) + 10;}
    //por si en algún caso hacemos más de 6 consultas aseguramos no pasar los segundos de un minuto
    if (Segundos_actual > 60 ) Segundos_actual = 61 - Segundos_actual ;
    if (Segundos_actual < 0 ) Segundos_actual = 00 ;
    Segundos = Integer.toString(Segundos_actual);
    if (Segundos.length() == 1) { Segundos = "0" + Segundos;}
    hora = hora.substring(0, 5)+" "+Segundos_actual;
    return hora;
}
```

Figura 26. Nuestra solución particular para grabar los segundos en la BBDD

Con este código generamos los datos necesarios para insertar en la tabla consumo de contadores, nos da un segundo aleatorio creciente, comparando con el anterior.

4.1.6 Conversión de los datos del protocolo

Los datos nos vienen en diferentes unidades y han de ser convertidos a formato legible, para ello hemos montado métodos dentro de una clase, ExcelUtils que nos ayudan a transformar los mismos. Esto lo veremos en el apartado de diseño de la aplicación, en el punto 4.3.

4.2 DISEÑO DE LA BASE DE DATOS

En este punto vamos a tratar de comentar como vamos a manejar los datos que vamos a incluir en nuestro sistema y la forma en la cual vamos a registrar los mismos.

4.2.1 Introducción

La BBDD que hemos construido es un modelo base con los requerimientos actuales a analizar, el objetivo es poder empezar a tomar los datos que necesitas ordenados y consultados de manera eficiente, hemos creado tablas para poder volcar esos datos en medias horarias más manejables, esta tabla es la tabla histórico contador que pretende tener medias horarias de los datos reflejados de un contador.

También hemos considerado medidas de interés han de ser guardadas en una tabla a parte para su fácil detección en interpretación, todavía se están definiendo por parte de AYF cuáles serán estos requisitos y habrá un proceso nocturno que diariamente revise en cada contador cuando se cumplen los requisitos comentados y los cargará

en la tabla para su posterior procesamiento, la idea será ciertos valores que se desvíen un amplio porcentaje de la media de valores obtenidos en el histórico.

Para que no hubiera confusión con los valores que se repiten en las tablas hemos asentado prefijos con los nombres acrónimos de las tablas delante de cada campo siguiendo algunos consejos de modelación de Bases de Datos.

4.2.2 Elección de la base de datos

La base de datos elegida es MySQL, hay varias razones que llevan a su elección, como por ejemplo la conectividad con gran variedad de sistemas y soportes para un gran número de lenguajes pero principalmente por ser la base de datos de código abierto más utilizada del mundo, esto junto a su soporte técnico y la gran cantidad de información que circula en la red en <http://dev.mysql.com/> con un workbench muy completo y gratuito nos daba todas las necesidades y requerimientos que necesitamos ahora y en previsión de que escalen los requisitos de esta en un futuro.

Su integración con Java y su amplia documentación fueron también un punto fuerte para su elección.

4.2.3 Elección de las variables y los tipos de dato

La elección de las variables de trabajo ha sido transmitida por la empresa AYF según sus requerimientos actuales y necesidades con las que actualmente contemplan las mejoras energéticas de sus clientes, hemos intentado adaptar esto a un modelo manejable de tablas que no sean muy pesadas y con tablas de histórico para volcar estos datos cuando ya no sean manejables en una sola tabla.

En la siguiente imagen se puede apreciar el modelo que tenemos de la BBDD con las tablas que vamos a mencionar,

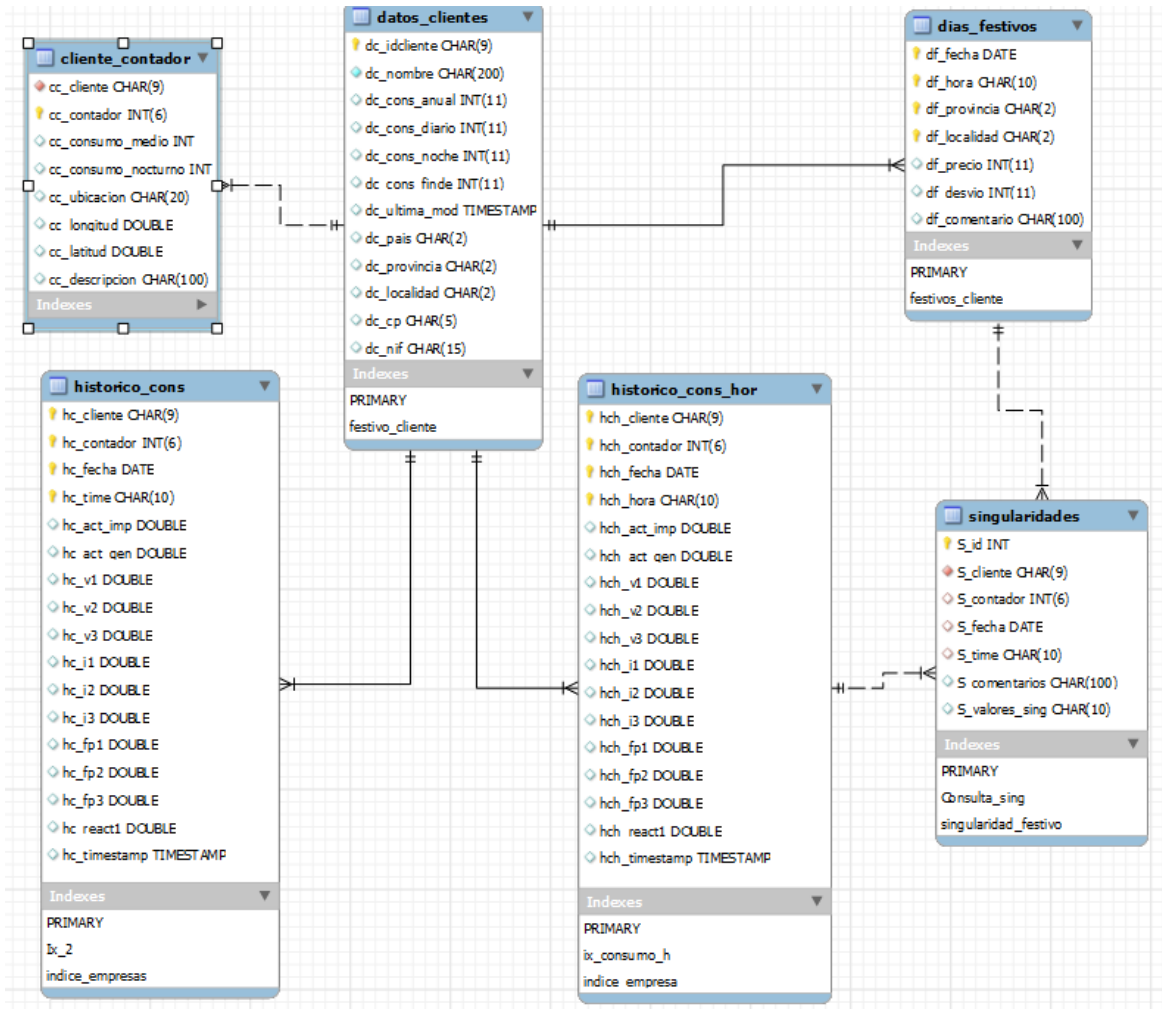


Figura 27. Nuestro modelo entidad relación.

Las variables a almacenar de los contadores son las siguientes, todas definidas en base de datos como Double,

- Potencia activa importada
- Potencia activa generada
- Potencia e intensidad en primera fase
- Potencia e intensidad en segunda fase
- Potencia e intensidad en tercera fase

- Potencia reactiva inductiva

Podemos ver también que estas variables se guardan en base de datos como datos Double para poder guardar el punto decimal.

Los tipos de datos más relevantes serán el IdCliente que definimos como alfanumérico de 9 para poder poner un acrónimo, será representativo a la hora de poder elegir el cliente en un menú desplegable de la aplicación.

El idContador en cambio sí que es un numérico auto incrementable de manera que constituya una Pk, junto al nombre del cliente.

- Nombre de cliente – acrónimo – Alfanumérico de 9 caracteres
- Contador de cliente - Entero de 6 dígitos

Los consumos horarios medios de los clientes son valores enteros de 11, nos sirven para dar una media en función de los consumos más divergentes de la empresa de los cuales podemos extraer pautas,

- Consumo anual – Entero de 11 dígitos
- Consumo diario– Entero de 11 dígitos
- Consumo nocturno– Entero de 11 dígitos
- Consumo fin de semana– Entero de 11 dígitos

País provincia y localidad a día de hoy son acrónimos ya que el 100% del volumen de trabajo hoy en día de la empresa sería aquí. Se deja la posibilidad de que de cara a un futuro se trabaje fuera, tener una tabla de conversión de estos acrónimos según países y/o localidades,

- País – Alfanumérico de 2
- Provincia – Alfanumérico de 2
- Localidad – Alfanumérico de 2

4.2.4 Elección de las tablas siguiendo las formas normales

Hemos de definir cuál es la base de nuestro modelo, la tabla de la que parte nuestro modelo y esta es sin duda la tabla de clientes, estos clientes serán la base de nuestro modelo y de ellos “cuelgan” contadores que son los atributos que nos interesan a día de hoy de nuestros clientes, de estos conseguimos nuestras lecturas y medidas que es lo que queremos procesar, estos contadores nos dan las medidas según una fecha (aaaa-mm-dd) y una hora (hh:mm:ss) con lo cual estos cuatro datos serán la clave primaria de nuestra tabla que contempla las medidas y los Id en caso del cliente será un alfanumérico y el contador un entero auto incrementable.

Hemos definido un timestamp a la hora de registrar las medidas y que se inserten en la BBDD cosa que nos puede ayudar en caso de que hagamos algún tipo de carga errónea o necesidad de algún tipo de modificación en cierto momento de carga.

En el apartado 7.2 podemos ver la definición de las formas normales para la creación de modelos entidad relación.

4.2.5 Modelo de la base de datos

Ahora veremos cada una de las tablas del modelo con una pequeña explicación.

The screenshot shows a database management tool interface. At the top, there is a 'Table Name' field containing 'datos_clientes' and a 'Schema' field. Below this, a 'Columns' table is displayed with the following data:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
dc_idcliente	CHAR(9)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dc_nombre	CHAR(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dc_cons_anual	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dc_cons_diario	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dc_cons_noche	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dc_cons_finde	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dc_ultima_mod	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dc_pais	CHAR(2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 28. Tablas datos_clientes

Como podemos ver, todo despliega de la tabla datos clientes, en ella tenemos una clave por cliente, en la cual guardamos todos sus datos como puede ser el nombre, el país, DNI-NIF, localidad, código postal, etc. También se alberga en esta tabla una

media de los consumos de todos sus contadores a nivel de cliente para mostrarse en la información que nos proporcionará la aplicación. Dispone de un campo timestamp que nos marca la última modificación que hemos hecho en los datos del cliente y el día que se dio de alta.

Esta tabla tiene una relación uno a muchos con la tabla cliente_contador, esto nos permite relacionar varios contadores para un cliente.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
df_fecha	DATE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
df_hora	CHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
df_provincia	CHAR(2)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
df_localidad	CHAR(2)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
df_precio	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
df_desvio	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
df_comentario	CHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 29. Tabla días_festivos

También tenemos un índice con la clave primaria de días_festivos para poder hacer consultas eficientes en caso de necesitar buscar los festivos de las localidades. Hemos acordado que cuando no haya una hora concreta y aplique a todo un día se pondrán las 0:0:0.0000. Ciertos días festivos o en ciertos periodos horarios la energía experimenta un precio diferente, nos interesa tenerlos cargados para luego poder contrastar con las singularidades que tengamos y poder contrastarlas.

En ella tenemos guardados unos periodos tarifarios en función del cliente-provincia-localidad a la hora, con esta tabla esperamos poder explicar los desvíos que encontraremos en la tabla "singularidades", a esta se le asigna un Id de singularidad por el cual luego buscaremos, también le hemos creado unos índices para poder consultar con las claves ajenas de las tablas con las que está relacionada.

Table Name: singularidades Schema:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
S_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S_cliente	CHAR(9)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S_contador	INT(6)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S_fecha	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S_time	CHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S_comentarios	CHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S_valores_sing	CHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 30. Tabla singularidades

Estas singularidades se procesarán posteriormente mediante un proceso selectivo o de manera manual.

Table Name: cliente_contador Schema:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
cc_contador	INT(6)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
cc_consumo_medio	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cc_consumo_nocturno	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cc_ubicacion	CHAR(120)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cc_longitud	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cc_latitud	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cc_descripcion	CHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 31. Tabla cliente_contador

La tabla cliente_contador es una de las tablas básicas de nuestro modelo de datos, en ella tenemos registrado todos los datos necesarios del contador a nivel de registro y ubicación, también el cliente con el cual se relaciona y todas las medias de sus consumos.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
hch_cliente	CHAR(9)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hch_contador	INT(6)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hch_fecha	DATE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hch_hora	CHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hch_act_imp	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
hch_act_gen	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
hch_v1	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
hch_v2	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figura 32. Tabla histórico_cons_hor

La tabla histórico_cons_hor pretende ser la tabla que guarde todos los registros de información volcados de un periodo considerable grabada en intervalos entre 15 y 25 segundos, no tenemos todavía datos de volumetría pero la idea es poder guardar periodos entre 6 meses a 1 año, con esto será suficiente para tener las consultas de consumos cercanos para su análisis. La idea una vez pasado dicho periodo es que se guardaran en la tabla histórico_cons todos los registros cada periodo marcado de tiempo, para ello se guardaran cada vez en una partición diferente de la tabla que irá por fecha, así podremos saber qué partición tenemos que acceder para un rango de fechas a consultar de un cliente. Como comentamos, esto se decidirá una vez se conozcan concretamente las volumetrías y cuando los accesos a tabla comiencen a estar penalizados.

Se estudiará la posibilidad también de modificar los índices según los tipos de consultar que se consideren son los más frecuentes y/o necesarios, una vez se siga desarrollando la aplicación o mediante las consultas directas por el gestor de MySQL.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
hc_cliente	CHAR(9)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hc_contador	INT(6)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hc_fecha	DATE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hc_time	CHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hc_act_imp	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
hc_act_gen	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
hc_v1	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
hc_v2	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figura 33. La tabla histórico_cons

Esta será la tabla histórica en la que se vuelquen los datos. La idea es separar en particiones períodos anuales o semestrales.

4.2.6 SQL que contiene la BBDD

Las sentencias SQL para ejecutar en un cliente de MySQL para tener nuestro modelo y poder trabajar con él las incluimos en el anexo, en el apartado 120 Definición de la BBDD para no penalizar la lectura de esta redacción.

4.3 DISEÑO DE LA APLICACIÓN

El diseño de la aplicación se basó en una interpretación cómoda y fácilmente entendible de los requisitos establecidos. Para ello elegimos un punto de partida que sería nuestro cliente, desde este punto se nos permite ir bajando en detalle hasta poder consultar la ubicación de un contador.

4.3.1 Navegación de la aplicación

Vamos a hacer un breve ejemplo de lo que sería la navegación por la aplicación, mediante esta navegación introduciremos los métodos que hemos utilizado para implementar la funcionalidad que tiene la misma y los desarrollaremos posteriormente.

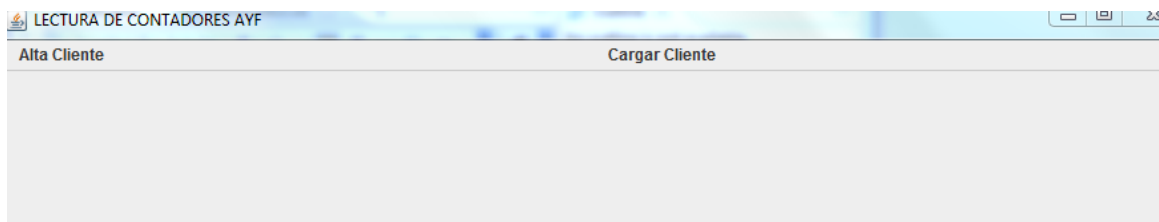


Figura 34. Pantalla de inicio

En esta pantalla inicial podemos ver dos opciones, dar de alta nuevos clientes en la aplicación o consultar los existentes.

A screenshot of the 'Alta de nuevo cliente' form. The form is titled 'Alta de nuevo cliente' and contains several input fields. On the left side, there are fields for 'IdCliente', 'Nombre', 'Pais', 'Provincia', 'Localidad', 'Codigo postal', and 'NIF'. On the right side, under the heading 'Medias de consumo', there are four fields: 'Consumo anual', 'Consumo diario', 'Consumo noche', and 'Consumo no laboral'. At the bottom of the form, there are two buttons: 'Confirmar' and 'Cancelar'.

Figura 35. Alta de cliente.

En esta pantalla poder ver la recogida de los datos del cliente, para su inserción en la base de datos en su tabla correspondiente. Tenemos unas validaciones para verificar la validez de los datos y formato, en caso de que algún dato fuera incorrecto, nos aparecería el mensaje correspondiente.

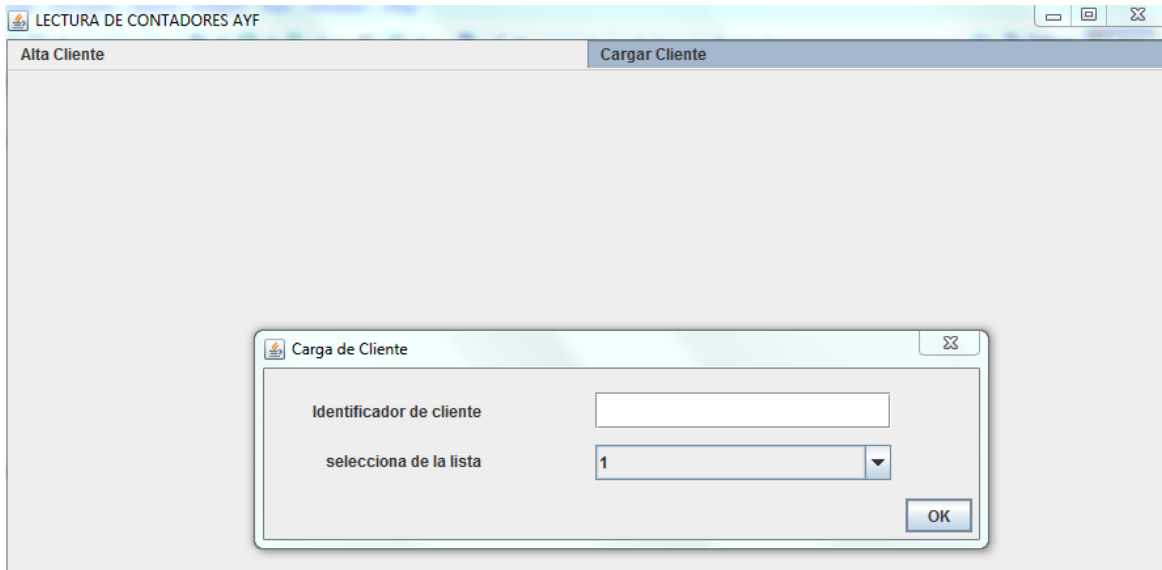


Figura 36. Pantalla de consulta de datos de cliente.

En esta pantalla nos da dos opciones, insertar el identificador de cliente directamente para buscarlo o elegir de una lista desplegable el IdCliente para su carga una vez se selecciona el botón ok.

LECTURA DE CONTADORES AYF

Alta Cliente Cargar Cliente

DATOS CLIENTE

Nombre Antonio Jose Vila García

IdUsuario ASEGOR

Pais ES

Provincia AB

Localidad A

CP 02005

NIF 48467238P

Consumos de cliente

Anual 120

Diario 110

Nocturno 80

Fin de semana 30

Cargar nuevos datos de un contador

Contadores Contador10

Consultar contador

Alta nuevo contador

Datos generales de los contadores

Salida

Consultar contadores

Figura 37. Datos de un cliente.

En esta pantalla podemos observar los datos que nos proporciona nuestra base de datos según el cliente, nos da una media de los consumos de todos sus contadores y según periodos tarifarios diferentes.

Desde aquí se nos permite 3 opciones,

- Consulta de todos los datos de todos los contadores del cliente.
- Consulta de un contador concreto en la lista desplegable de contadores del cliente.
- Alta de un nuevo contador para este cliente.

Las opciones de modificación de la Base de datos, eliminando contadores las dejamos para hacerlas desde el gestor.

LECTURA DE CONTADORES AYW

Alta Cliente Cargar Cliente

DATOS CLIENTE CONTADOR:10 CONTADOR:11

Nombre Antonio Jose Vila García

IdUsuario ASEGOR

País ES

Provincia AB

Localidad A

CP 02005

NIF 48467238P

Consumos de cliente

Anual	120
Diario	110
Nocturno	80
Fin de semana	30

Cargar nuevos datos de un contador

Contadores Contador11

Consultar contador

Alta nuevo contador

Datos generales de los contadores

Salida

Consultar contadores

Figura 38. Vista de consulta de los contadores del cliente

En la figura superior podemos observar cómo se despliegan las pestañas de consulta de los contadores del cliente que queremos analizar.

LECTURA DE CONTADORES AYF

Alta Cliente Cargar Cliente

DATOS CLIENTE CONTADOR:1 CONTADOR:6 CONTADOR:12

Datos del contador

Cliente	Asirvent	Consumo medio	5	Long	-0.4164714665636061
IdContador	12	Consumo nocturno	2	Lat	38.368972254282774
Localizacion	Calle san sebastian n°13				
Descripcion	Contador de Esmeralda				

Cargar datos al contador

Entrada

Salida

Previsualizar Datos Cargar Datos Consultar Historico Visualizar posicion

Cerrar

Figura 39. Vista de contador de cliente.

En el ejemplo de la figura anterior, podemos ver los datos que se muestran en la consulta del contador, con sus opciones correspondientes, que serían:

- Pre visualizar los datos a cargar en BBDD en una Excel, sin llegar a insertarlos (verificación previa).
- Carga de datos a la base de datos.
- Consulta a la base de datos.
- Visualizar la ubicación del contador.

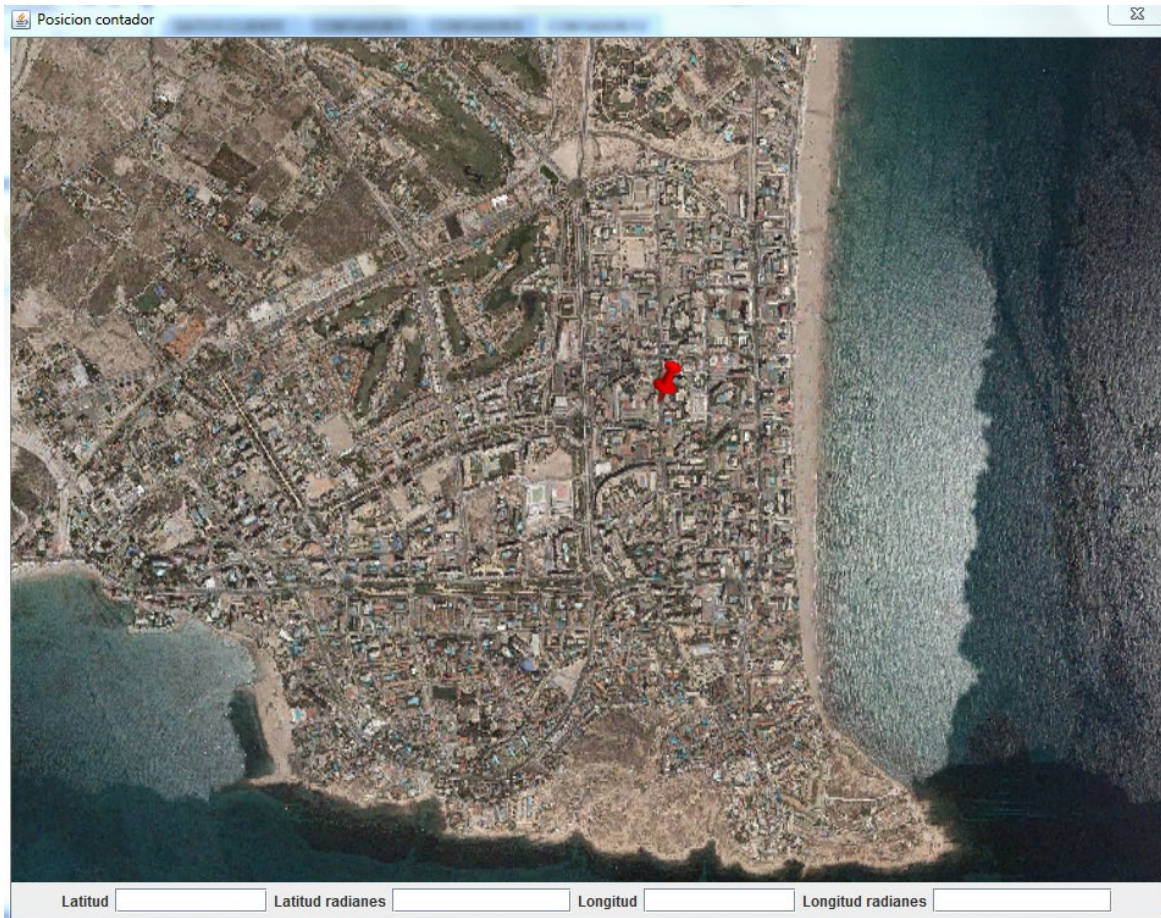


Figura 40. Localización del contador

La localización del contador de manera visual fue un requisito que se concibió en la fase de pruebas de la aplicación, con motivo de la difícil ubicación que ciertos contadores ubicados en sitios no urbanizados. Utilizando las librerías Nasa Word Wind para aplicaciones Java obtuvimos el resultado requerido, es un software muy potente y de increíble precisión, da la posibilidad de cargar distintos tipos de mapas de detalle, para ello el pc debe de tener conexión a internet para ir refrescando por capas las imágenes del mapa.

La consulta y la pre visualización de los datos nos mostrarán una pantalla como podemos ver en la imagen inferior, en caso de que todo sea correcto nos permitirá hacer un filtrado por fechas para agilizar la consulta de los datos,

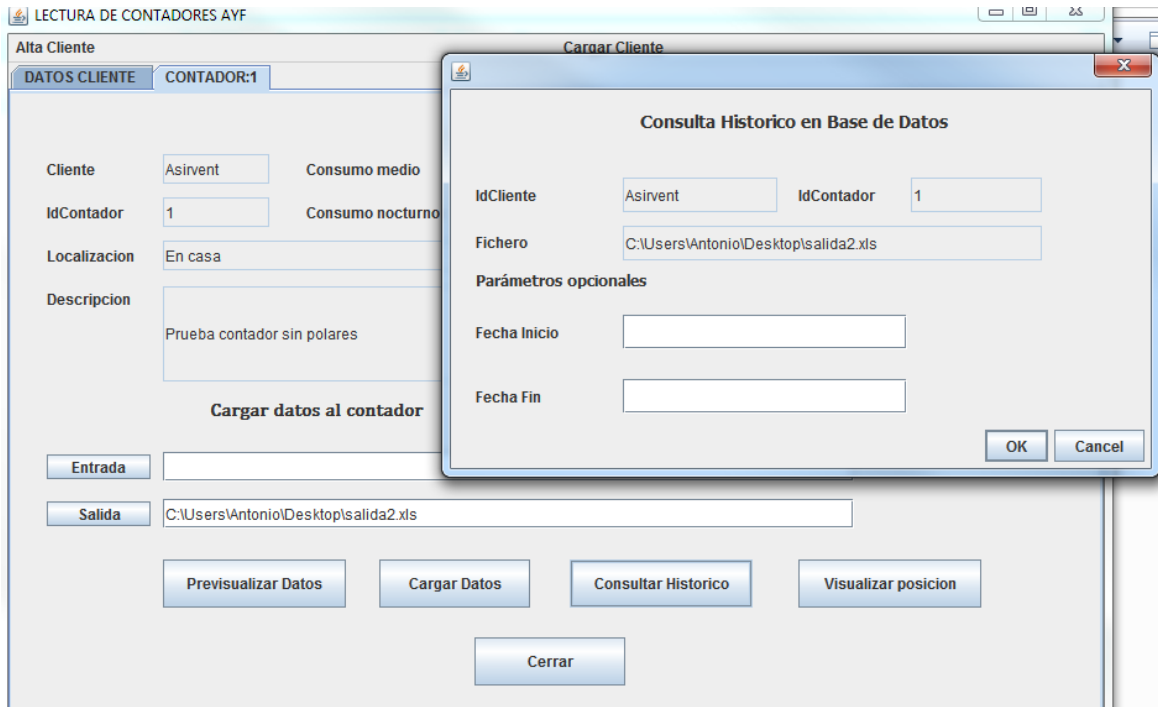


Figura 41. Consulta histórico contador

Nos dará un mensaje en caso de que todo haya ido ok,

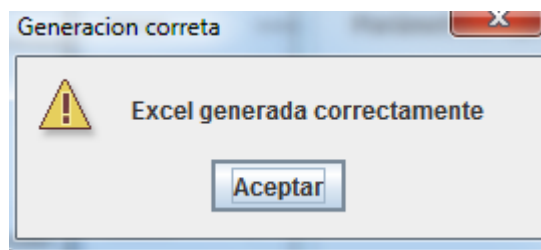


Figura 42. Salida Ok de la consulta.

En este caso el contador no tenía datos,

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	CLIENTE:	CONTADOR:	FECHA:	HORA:	EAC IMPORTADA:	EAC GENERADA:	V1:	V2:	V3:	I1:	I2:	I3:	FP1:	FP2:	FP3:	REACT 1:	TimeStamp:
2																	
3																	
4																	
5																	
6																	

Figura 43. Contador vacío.

En caso de que en la consulta no hubiéramos informado correctamente, nos habría dado sus correspondientes mensajes de error en caso de no informar alguno de los datos obligatorios,



Figura 44. Mensaje de error, por la no especificación del fichero de salida.

Otra de las opciones que teníamos a la hora de consultar un cliente, era la opción de dar de alta un nuevo contador para este cliente, la página en la que haríamos la funcionalidad sería la que sigue.

Figura 45. Alta de contador cliente.

Para finalizar, podemos ver el resultado de la consulta de los datos de un cliente, se visualizaría en una Excel con el formato que podemos ver a continuación.

Figura 46. Pantalla de descarga de datos

CLIENTE	CANTADOR	FECHA	HORA	EAC IMPORTADA	EAC GENERADA	V1	V2	V3	I1	I2	I3	FP1
ASEGOR		10/2013-12-21	02:22:14	0	234.8	236.200000000000002	234.4	7.4	7.800000000000001	7.7	0.2071	
ASEGOR		10/2013-12-21	02:22:15	0	234.8	236.200000000000002	234.4	7.4	7.800000000000001	7.7	0.2071	
ASEGOR		10/2013-12-21	02:22:18	0	234.8	236.100000000000002	234.3	7.4	7.9	7.7	0.2068	
ASEGOR		10/2013-12-21	02:22:21	0	234.8	236.100000000000002	234.3	7.4	7.9	7.7	0.2068	
ASEGOR		10/2013-12-21	02:23:12	0	234.8	236.200000000000002	234.4	7.4	7.9	7.800000000000001	0.2069	
ASEGOR		10/2013-12-21	02:23:14	0	234.8	236.200000000000002	234.4	7.4	7.9	7.800000000000001	0.2069	
ASEGOR		10/2013-12-21	02:23:17	0	234.9	236.3	234.5	7.300000000000001	8	7.800000000000001	0.2068	
ASEGOR		10/2013-12-21	02:23:18	0	234.9	236.3	234.5	7.300000000000001	8	7.800000000000001	0.2068	
ASEGOR		10/2013-12-21	02:23:23	0	235.100000000000002	236.5	234.8	7.300000000000001	8	7.9	0.20600000000000002	
ASEGOR		10/2013-12-21	02:23:26	0	235.100000000000002	236.5	234.8	7.300000000000001	8	7.9	0.20600000000000002	
ASEGOR		10/2013-12-21	02:23:30	0	235.100000000000002	236.600000000000002	234.9	7.300000000000001	8	7.6000000000000005	0.20600000000000002	
ASEGOR		10/2013-12-21	02:23:34	0	235.100000000000002	236.600000000000002	234.9	7.300000000000001	8	7.6000000000000005	0.20600000000000002	
ASEGOR		10/2013-12-21	02:24:14	0	235.200000000000002	236.600000000000002	235	7.300000000000001	8	7.6000000000000005	0.2066	
ASEGOR		10/2013-12-21	02:24:15	0	235.200000000000002	236.600000000000002	235	7.300000000000001	8	7.6000000000000005	0.2066	
ASEGOR		10/2013-12-21	02:24:18	0	234.9	236.3	234.9	2.300000000000003	4.4	0.30460000000000004	0.2064	
ASEGOR		10/2013-12-21	02:24:25	0	235.200000000000002	236.600000000000002	235.200000000000002	7.300000000000001	8.1	7.6000000000000005	0.2064	
ASEGOR		10/2013-12-21	02:24:26	0	235.200000000000002	236.600000000000002	235.200000000000002	7.300000000000001	8.1	7.6000000000000005	0.2064	
ASEGOR		10/2013-12-21	02:25:13	0	236.4	236.8	235.3	7.300000000000001	7.9	7.6000000000000005	0.2064	
ASEGOR		10/2013-12-21	02:25:14	0	236.4	236.8	235.3	7.300000000000001	7.9	7.6000000000000005	0.2064	

Figura 47. Fichero Excel de salida parte 1

Podemos ver los datos de la clave de la tabla, junto al resto de campos de detalle que nos interesa para la medida.

V2	V3	I1	I2	I3	FP1	FP2	FP3	REACT 1	TimeStamp
236.200000000000002	234.4	7.4	7.800000000000001	7.7	0.2071	0.2087	0.2393	59698	2014-04-02 19:05:11.0
236.200000000000002	234.4	7.4	7.800000000000001	7.7	0.2071	0.2087	0.2393	59698	2014-04-02 19:14:09.0
236.100000000000002	234.3	7.4	7.9	7.7	0.2068	0.30860000000000004	0.2393	59698	2014-04-02 19:14:09.0
236.100000000000002	234.3	7.4	7.9	7.7	0.2068	0.30860000000000004	0.2393	59698	2014-04-02 19:05:11.0
236.200000000000002	234.4	7.4	7.9	7.800000000000001	0.2069	0.309	0.23870000000000002	59698	2014-04-02 19:05:11.0
236.200000000000002	234.4	7.4	7.9	7.800000000000001	0.2069	0.309	0.23870000000000002	59698	2014-04-02 19:14:09.0
236.3	234.5	7.300000000000001	8	7.800000000000001	0.2068	0.3108	0.2384	59698	2014-04-02 19:14:09.0
236.3	234.5	7.300000000000001	8	7.800000000000001	0.2068	0.3108	0.2384	59698	2014-04-02 19:05:11.0
236.5	234.8	7.300000000000001	8	7.9	0.20600000000000002	0.2101	0.2379	59698	2014-04-02 19:14:09.0
236.5	234.8	7.300000000000001	8	7.9	0.20600000000000002	0.2101	0.2379	59698	2014-04-02 19:05:11.0
236.600000000000002	234.9	7.300000000000001	8	0.20600000000000005	0.20600000000000002	0.30970000000000003	0.2369	59698	2014-04-02 19:14:09.0
236.600000000000002	234.9	7.300000000000001	8	7.6000000000000005	0.20600000000000002	0.30970000000000003	0.2369	59698	2014-04-02 19:05:11.0
236.600000000000002	235	7.300000000000001	8	7.6000000000000005	0.2066	0.3098	0.2358	59698	2014-04-02 19:05:11.0
236.600000000000002	235	7.300000000000001	8	7.6000000000000005	0.2066	0.3098	0.2358	59698	2014-04-02 19:14:09.0
236.3	234.9	2.300000000000003	4.4	0.30460000000000004	0.304	0.09960000000000001	59698	2014-04-02 19:05:11.0	
236.600000000000002	235.200000000000002	7.300000000000001	8.1	7.6000000000000005	0.2064	0.3079	0.23600000000000002	59698	2014-04-02 19:05:11.0
236.600000000000002	235.200000000000002	7.300000000000001	8.1	7.6000000000000005	0.2064	0.3079	0.23600000000000002	59698	2014-04-02 19:14:09.0
236.8	236.3	7.300000000000001	7.9	7.6000000000000005	0.2064	0.30870000000000003	0.23620000000000002	59698	2014-04-02 19:14:09.0
236.8	236.3	7.300000000000001	7.9	7.6000000000000005	0.2064	0.30870000000000003	0.23620000000000002	59698	2014-04-02 19:05:11.0

Figura 48. Fichero Excel de salida parte 2

Resto de datos de interés.

4.3.2 Esquemas jerárquicos de los módulos

En este punto podemos ver a nivel genérico nuestros métodos, las clases por las que están compuestos.

Aquí ponemos una captura de los mismos y los iremos viendo en profundidad en los siguientes puntos.

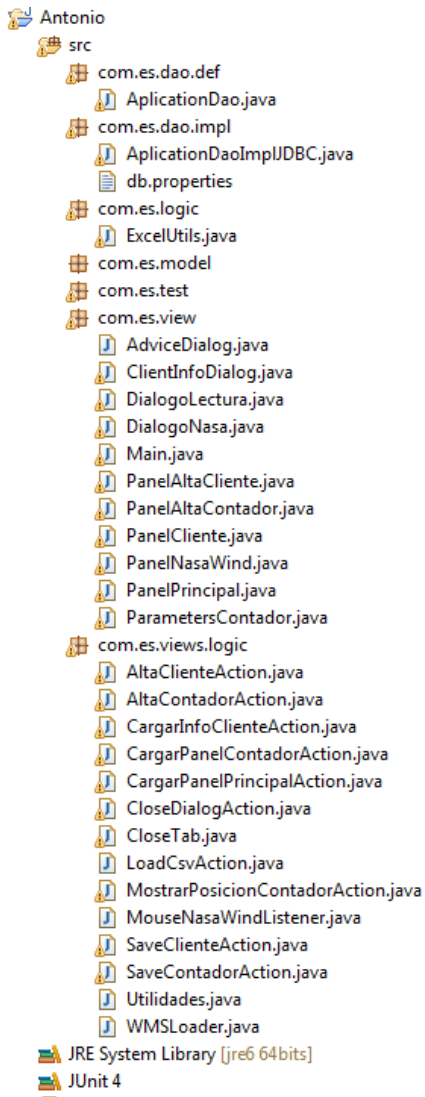


Figura 49. Los métodos de nuestra aplicación dentro de sus clases

El package test es simplemente una clase “lanzadora” de clases de test, desde aquí hemos podido probar inicialmente todos nuestros módulos por separado y les hemos realizado pruebas unitarias, proceso que hemos realizado previamente a una prueba integrada de la aplicación.

4.3.3 Clase ApplicationDao

Una de las clases más importantes de nuestro sistema sería la Application Data Object, realmente es un interfaz que está preparado para ejecutar la clase en función del conector, a día de hoy solo tenemos el JDBC para conectar con la base de datos, pero ya tenemos preparado un sistema escalable desde este punto de vista.

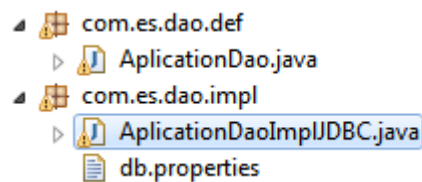


Figura 50. Interfaz y clase que implementa la interfaz.

El fichero db.properties contiene todo lo necesario para la conexión con nuestra base de datos, el perfil que nos da permisos para el acceso. La idea a futuro es tener una tabla usuarios que nos de diversos perfiles, con ellos podremos hacer o no cierto tipo de consultas.

Esta es la clase que trabaja con los objetos a nivel de BBDD, trabaja con objetos de tipo cliente y de tipo contador principalmente, también definimos un objeto “Data” en el cual volcamos toda la información que se registra en la tabla de consulta histórico.

```
public interface AplicationDao {  
    public void insertCliente(Cliente cliente);  
    public Cliente getCliente(String idCliente);  
    public List<Cliente> getClients();  
    public void insertContador(Contador contador);  
    public Contador getContadorByClienteId(int idContador,String idCliente);  
    public List<Contador> getContadoresByClienteId(String idCliente);  
    public void insertHistoricao(Map data);  
    public Data selectHistorico(String cliente, int numcontador , String fechaIni, String fechaFin);  
}
```

Figura 51. Clase Application Data Object

Podemos ver los diferentes tipos de métodos que tenemos de inserción y consulta, en cada uno de ellos enviamos o recogemos la información necesaria, a destacar el objeto Data, en el guardamos todos los datos necesarios del cliente relevantes para un contador específico. El array fileData contiene todo el array de datos leídos el contador, en formato hexadecimal.

The screenshot shows an IDE window with a project explorer on the left and a code editor on the right. The project explorer shows a package 'com.es.test' containing several Java files: Cliente.java, Contador.java, Data.java (selected), AllTests.java, Test.java, and TestConsultaBBDD.java. The code editor displays the following code for Data.java:

```

public class Data {

    String cliente;
    int contador;
    @SuppressWarnings("rawtypes")
    List<Map> fileData= new ArrayList<Map>();
    String date;
    String hour;

    public String getCliente() {
        return cliente;
    }
    public void setCliente(String cliente) {
        this.cliente = cliente;
    }
    public int getContador() {
        return contador;
    }
    public void setContador(int contador) {
        this.contador = contador;
    }

    @SuppressWarnings("rawtypes")
    public List<Map> getFileData() {
        return fileData;
    }
}

```

Figura 52. Objeto Data

```

public Data readExcelData(String fileIn,String cliente,int contador) throws IOException{
    Data data=new Data();
    archivo = new File(fileIn);
    fr = new FileReader (archivo);
    br = new BufferedReader(fr);
    line= null ;
    String hora_ant = "00:00:00" ;
    String hora_act = null ;
    while ((line = br.readLine()) != null) {
        Map b = new HashMap();
        int y = line.length()/3;
        String [] sCadena = new String[y];
        String fila = line.intern();
        Timestamp Timestamp = new java.sql.Timestamp(System.currentTimeMillis());
        for(int i=0; i< y ;i++){
            int a = i*3;
            sCadena[i] = fila.substring(a, a+2);
        }
        b.put(0, cliente);
        b.put(1, contador);
        //verificamos que la fecha y la hora sean distintas. En caso de ser iguales hemos de aumentar los segundos(se insertará un reloj para poder hacer esto apropiadamente)
        b.put(2,returndate(sCadena));
        hora_act = returnhour(sCadena, hora_ant);
        hora_ant = hora_act ;
        b.put(3, hora_act);
        b.put(4,activaimport(sCadena));
        b.put(5,activaexport(sCadena));
        b.put(6,v1(sCadena));
        b.put(7,v2(sCadena));
        b.put(8,v3(sCadena));
        b.put(9,i1(sCadena));
        b.put(10,i2(sCadena));
        b.put(11,i3(sCadena));
        b.put(12,fpf1(sCadena));
        b.put(13,fpf2(sCadena));
        b.put(14,fpf3(sCadena));
        b.put(15,reactiva(sCadena));
        b.put(16, Timestamp);
        data.getFileData().add(b);
    }
    br.close();
    return data;
}

```

Figura 53. Lectura del objeto data.

Estos son los objetos y métodos básicos con los que manejamos la información en nuestra aplicación.

4.3.3.1 Métodos de inserción y consulta

Los métodos de consulta, carga y modificación de BBDD consideramos que no tienen una relevancia significativa ya que son instrucciones básicas eminentemente. Simplemente resaltar un par de puntos, en las query hemos utilizado preparedStatement para que sea el optimizador de base de datos en tiempo de BIND quien decida en función de las variables cual debe ser el camino para acceder a datos y en qué página del índice ya que cuando trabajamos en nuestro modelo con los datos

de uno u otro contador estamos ejecutando las mismas query con muy pocos datos clave diferentes.

El fichero db.properties contiene los datos de acceso a la BBDD, esto en principio solo va a ser utilizado siempre desde el mismo HOST, se queda adaptado a que en un futuro haya permisos de acceso por usuario a la BBDD.

```
dbName=jdbc:mysql://localhost/elect?
dbUser=root
pw=duncan
adress=localhost
port=3306
```

Figura 54. Contenido del fichero de propiedades

4.3.4 Clase de generación de Excel

El método principal de la aplicación sería el siguiente,

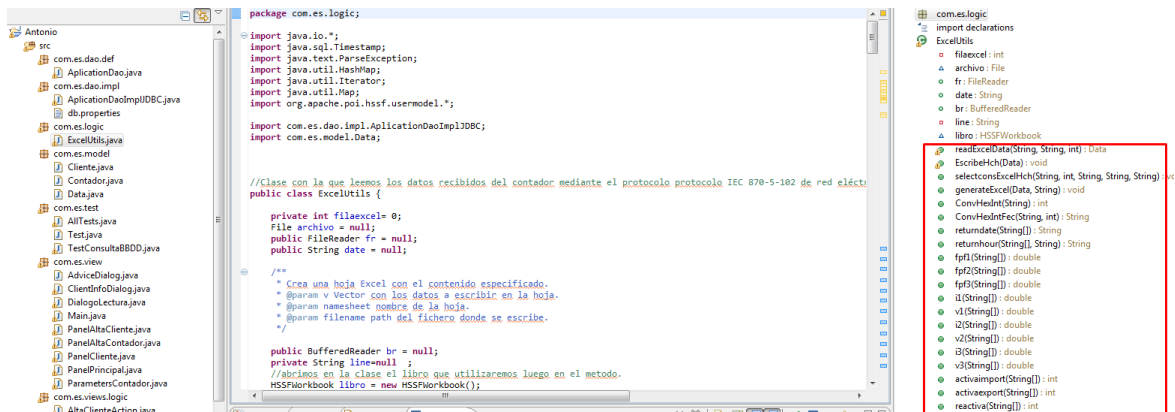


Figura 55. Clase ExcelUtils que genera la Excel de salida

Por no extendernos en exceso simplemente comentaremos los aspectos que consideramos más relevantes de esta clase, sus métodos básicamente que no se refieren a cálculos, ya que los referidos a la obtención de valores son meramente implementaciones del protocolo a nivel de código.


```

public Data readExcelData(String fileIn,String cliente,int contador) throws IOException{}
public void EscribeHch( Data data ) {}
public void selectconsExcelHch(String cliente, int contador, String fechaIni, String fechaFin, String nombre) throws ParseException {}
public void generateExcel(Data data,String output){}
public int ConvHexInt (String aux){}
public String ConvHexIntFec (String aux, int inicad ){}

public String returndate (String[] sCadena){}

public String returnhour (String[] sCadena, String hora_ant){}
public double fpf1 (String[] aux){}
public double fpf2 (String[] aux){}
public double fpf3 (String[] aux){}
public double i1 (String[] aux){}
public double v1 (String[] aux){}
public double i2 (String[] aux){}
public double v2 (String[] aux){}
public double i3 (String[] aux){}
public double v3 (String[] aux){}
public int activaimport (String[] aux){}
public int activaexport (String[] aux){}
public int reactiva (String[] aux){}

```

Figura 56. Métodos de la clase ExcelUtils

Podemos ver dos métodos que nos consiguen la información, serían los siguientes,

- readExcelData → Nos permite leer el fichero plano en hexadecimal que viene del contador y guardar todo el contenido en el objeto data.
- selectconsExcelHch → Nos lanza una consulta al histórico con los datos necesarios para la consulta del contador del cliente y con ellos nos genera la Excel.

```

public void selectconsExcelHch(String cliente, int contador, String fechaIni, String fechaFin, String nombre) throws ParseException {
    Data data = new Data();
    ApplicationDaoImplJDBC Dao = new ApplicationDaoImplJDBC();
    if (fechaIni.isEmpty()) {fechaIni = null;};
    //if (fechaIni.length() == 0) {fechaIni = null;};
    if (fechaFin.isEmpty()) {fechaFin = null;};
    //if (fechaFin.length() == 0) {fechaFin = null;};
    data = Dao.selectHistorico(cliente, contador, fechaIni, fechaFin);
    generateExcel(data, nombre);
}

```

Figura 57. Consulta desde BBDD para generación de la salida.

El método generateExcel nos generará las cabeceras necesarias para que se muestren los datos, todo esto en un formato agradable en el que se aplicación ciertos criterios visuales y de estilos. Mediante el objeto Data y el resto de métodos nos darán la información en el formato requerido, como por ejemplo ConvHexInt que nos convierte hexadecimal a entero o el método returndate que nos devuelve la fecha leída en hexadecimal a formato fecha, para ello hemos de procesar ciertas peculiaridades en función del número de bits binarios en función de las tuplas.

4.3.4.1 Método escritura en el histórico.

Es básicamente un bucle en función de las ocurrencias del archivo Data, este por cada una de ellas abrirá los datos que hay en su interior, finalmente llamara a la clase de ApplicationDao al método db2.inserthistorico,

```
public void EscribeHch( Data data ) {  
    AplicacionDaoImplJDBC db2 = new AplicacionDaoImplJDBC();  
  
    for (Iterator iterator = data.getFileData().iterator(); iterator.hasNext();) {  
        Map datos = (Map) iterator.next();  
        db2.insertHistorico(datos);  
    }  
}
```

Figura 58. Método de escritura del histórico.

Utilizamos esta clase ya que maneja el objeto Data, una vez separado en datos iterados, podremos manejar esta información con la clase de inserción en la base de datos.

4.3.4.2 Método de retorno del tiempo

Como se ha comentado anteriormente en la lectura instantánea el contador no nos da los segundo y milisegundos de esta lectura, en caso de querer estos con detalle necesitaríamos un reloj en nuestra placa de hardware que tendría que estar perfectamente sincronizado con el contador, esto se evaluó y se desestimó ya que no era un factor tan importante a la hora de almacenar el dato ya que no se nos permite más de 3-4 lecturas por minuto al contador(aproximadamente se producen una cada 15-17 segundos.) , para poder almacenar varias lecturas durante el mismo minuto

necesitábamos que tuvieran segundos diferentes ya que formaban parte de la Primary Key de la tabla y por ello consensuamos en tomar esta decisión por ahora.

El código se ha incluido en el apartado 4.1.5.

4.3.4.3 Resto de métodos

Como se ha comentado anteriormente con el estudio del ASDU la empresa AYF no quiere que mostremos en un documento público lo que se deduce del estudio de este protocolo debido a su complejidad y el trabajo que ha llevado adaptar muchas de estas transformaciones de las cadenas a un formato correcto y por lo tanto no me ha dado su consentimiento expreso para introducirlo.

4.3.5 Clase NasaWorldWind

Esta clase WMSLoader contiene toda la información necesaria para generar un objeto WorldWindowGLCanvas, sobre el cual trabajamos según las posiciones de latitud y longitud en radianes mostradas desde el navegador, estas las insertaremos en la base de datos para posteriormente consultarlas y que el objeto nos muestre su posición en el mapa.

Hemos barajado diferentes webs de mapas para trabajar con ellas,

```
//          URI serverURI = new URI("http://www.ideo.es/wms/PNOA-  
MR/PNOA-MR?REQUEST=GetCapabilities&SERVICE=WMS&VERSION=1.1.1");  
//          URI serverURI = new  
URI("http://www.ideo.es/wms/PNOA/PNOA");  
//          URI serverURI = new URI("http://wms.jpl.nasa.gov/wms.cgi");  
//          URI serverURI = new  
URI("http://wms.onterrasystems.com/WMSservice.svc/5e883f14f72e45e985f5770b1c629  
39f/WMSLatLon");  
//          URI serverURI = new URI("http://129.206.228.72/cached/osm-  
wms");  
//          URI serverURI = new  
URI("http://129.206.228.72/cached/hillshade");  
//          URI serverURI = new  
URI("http://vmap0.tiles.osgeo.org/wms/vmap0");
```

También fue necesario crear una clase de utilidades para los cálculos que nos requería esta aplicación,

```
public class Utilidades {  
  
    static Logger logger=Logger.getLogger(Utilidades.class);  
  
    public static double interpolar(double x1,double x2, double y1,double y2,double y0){  
        double temp1= (x2-x1)/(y2-y1);  
        double temp2= (y0-y1);  
        double x=x1 +temp1*temp2;  
        // logger.info("valor interpolado= " + x);  
        return x;  
    }  
  
    public static double gradosToRadianes(double grados, double minutos,double segundos){  
        double resultado=(grados+(minutos/60)+(segundos/3600))*(  
        double minutos - com.es.views.logic.Utilidades.gradosToRadianes(double, doubl  
        return resultado;  
    }  
  
    public static String radianesToGrados(double radianes, boolean isLongitude){  
        String resultado ="";  
        return resultado;  
    }  
}
```

Figura 59. Clase utilidades con los cálculos necesarios para procesar los datos entre radianes y grados.

En la imagen que sigue a continuación, podemos observar como con un action de clic de ratón en un punto determinado de la pantalla, generará un evento,

```

public MouseNasaWindowListener(JTextField text, JTextField textField_lat, JTextField textField_long_gra, JTextField textField_lat_gra) {
    super();
    this.text=text;
    this.textField_lat=textField_lat;
    this.textField_long_gra=textField_long_gra;
    this.textField_lat_gra=textField_lat_gra;
}

@Override
public void mouseClicked(MouseEvent e) {
    WorldWindowGLCanvas canvas=null;

    if(e.getButton()==3){
        canvas=(WorldWindowGLCanvas) e.getSource();
        PickedObjectList list=canvas.getObjectsAtCurrentPosition();
        PickedObject picked=list.get(0);

        @SuppressWarnings("rawtypes")
        List listLat=Utilidades.radiansToGrados(picked.getPosition().getLatitude().radians);
        @SuppressWarnings("rawtypes")
        List listLon=Utilidades.radiansToGrados(picked.getPosition().getLongitude().radians);
        double seclat=Math.abs(Utilidades.redondear(Double.parseDouble(listLat.get(2)+""),4));
        double seclon=Math.abs(Utilidades.redondear(Double.parseDouble(listLon.get(2)+""),4));
        String sentidoLat="";
        String sentidoLon="";
        if(Double.parseDouble(listLat.get(0)+"")>0){
            sentidoLat="N";
        }
        else{
            sentidoLat="S";
        }
        if(Double.parseDouble(listLon.get(0)+"")>0){
            sentidoLon="E";
        }
        else{
            sentidoLon="W";
        }

        String lat= Math.abs(Integer.parseInt(listLat.get(0)+""))+"°S " + Math.abs(Integer.parseInt(listLat.get(1)+""))+" ' " + seclat+"'''' + sentidoLat;
        String lon= Math.abs(Integer.parseInt(listLon.get(0)+""))+"°S " + Math.abs(Integer.parseInt(listLon.get(1)+""))+" ' " + seclon+"'''' + sentidoLon;

        this.text.setText(lon);
        this.textField_lat.setText(lat);
        this.textField_long_gra.setText(picked.getPosition().getLongitude().toString());
        this.textField_lat_gra.setText(picked.getPosition().getLatitude().toString());
    }
}

```

Figura 60. Método que nos devuelve la latitud y longitud de una posición del canvas.

Este método se lanza al activar el botón del ratón, nos dará en grados y radianes las posiciones de un punto en el mapa, con estos datos podremos luego registrar la ubicación del contador mediante su latitud y longitud.

```

public DialogoNasa(String latitud, String longitud) throws InterruptedException, InvocationTargetException {

    this.latitud=latitud;
    this.longitud=longitud;
    setTitle("Posicion contador");
    setBounds(100, 100, 960, 760);
    getContentPane().setLayout(new BorderLayout());
    contentPanel.setLayout(new FlowLayout());
    contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
    getContentPane().add(contentPanel, BorderLayout.CENTER);

    JPanel buttonPane = new JPanel();

    lblLatitud = new JLabel("Latitud");
    buttonPane.add(lblLatitud);

    textField_lat = new JTextField();
    buttonPane.add(textField_lat);
    textField_lat.setColumns(11);

    lblLatitudRadianes = new JLabel("Latitud radianes");
    buttonPane.add(lblLatitudRadianes);

    textField_lat_gra = new JTextField();
    textField_lat_gra.setColumns(13);
    buttonPane.add(textField_lat_gra);

    lblLongitud = new JLabel("Longitud");
    buttonPane.add(lblLongitud);

    textField = new JTextField();
    buttonPane.add(textField);
    textField.setColumns(11);

    lblLongitudRadianes = new JLabel("Longitud radianes");
    buttonPane.add(lblLongitudRadianes);

    textField_long_gra = new JTextField();
    textField_long_gra.setColumns(13);
    buttonPane.add(textField_long_gra);

    textField_lat_gra = new JTextField();
    textField_lat_gra.setColumns(13);
    buttonPane.add(textField_lat_gra);

    lblLongitud = new JLabel("Longitud");
    buttonPane.add(lblLongitud);

    textField = new JTextField();
    buttonPane.add(textField);
    textField.setColumns(11);

    lblLongitudRadianes = new JLabel("Longitud radianes");
    buttonPane.add(lblLongitudRadianes);

    textField_long_gra = new JTextField();
    textField_long_gra.setColumns(13);
    buttonPane.add(textField_long_gra);

    //inicializamos valores
    if (latitud == null ) {textField.setText("38.3f");}
    if (longitud == null ) {textField.setText("-0.56f");}

    PanelNasaWind panel= new PanelNasaWind(latitud , longitud , textField , textField_lat , textField_long_gra ,textField_lat_gra);

    getContentPane().add(panel.getWorldWindCanvas());
    getContentPane().add(buttonPane, BorderLayout.PAGE_END);
}

```

Figura 61. El panel de dialogoNasa nos muestra el mapa con la información del WorldWindCanvas.

Esta clase es muy potente y nos permite muchas utilidades, el resultado visual es magnífico y da una imagen de consistencia a la aplicación. A día de hoy estamos utilizando la posición del contador, pero también podremos en un futuro utilizarlo en más opciones.

5 RESULTADOS EN UN CLIENTE

Realizamos la captura y aplicación de los resultados del análisis sobre los datos de un cliente real, para ello la empresa utiliza el software creado y evaluamos las conclusiones.

5.1 DATOS PREVIOS DEL CLIENTE

Como ejemplo del resultado de la aplicación del Proyecto, se tomarán los datos conseguidos de un suministro de cliente determinado. El suministro en cuestión está dedicado principalmente a cámaras frigoríficas, siendo un total de 6. Lógicamente también tiene otros consumos, como oficinas, alarmas, iluminación, etc...

En cuanto al reparto de potencias, las cámaras frigoríficas están necesitando hasta la fecha una potencia superior a 200kW mientras que el resto de suministros registra una media de 20kW.

Las cámaras frigoríficas se encuentran en perfecto estado de mantenimiento, por lo que se suponen unas pérdidas (muy conservadoras para el lado de la seguridad) de 1°C/hora en caso de no apagar las máquinas. Cabe destacar que en caso de no estar bien aisladas las cámaras las medidas que a continuación se describirán no tendrían cabida ya que las pérdidas las convertirían en no rentables.

A consecuencia de los aspectos descritos en los dos párrafos anteriores, las medidas que se describirán en este punto (ya que consideramos que son las que tienen cabida dentro del marco de este PFC) son las relativas a las cámaras frigoríficas. El punto de partida para las dichas medidas será la curva de potencias cada 15 segundos del circuito de fuerza que las alimenta, las registraremos mediante nuestro software en la Base de datos creada para ello. Los valores reales de dicho circuito en el día anterior a la puesta en marcha de las medidas pueden consultarse en el 7.4. Dichos valores corresponden al mes de julio, con una temperatura media durante el periodo muestreado de 28°C.

Es este Anexo se puede apreciar que la energía total consumida en este periodo fue de 1.660,325kWh, distribuidos en 683,37 en el periodo punta y el resto en el periodo valle.

En este periodo se detectaron una serie de carencias, tras un análisis de los datos se llegaron a una serie de conclusiones que se describen en el siguiente punto.

5.2 APLICACIÓN DE NUESTRO SISTEMA

Una vez extraídas las conclusiones del análisis de los datos que teníamos reflejados en la Base de Datos, se decidió qué medidas tomar y como aplicarlas. Dadas las características de los productos almacenados, y por supuesto con el visto bueno de la Propiedad, se procedió a ejecutar las siguientes medidas:

- Se escalonaron los encendidos de las máquinas de manera que no se encendiese más de una cada vez.
Esta medida afecta a la potencia demandada y por tanto contratada, no así a la energía necesaria ya que, si bien durante el tiempo en que permanece apagada la máquina no se pierde mucha temperatura ha de recuperarse luego.
Los intervalos de apagado y encendido se programaron de manera que el punto de funcionamiento de las máquinas se encontrasen cerca del punto de rendimiento óptimo de las máquinas. Estos cálculos quedan fuera del alcance del presente PFC.
- Se redujo el número de encendidos el periodo punta, siguiéndose el criterio del apartado anterior de no alejar las máquinas de su punto de rendimiento óptimo.
Con ello lógicamente el consumo una vez pasado el periodo punta se incrementa durante un tiempo, requiriéndose en éste un consumo energético mayor.

La curva de carga resultante se puede consultar en el 7.5.

El resultado de estas medidas se puede apreciar en los siguientes puntos:

- Menores pérdidas por disipación de calor en los circuitos. Al pasar menos intensidad por los conductores se disminuyen las pérdidas por efecto Joule en los mismos. El cálculo de este ahorro se antoja prácticamente imposible de calcular ya que se necesitan muchos más datos de los disponibles.
- Ahorro en el término de potencia. Al no existir la posibilidad de que varias máquinas arranquen a la vez no pueden existir picos de potencia demandada con lo que se pueden disminuir las potencias contratadas.

En el caso particular de las instalaciones que nos ocupan permitiría contratar del orden de 200kW, lo que para una tarifa ATR 3.1A como la del ejemplo mostrado supone un ahorro de (IVA no incluido):

$$\begin{aligned} \text{Ahorro} &= \Delta \text{Potencia} \cdot \text{Precio potencia} \cdot \text{Imp. eléctrico} \\ &= 200 \text{ kW} \cdot 104,03 \frac{\text{€} \cdot \text{año}}{\text{kW}} \cdot (1 + 1,05113 \cdot 0,04864) \\ &= 21.869,75 \text{ €/año} \end{aligned}$$

- Ahorro en el término de energía en punta. En el periodo punta se ha pasado de registrarse un consumo de 976,96kWh a un consumo de 97,56kWh, suponiendo un ahorro en este periodo de 879,36kWh. Atendiendo al precio medio de la energía para el periodo punta y el periodo llano, puesto que el precio en el mercado es variable, se puede estimar un ahorro en este término de:

$$\begin{aligned} \text{Ahorro anual} &= \text{Energía ahorrada día} \cdot \text{días} \cdot \Delta \text{Precio} \cdot \text{Imp. eléctrico} \\ &= 879,36 \frac{\text{kWh}}{\text{día}} \cdot 52 \frac{\text{semanas}}{\text{año}} \cdot 5 \frac{\text{día}}{\text{semana}} \\ &\quad \cdot 0,0099478 \frac{\text{€}}{\text{kWh}} (1 + 1,05113 \cdot 0,04864) = 2.390,68 \text{ €/año} \end{aligned}$$

- Ahorro por mejor compra. Comparando las tablas mostradas en los Anexos 7.4 y 7.5 se puede ver que la varianza de los consumos reduce en un orden 10. Ello implica menores desvíos a la hora de la compra de energía en el mercado eléctrico, lo que supone menores penalizaciones. El cálculo del ahorro

derivado de este punto depende de multitud de factores que escapan al alcance de este PFC, por lo que no se calculará.

5.3 TARIFA DE MANO DE OBRA

Las horas de programación y de análisis aproximadas se detallan seguidamente, el precio aproximado por hora de programación ronda los 10 euros, siendo 15 euros/hora un precio razonable para el análisis técnico realizado, teniendo en cuenta los desplazamientos y visitas a clientes.

Ha sido alrededor de un mes y medio en tareas de análisis y unos cuatro meses a nivel de programación, prueba del software y depuración del sistema.

Teniendo una aproximación de trabajo de unas 40 horas semanales y 4 semanas cada mes, trataremos de hacer un compendio de todo lo invertido.

40 horas semana * 6 semanas * 15 euros / hora = 3600 euros.

40 horas semana * 16 semanas * 10 euros / hora = 6400 euros.

Total = 10000 euros costaría el desarrollo del sistema, las licencias son todas gratuitas, ya que el entorno sobre el que se ha desarrollado es eclipse, Mysql y los XLS se pueden abrir con open office.

5.4 PRESUPUESTO Y VIABILIDAD DEL PFC

Para finalizar este punto de mejoras y resultados en un cliente, vamos a tratar de hacer un estudio de cuanto ahorro neto y beneficio le ha supuesto a AYF Energía el desarrollo de mismo.

El coste por parte de la empresa ha sido gratuito al tratarse de un PFC. Ahora mismo disponen de un sistema que les puede dar la herramienta para trabajar con más clientes de manera autónoma utilizando la misma infraestructura que le proporciona red eléctrica.

Un analizador de redes convencional rondaría los 1500-2500 euros por equipo, que debería quedarse fijo en la empresa o cliente en cuestión, con el riesgo que esto conlleva. Luego estos datos igualmente, deberían quedarse establecidos o guardados en una base de datos con una aplicación, el costo por cliente sería una cifra que habría sido difícil de asumir.

Viendo que el ahorro en un cliente puntual que ronda los 2500 euros anuales, sumado a esto los 2000 euros que costaría un analizador de redes, extrapolamos estos datos a los 15 clientes que tiene actualmente la empresa y estaríamos hablando de un ahorro de:

- 2000 euros de analizador * 15
- 2400 euros de ahorro anual * 15
- 10000 euros de programa de análisis y registro de datos

Estamos hablando de un ahorro fijo de 40000 euros y una parte variable de ahorro de unos 36000 euros anuales.

6 CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

6.1 CONCLUSIONES

El proyecto ha creado de manera eficiente para la empresa AYF un soporte para clientes mediante el análisis y la mejora continua. La realización del sistema de medida del contador evita la necesidad de un analizador de redes aprovechando los protocolos que los mismos contadores de red eléctrica proporcionan.

Finalmente se ha construido una base sobre la que trabajar y dar forma a todos los datos extraídos en forma de aplicación, esta nos aportará y calculará fácilmente los datos del cliente por cada uno de sus contadores. Será la base de desarrollos futuros e irá creciendo hasta conseguir un método autónomo con el cual se cree un procedimiento automático que calcule la demanda que deba de tener el cliente en el mercado intradiario optimizando así su consumo energético. Una vez conseguido este objetivo, el siguiente paso será automatizar el inicio de algunos de los sistemas más costosos de los clientes y que se ejecuten en los periodos 0, en los que no hay coste por el uso de esta energía.

Como dijo Lord Kelvin, *Lo que no se define no se puede medir. Lo que no se mide, no se puede mejorar. Lo que no se mejora, se degrada siempre.*

Esta es la idea base para la búsqueda de esta mejora continua que se desea obtener.

6.2 MEJORAS DE SOFTWARE

En este punto comentaremos posibles ampliaciones software de la aplicación, algunas de ellas ya están en curso y otras estarían bien que se implementaran para seguir automatizando el sistema.

6.2.1 Mejoras de procesos

Desarrollaremos los posibles puntos de mejora a nivel de procesos que tenemos en mente para mejorar la aplicación.

6.2.1.1 Proceso nocturno de recálculo de medias de clientes y contadores.

Cada cliente y cada contador en su definición de la tabla tienen unas medias diarias, nocturnas, etc... Con las que hemos de trabajar, estas han de ser recalculadas cada noche con un proceso nocturno que según unas reglas han de recalcular estos valores para tenerlos actualizados diariamente y reflejando la realidad de estos valores. Crear esto dentro de la aplicación actual es muy pesado y debe tenerse un proceso a parte.

6.2.1.2 Proceso nocturno de carga de singularidades en el modelo de datos

Dentro de las medidas que se carguen a lo largo de un día en todos los clientes pueden haber datos “extraños” o que sean especialmente relevantes, estos datos han de cargarse en una tabla especialmente creada en el modelo para esto, hablamos de la tabla singularidades que tiene como clave primaria un Id de singularidad junto con la clave de la tabla de la que se nutre que es la tabla histórico de contadores, estos datos han de estar en esta tabla reflejados para que se puedan hacer pequeños análisis de estos datos cuando se requieran y dejarlos reflejados en un comentario que tiene la tabla.

6.2.1.3 Mejora del proceso de extracción de variables

Optimizar la extracción de datos de las tramas, para así solo procesar las variables que son realmente utilizables por el sistema. Para ello habrá de haber corrido la aplicación un tiempo y ver qué datos merecen la pena registrar y cuáles no. A día de hoy se están registrando todos y conlleva más memoria de la habitual.

6.2.2 Mejoras de la BBDD

Se podrían incluir triggers de la BBDD, cuando hagamos borrado de clientes/contadores para así actualizar los datos de un cliente con los contadores que le quedaran activos.

6.2.2.1 Perfiles de acceso. Usuarios, súper-usuarios e invitados.

En un futuro, cuando haya definidas más operaciones sobre la BBDD y se pueda acceder vía web a una aplicación o desde varios dispositivos a la misma base de datos necesitaremos crear perfiles de acceso para poder delimitar el tipo de operaciones que crea cada tipo de usuario en nuestro sistema, como limitar cargas o tipos de consultas, dar prioridades, etc...

6.2.3 Mejora de los mapas

6.2.3.1 Encontrar mapas con nombres de calles actualizado.

Estuvimos probando diversos proveedores de mapas, pero es complicado tener un proveedor de mapas por capas gratuito que ofreciera los nombres de las calles y diera un buen rendimiento, el actual permite un buen filtrado de acercamiento alejamiento de la imagen pero no aparece el nombre de la calle cosa que no preocupa demasiado ya que está guardado en el registro del contador pero ha de reconocerse que sería interesante conseguir esta opción.

6.2.3.2 Más información registrada.

Es posible tener también fácilmente ubicadas las diferentes sedes del cliente, no solo sus contadores, posiblemente actualmente estén relacionados, pero es una opción a tener en cuenta para un futuro. Como tener previsiones de citas a contadores o reparaciones/instalaciones en ciertos puntos concretos.

6.3 MEJORAS DE HARDWARE

6.3.1 Medidas remotas mediante un modem GSM/4g

La comodidad de no tener que desplazarse o de que pueda borrarse por una sobrecarga la memoria de datos es un problema real y que se querría evitar a toda costa una vez tenido el sistema en funcionamiento, actualmente ya se está trabajando en esto para poder desarrollar este sistema de tele medida de contadores, en un futuro esta misma información puede ser volcada incluso directamente a la BBDD y así tenerla actualizada constantemente y se podría hasta tener procesos ejecutándose en background que nos alertaran de medidas anómalas o procesos en tiempo real que nos extraigan conclusiones al momento para poder anticipar nuestro trabajo y ser un motivo diferenciador del resto.

Con esta opción nos evitaríamos la necesidad de tener que acudir a todos los puntos en los que tengamos contadores, en ciertos casos de difícil acceso o con ciertas imposibilidades.

6.3.2 Reconocimiento de periodos tarifarios 0.

Otro de los puntos de interés de este PFC es el reconocimiento según los datos de consumo de los procesos que se puedan “deslizar” de la hora en la que se están ejecutando. Se conoce que gran parte de los procesos que ejecutan las empresas han de ser diarios/semanales/mensuales y actualmente se ejecutan sin ningún criterio, si localizamos esos procesos y comparamos las horas de ejecución con el periodo tarifario veremos que en más de un caso estos procesos son lentos y se acaban ejecutando en las “horas cargadas” que son las más caras y las que más repercuten en la tarifa eléctrica, la intención al tener localizados estos procesos es saber cuándo podemos ejecutarlo teniendo nuestras previsiones de consumo y saber con anticipación los momentos más baratos de consumo ejecutarlos en ese momento, incluso procesos mensuales / semanales podemos buscar momentos de factura “0”

en los que sobra energía del mercado eléctrico ya que se produce un exceso y sino habría que desecharlo(casos de mucha producción eólica en días de fuertes vientos), con ello se consigue un fuerte ahorro sustancial.

7 ANEXO - REFERENCIAS BIBLIOGRÁFICAS

7.1 GLOSARIO

[ASDU] (Application Service Data Unit), estándar que nos indica cómo debemos informar las tramas de comunicación, en el anexo 7.1.3 desarrollaremos en profundidad esto.

[AYF] Empresa de optimización energética para la que desarrollamos este proyecto.

[BBDD] Base de datos, aquí se encuentra nuestro modelo entidad-relación con todas las tablas y tipos de datos necesarios para nuestro modelo.

[CM] Concentradores de medida.

[Datasheet] es un documento que resume el funcionamiento y otras características de un componente (por ejemplo, un componente electrónico) o subsistema (por ejemplo, una fuente de alimentación) con el suficiente detalle para ser utilizado por un ingeniero de diseño y diseñar el componente en un sistema.

Comienza típicamente con una página introductoria que describe el resto del documento, seguido por los listados de componentes específicos, con la información adicional sobre la conectividad de los dispositivos. En caso de que haya código fuente relevante a incluir se une cerca del extremo del documento o se separa generalmente en otro archivo.

[EEPROM] o E²PROM son las siglas de Electrically Erasable Programmable Read-Only Memory (ROM programable y borrada eléctricamente). Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente, a diferencia de la EPROM que ha de borrarse mediante un aparato que emite rayos ultravioleta. Son memorias no volátiles.

[FPGA] (del inglés Field Programmable Gate Array) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada 'in situ' mediante un lenguaje de descripción especializado. La lógica programable puede reproducir desde funciones tan sencillas

como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip.

[Google Earth] es un programa informático que muestra un globo virtual que permite visualizar múltiple cartografía, con base en la fotografía satelital. El programa fue creado bajo el nombre de EarthViewer 3D por la compañía Keyhole Inc, financiada por la Agencia Central de Inteligencia. La compañía fue comprada por Google en 2004 absorbiendo el programa. El mapa de Google Earth está compuesto por una superposición de imágenes obtenidas por Imagen satelital, fotografía aérea, información geográfica proveniente de modelos de datos SIG de todo el mundo y modelos creados por ordenador. El programa está disponible en varias licencias, pero la versión gratuita es la más popular, disponible para móviles, tablets y PCs. La primera versión de Google Earth fue lanzada en 2005 y actualmente está disponible en PCs para Windows, Mac y Linux. Google Earth también está disponible como plugin para visualizarse desde el navegador web. En 2013 Google Earth se había convertido en el programa más popular para visualizar cartografía, con más de mil millones de descargas.^{1 2}

Muchos usuarios utilizan la aplicación para añadir sus propios datos, haciéndolos disponibles mediante varias fuentes, tales como el Bulletin Board Systems o blogs. Google Earth es capaz de mostrar diferentes capas de imagen encima de la base y es también un cliente válido para un Web Map Service. Google Earth soporta datos geoespaciales tridimensionales mediante los archivos Keyhole Markup Language o .kml.

[GSM] Global System for Mobile communications se considera, por su velocidad de transmisión y otras características, un estándar de segunda generación (2G). Su extensión a 3G se denomina UMTS y difiere en su mayor velocidad de transmisión, el uso de una arquitectura de red ligeramente distinta y sobre todo en el empleo de diferentes protocolos de radio (W-CDMA).

Tipo de dato informático es un atributo de una parte de los datos que indica al ordenador (y/o al programador) algo sobre la clase de datos sobre los que se va a procesar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar. Tipos de datos comunes son: enteros, números de coma flotante (decimales), cadenas alfanuméricas, fechas, horas, colores, etc.

[HOST] ("anfitrión", en español) es usado en informática para referirse a las computadoras conectadas a una red, que proveen y utilizan servicios de ella. Los usuarios deben utilizar anfitriones para tener acceso a la red. En general, los anfitriones son computadores monousuario o multiusuario que ofrecen servicios de transferencia de archivos, conexión remota, servidores de base de datos, servidores web, etc.

[IA] Inteligencia artificial

[IEC] La Comisión Electrotécnica Internacional (CEI o IEC por sus siglas en inglés, International Electrotechnical Commission) es una organización de normalización en los campos eléctrico, electrónico y tecnologías relacionadas. Numerosas normas se desarrollan conjuntamente con la ISO (normas ISO/IEC).

[Interfaz de programación de aplicaciones (IPA) o API] (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

[JAVA] El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina (JVM) sin importar la arquitectura de la computadora subyacente.

[JDBC] Java Database Connectivity, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice. [Trigger (o disparador)] en una Base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación. Dependiendo de la base de datos, los triggers pueden ser de inserción (INSERT), actualización (UPDATE) o borrado (DELETE). Algunas bases de datos pueden ejecutar triggers al crear, borrar o editar usuarios, tablas, bases de datos u otros objetos.

[MySQL] Gestor de base de datos de Oracle.

[NasaWind] World Wind es un programa que actúa como un globo terráqueo virtual, o globo virtual desarrollado por la NASA para ser usado en ordenadores con Microsoft Windows, además de worldwind, existió un paquete para la distribución Ubuntu. Superpone imágenes de satélites de la NASA y fotografías aéreas del United States Geological Survey (USGS) sobre modelos tridimensionales de la Tierra, y en las últimas versiones, Marte y la Luna.

El usuario puede interactuar con el planeta seleccionado rotándolo y ampliando zonas. Además se pueden superponer topónimos y fronteras, entre otros datos, a las imágenes. El programa también contiene un módulo para visualizar imágenes de otras fuentes en Internet que usen el protocolo del Open Geospatial Consortium Web Map Service. Adicionalmente existen multitudes de ampliaciones para World Wind que aumentan su funcionalidad, como por ejemplo, poder medir distancias u obtener datos de posición desde un GPS.

[OSI] (En inglés, Open System Interconnection 'sistemas de interconexión abiertos') es el modelo de red descriptivo, que fue creado por la Organización Internacional para la Estandarización (ISO) en el año 1980.¹ Es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones.

[PCB] En electrónica, un circuito impreso, tarjeta de circuito impreso o PCB (del inglés printed circuit board), es una superficie constituida por caminos o pistas de material conductor laminadas sobre una base no conductora. El circuito impreso se utiliza para conectar eléctricamente - a través de los caminos conductores, y sostener mecánicamente - por medio de la base, un conjunto de componentes electrónicos. Los caminos son generalmente de cobre mientras que la base se fabrica de resinas de fibra de vidrio reforzada (la más conocida es la FR4), cerámica, plástico, teflón o polímeros como la baquelita.

[PIC] son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instrument. El nombre actual no es un acrónimo. En realidad, el nombre completo es PICmicro, aunque generalmente se utiliza como Peripheral Interface Controller (controlador de interfaz periférico).

[POI] consiste en varias APIs para manipular los varios formatos de documentos Microsoft OLE 2 utilizando para ello una solución java. El soporte para la lectura y escritura de ficheros Excel con java está ya muy maduro y el trabajo continúa para mejorar el soporte de lectura y escritura de ficheros Word en java.

[Puerto serie o puerto serial] es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente.¹ La comparación entre la transmisión en serie y en paralelo se puede explicar usando una analogía con las carreteras. Una carretera tradicional de un sólo carril por sentido sería como la

transmisión en serie y una autovía con varios carriles por sentido sería la transmisión en paralelo, siendo los vehículos los bits que circulan por el cable.

[RAM] memoria de acceso aleatorio (en inglés: random-access memory) se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayor parte del software. Es allí donde se cargan todas las instrucciones que ejecutan el procesador y otras unidades de cómputo. Se denominan «de acceso aleatorio» porque se puede leer o escribir en una posición de memoria con un tiempo de espera igual para cualquier posición, no siendo necesario seguir un orden para acceder a la información de la manera más rápida posible. Durante el encendido del computador, la rutina POST verifica que los módulos de memoria RAM estén conectados de manera correcta. En el caso que no existan o no se detecten los módulos, la mayoría de tarjetas madres emiten una serie de pitidos que indican la ausencia de memoria principal. Terminado ese proceso, la memoria BIOS puede realizar un test básico sobre la memoria RAM indicando fallos mayores en la misma.

[RM] Registradores de medida eléctrica.

[RS-232] (Recommended Standard 232, también conocido como EIA/TIA RS-232C) es una interfaz que designa una norma para el intercambio de una serie de datos binarios entre un DTE (Equipo terminal de datos) y un DCE (Data Communication Equipment, Equipo de Comunicación de datos).

[RS-485] o también conocido como EIA-485, que lleva el nombre del comité que lo convirtió en estándar en 1983. Es un estándar de comunicaciones en bus de la capa física del Modelo OSI. Está definido como un sistema en bus de transmisión multipunto diferencial, es ideal para transmitir a altas velocidades sobre largas distancias (35 Mbit/s hasta 10 metros y 100 kbit/s en 1200 metros) y a través de canales ruidosos, ya que reduce los ruidos que aparecen en los voltajes producidos en la línea de transmisión. El medio físico de transmisión es un par entrelazado que admite hasta 32 estaciones en 1 solo hilo, con una longitud máxima de 1200 metros operando entre 300 y 19 200 bit/s y la comunicación half-duplex (semiduplex). Soporta 32 transmisiones y 32 receptores. La transmisión diferencial permite múltiples drivers dando la posibilidad de una configuración multipunto. Al tratarse de un estándar bastante abierto permite muchas y muy diferentes configuraciones y utilizaciones.

[Suministro] Los puntos de suministro a los que hace referencia este documento serían los contadores de los cuales disponen los clientes y en los cuales podemos hacer nuestras medidas.

[TTM] Time to market, tiempo que transcurre entre que un producto es concebido y está disponible para la venta.

7.2 PROTOCOLO DE COMUNICACIONES ASDU

En este punto vamos a describir los antecedentes conocidos que nos suministra red eléctrica española y que hemos de seguir para llevar a cabo nuestro sistema de control, bajo estas condiciones y con un profundo estudio del protocolo podemos llegar a extraer mediante el desarrollo de nuestro propio sistema hardware la información en tramas hexadecimales que posteriormente serán procesadas por nuestra aplicación mediante el traspaso de dichas tramas a un soporte físico.

7.2.1 Normativa

Los contadores de energía de cualquier fabricante deben cumplir con la norma IEC 870-5-102, que en España es de obligatorio cumplimiento según las regulaciones (ver anexo) de Red Eléctrica Española, REE. Para la medida remota automatizada el usuario debe instalar un modem GSM y suministrar a las compañías eléctricas el número de teléfono y las direcciones de los contadores y sus claves de acceso. La implementación del protocolo IEC 870-5-102 es inmediata para cualquier programador profesional o habituado a realizar este tipo de actividades. En este documento se intenta explicar de un modo sencillo este protocolo. En cualquier caso y tal como se ha comentado anteriormente los diferentes fabricantes de contadores y las empresas de ingeniería ofertan comercialmente sus programas para realizar estas funciones.

Actualmente nosotros no utilizaremos la medida remota todavía pero como hemos comentado es el propósito futuro

7.2.2 Protocolo de comunicaciones

El protocolo de comunicaciones está especificado detalladamente en los norma IEC 870-5-102:1996 y en el documento de referencia de REE. Se reproducen en este trabajo algunos de los conceptos considerados de especial relevancia con el objeto de facilitar al lector su comprensión y poder interpretar con facilidad los comandos básicos para efectuar las lecturas de los valores de energía (integrados por periodos de tiempo o curvas de carga e Información de tarificación o lecturas de cierres).

El protocolo de comunicaciones es no balanceado o desequilibrado, lo cual quiere decir que hay un equipo primario fijo (maestro) que solicitará información a una o varias estaciones secundarias (esclavos). En nuestro caso el maestro será un ordenador (PC) y los esclavos serán cada uno de los contadores, identificados con una única dirección dentro de una red RS485 podemos ver en la imagen 97 más adelante, en la que se muestra el caso habitual en la que cada contador tiene una salida RS232 y es necesario un convertidor RS232/RS485 por cada contador. El intercambio de información se realiza mediante el método pregunta-respuesta (*request/respond*), pero también soporta envío/respuesta (*Send/reply*) y envío/confirmación (*send/confirm*).

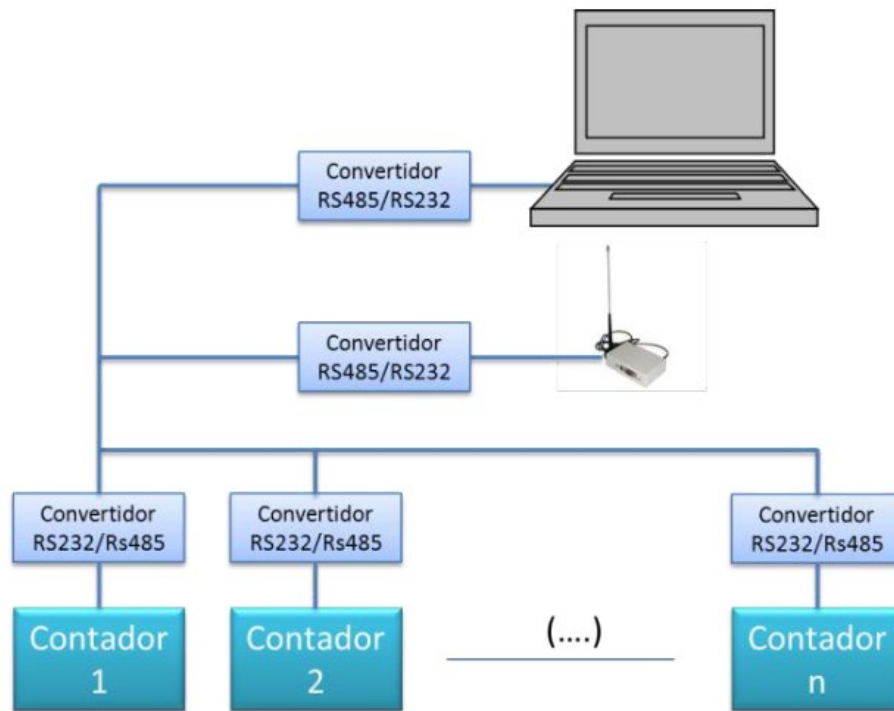


Figura 62. Esquema de conexionado para comunicación con de los contadores de energía

Los comandos (tramas de enlace) pueden ser de longitud fija o de longitud variable, como podemos ver 98 la figura 62. En los comandos de longitud variable la longitud de la trama es configurable mediante parámetros con un máximo de 255 caracteres. Los comandos constan de los siguientes campos:

- Carácter de inicio
- Longitud
- Campo de control
- Dirección

- Datos de aplicación (ASDU)
- Checksum
- Carácter End

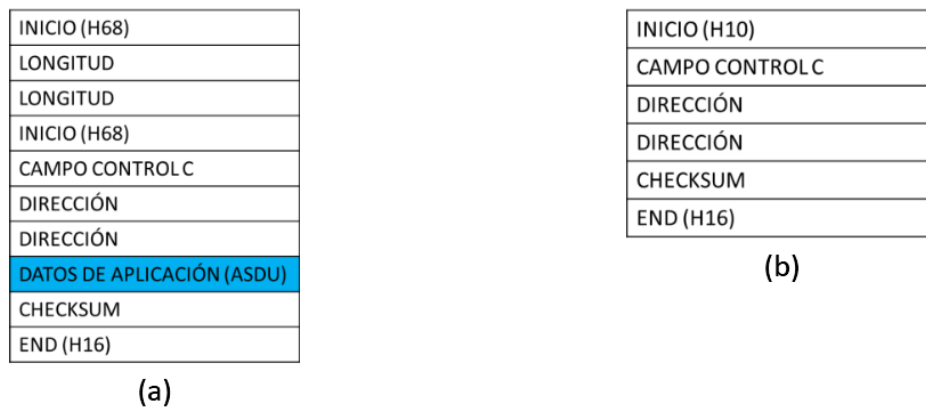


Figura 63. Formato de los comandos (tramas) de (a) longitud variable o (b) de longitud fija.

7.2.2.1 Carácter de inicio

Es el que inicia la trama (1 byte). En las tramas de longitud variable es el byte hexadecimal 68, indicado como 16#68 o H68. Se utiliza dos veces, una para comenzar la trama y otra para indicar el lugar en el que comienzan los comandos de la misma. En las tramas de longitud fija es el byte H10.

7.2.2.2 Longitud

Se envía por duplicado. Son dos bytes repetidos y cada uno indica el número de bytes que se envían en la trama comenzando por el campo de control (incluido) hasta el Checksum (no incluido).

7.2.2.3 Campo de control

El byte (8 bits o 1 octeto) campo de control del enlace, C, de la figura anterior tiene la estructura indicada en la 100 más adelante, en donde:

RES: Reserva (Siempre a 0)

Bite reservado para futuras aplicaciones. Actualmente vale siempre 0.

PRM: Dirección de Control

<0> Mensaje del secundario (que responde)

<1> Mensaje del primario (que inicia)

Expresa el flujo de la información. Si el comando es de maestro a esclavo vale 1, PRM=1, y vale 0, PRM=0, si el comando es de respuesta de esclavo a maestro.

FCB: Es el bit de cuenta de trama (Frame Count Bit).

<0> <1> = bit alternante para sucesivos mensajes de send/confirm o request/respond.

Se utiliza para borrar pérdidas o duplicados de transferencia de información. El maestro alterna el bit FCB para cada nueva transmisión dirigida al mismo contador esclavo. En caso de no alternar este bit, el esclavo entiende que es una repetición del comando recibido con anterioridad. Por tanto el master ha de retener este bit para cambiarlo en cada mensaje a un mismo esclavo. Está relacionado con el bit FCV que lo valida o inválida.

FCV: Habilita el bit FCB.

<0> FCB inhabilitado

<1> FCB habilitado

Campo C	RES	PRM	FCB	FCV	2 ³	2 ²	2 ¹	2 ⁰	Maestro a esclavo
			ACD	DFC	Código de función				Esclavo a maestro
Bit	8	7	6	5	4	3	2	1	

Figura 64. Estructura del campo de control C de la trama en los dos casos de dirección de comunicación, de maestro a esclavo, PRM=1, o de esclavo a maestro, PRM=0.

ACD: Bit de solicitud de acceso. Hay dos clases de datos proporcionados, referenciados como clase 1 y clase 2.

<0> No acceso a datos clase 1

<1> Acceso a datos clase 1

Según el reglamento de REE, únicamente se utilizan datos de clase 2 y por tanto se ignorará el bit de estado ACD.

DFC: Control de flujo de datos.

<0> Se aceptan futuros mensajes

<1> Los mensajes futuros causarían desbordamiento de datos (data overflow)

Bit utilizado por los contadores esclavos para indicar al ordenador maestro que el siguiente mensaje causará desbordamiento de datos.

Los códigos de función aceptados en las tramas enviadas por la estación primaria o maestro (PRM=1) son:

0 Reposición del enlace remoto. Bit FCV de código de control a 0

3 Envío de datos de usuario. Bit FCV de código de control a 1

9 Solicitud de estado del enlace. Bit FCV de código de control a 0

11 Solicitud de datos clase 2. Bit FCV de código de control a 1

Los códigos de función para las tramas enviadas por la estación secundaria o contadores esclavos (PRM=0) son:

- 0 ACK. Reconocimiento positivo.
- 1 NACK. Comando no aceptado.
- 8 Datos de Usuario.
- 9 NACK. Datos solicitados no disponibles.
- 11 Estado del enlace o demanda de acceso.

7.2.2.4 Dirección

Se refiere a la dirección del esclavo a quien va dirigida la trama. La dirección tendrán una longitud de 2 octetos (2 bytes) pudiendo tomar los valores desde 0 (H0000) hasta 65535 (HFFFF), Figura 3. Cada contador tendrá una única dirección de enlace que será distintas entre sí en las configuraciones multipunto.

7.2.2.5 Datos de la aplicación ASDU

Los datos de aplicación (Application Service Data Unit) contiene la información enviada por los maestros o esclavos en los comandos o tramas de longitud variable. De acuerdo con REE se enviará un ASDU por trama. Su estructura se analiza en detalle en el siguiente apartado.

7.2.2.6 Checksum

Es un byte con la suma aritmética de todos los bytes comenzando por el campo de control (incluido) hasta el checksum (no incluido).

7.2.2.7 Carácter END

El carácter END es el que indica el fin del mensaje o trama. Se corresponde con el byte hexadecimal 16H.

7.2.3 Estructura de los datos de aplicación (ASDU)

La estructura general de datos de la aplicación sería la siguiente,

Una ASDU se compone de:

- Un identificador de unidad de datos.
- Uno o más objetos de información.
- Una o ninguna etiqueta de tiempo común.

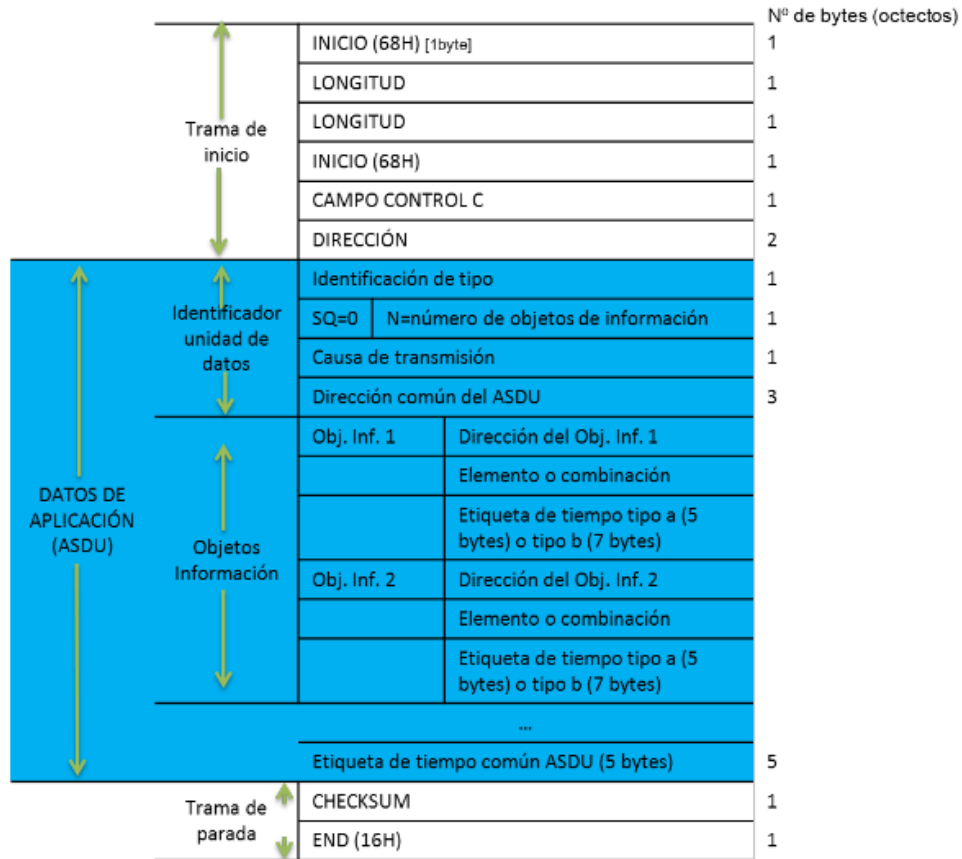


Figura 65. Formato de las tramas de longitud variable, indicando la estructura completa de los datos de aplicación (ASDU).

7.2.3.1 Identificador de unidad de datos

El *identificador de unidad de datos* tiene siempre la misma estructura para todas las ASDU:

- ☐ La *identificación de tipo* (1 byte).
- ☐ Un *calificador de estructura variable* (1 byte).
- ☐ Una *causa de transmisión* (1 byte).

☒ Una *dirección común* del ASDU (3 bytes).

La identificación de tipo es un número de función indicado en la RM=Registadores de medida; CM=Concentradores de medida

En la tabla que podemos ver más adelante, utilizada para indicar el tipo de acción o lectura que se desea realizar. El documento de REE [7] contiene una descripción detallada de cada uno de estos identificadores.

La nomenclatura utilizada en la IEC 870-5 es:

Identificador de tipo :=UI8[1..8]<1..255>

Y se interpreta como que el identificador de tipo de un entero sin signo de 8 bits que puede tomar los valores de 1 a 255 (H01 a HFF en notación hexadecimal). Los valores <1..127> están definidos en dicha norma, dejando los valores <128..255> para uso especial y se especifican en el documento de REE.

Identificadores de tipo		
Id.	Uso	Mnemónico
<1>	Información de evento (single-point) 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 (single-point) 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 de energía 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..225>	Uso libre para cada fabricante	

Figura 66. Identificadores de tipo, referencia.

7.2.3.2 Cualificador de estructura variable

El cualificador de estructura variable contiene información acerca del número de objetos de información enviado en la trama de longitud variable.

En el caso en que el número de objetos de información sea superior a 1, la norma IEC 870-5-102:1996 establece dos tipos de acceso a las direcciones de los objetos de información. La selección del tipo de acceso se realiza mediante el bit nº 8 de este cualificador denominado SQ.

SQ := BS1[8] <0..1>

<0> := Para cada objeto de información se indica su dirección

<1> := Se indica la dirección exclusivamente al primer objeto, siendo las direcciones del resto consecutivas.

El protocolo de REE indica que este bit toma siempre el valor cero, SQ=0. Los bits 7 a 1 indican el número N de objetos de información.

Cualificador de estructura variable := CP8{N,SQ}

N=número de objetos de información := UI7[1..7] <0..127>

SQ:Secuencia := BS1[8] <0..1> (siempre SQ=0).

Bit	8	7	6	5	4	3	2	1
	SQ=0	2 ⁶	N=número de objetos de información		2 ³	2 ²	2 ¹	2 ⁰

Figura 67. Cualificador de estructura variable.

7.2.3.3 Causa de transmisión

El tercer octeto (byte) del identificador de la unidad de datos define la causa de la transmisión. El bit nº 8 indica si es prueba o no (test), el bit nº 7 se refiere a la confirmación positiva o negativa y los bits 6 a 1 contienen el nº de la causa que puede tomar los valores desde 1 a 63, . Se especifica como:

Causa de transmisión :=CP8{Causa,P/N,T}

Causa :=UI6[1..6]<0..63>

P/N :=BS1[7]<0..1>

<0> :=Confirmación positiva

<1> :=Confirmación negativa

T=Test :=BS1[8]<0..1>

<0> :=no test

<1> :=test

Normalmente los bit P/N y T valen 0.

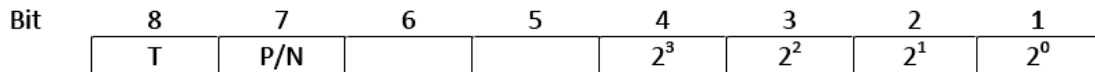


Figura 68. Causa de transmisión.

Causa	Significado de la causa de transmisión
<4>	Inicializada
<5>	Petición o solicitada (request or requested)
<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

Figura 69. Causa de transmisión.

7.2.3.4 Dirección común del ASDU

Los tres últimos bytes del identificador de unidad de datos contienen la dirección común del ASDU que se compone a su vez de:

☐ Dirección del punto de medida, 2 bytes :=UI16[1..16]<0..65535>

☐ Dirección de registro, 1 byte :=UI8[1..8]<0..255>

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 (single-point), sección 1: incidencias de arranques y tensión bajo límites
<53>	Información de evento (single-point), sección 2: incidencias de sincronización y cambio de hora
<54>	Información de evento (single-point), sección 3: incidencias de cambio de parámetros
<55>	Información de evento (single-point), sección 4: errores internos
<128>	Información de evento (single-point), sección 5: incidencias de intrusismo
<129>	Información de evento (single-point), sección 6: incidencias de comunicaciones
<130>	Información de evento (single-point), sección 7: incidencias de clave privada
<131>	Información de evento (single-point), sección 8: incidencias de Contrato I
<132>	Información de evento (single-point), sección 9: incidencias de Contrato II
<133>	Información de evento (single-point), 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

Figura 70. Direcciones de registro.

Un Contador de energía puede gestionar hasta tres Contratos independientes entre sí, de manera que existen tres conjuntos de información, uno para cada contrato. Los contratos se asignan para los siguientes propósitos.

- ☒ Contrato I: Tarifas de Acceso.
- ☒ Contrato II: Tarifas generales de Compra.
- ☒ Contrato III: Uso genérico: Tarifas Generales de Compra.

En instalaciones FV el tipo de contrato habitual es el contrato tipo III.

En relación con los *puntos de medida*, REE los define como la unidad de direccionamiento básico en el nivel de aplicación, por contraposición a la unidad de direccionamiento en el nivel de enlace, que es el registrador de medida o contador. Habrá al menos una clave para cada uno de los puntos de medida del contador de energía. Opcionalmente podrá haber otras claves para diferentes niveles de acceso a la información y funcionalidad del contador (p.e. una clave de acceso para operaciones de sólo lectura que no permita la alteración de ningún parámetro del registrador). En una sesión abierta para un punto de medida solo se responderá a los mensajes para dicho punto de medida. Para obtener datos de otro punto de medida es necesario cerrar la sesión y abrir otra para ese punto de medida.

7.2.3.5 Dirección de objeto

Las direcciones de objeto se codificarán de acuerdo con la tabla de la página 110.

Dirección de objeto, 1 byte :=UI8[1..8]<0..29>

Dirección	Objeto de Información
<1>	Totales Integrados de Activa Entrante
<2>	Totales Integrados de Activa Saliente
<3>	Totales Integrados de Reactiva primer cuadrante
<4>	Totales Integrados de Reactiva segundo cuadrante
<5>	Totales Integrados de Reactiva tercer cuadrante
<6>	Totales Integrados de Reactiva cuarto cuadrante
<7>	Datos de reserva 1
<8>	Datos de reserva 2
<9>	Bloque de totales integrados genérico con datos de reserva(Punto de medida con direcciones de objeto 1 al 8)
<10>	Bloque de totales integrados genérico sin datos de reserva(Punto de medida con de direcciones de objeto 1 al 6)
<11>	Bloque de totales integrados de consumo puro sin reservas (Punto de medida con direcciones de objeto 1, 3 y 6)
<12..19>	Reservados para futuras ampliaciones del Protocolo
<20>	Información de Tarificación (Totales)
<21>	Información de Tarificación (período tarifario 1)
<22>	Información de Tarificación (período tarifario 2)
<23>	Información de Tarificación (período tarifario 3)
<24>	Información de Tarificación (período tarifario 4)
<25>	Información de Tarificación (período tarifario 5)
<26>	Información de Tarificación (período tarifario 6)
<27>	Información de Tarificación (período tarifario 7)
<28>	Información de Tarificación (período tarifario 8)
<29>	Información de Tarificación (período tarifario 9)

Figura 71. Direcciones de objeto.

7.2.3.6 Elementos de información

Un conjunto de elementos de información puede ser un elemento de información única, una combinación de elementos, o una secuencia de elementos de información que comparten la misma dirección y la misma etiqueta de tiempo. Se usarán los formatos indicados en el apartado de Elementos de información: Información de tarificación de la tabla de la página 112. A continuación se indica el formato de algunos de ellos.

7.2.3.6.A. Elementos de información: totales integrados

Constan de un número de 32 bits (4 bytes u octetos) seguidos de un byte de cualificadores.

Totales integrados:=CP40 {energía, cualificador}

Energía (kWh o kVar) :CP32[1..32]<-2,147,483,648..2,147,483,647>

Cualificador :UI8[1..8]<0..255>

IV=La lectura es válida (IV=0)

CA=Contador sincronizado durante el periodo

CY= Overflow (CY=1)

VH= Verificación horaria durante el periodo (VH=1)

MP= Modificación de parámetros durante el periodo (MP=1)

INT= Se produjo un intrusismo durante el periodo (INT=1)

AL= Periodo incompleto por fallo de alimentación en el periodo (AL=1)

RES= Reserva.

Bit	8	7	6	5	4	3	2	1
	IV	CA	CY	VH	MP	INT	AL	RES

Figura 72. Byte de cualificadores

7.2.3.6.B. Elementos de información: Información de tarificación

La información de tarificación está constituida por el conjunto de valores de interés desde el punto de vista de Tarificación elaborados por el contador de energía en cada Período de Facturación para cada período tarifario. Este conjunto incluye los valores de energía, máximos, excesos y registros de reserva asociados a cada uno de los períodos tarifarios considerados de acuerdo a la discriminación horaria, así como el total referido al conjunto de todos los períodos tarifarios. Los límites en la discriminación horaria de períodos tarifarios de la Tarifa de Acceso coinciden con múltiplos de cuarto de hora. Por coherencia con el menor período para totales integrados, la máxima resolución de discriminación horaria es de múltiplos de cinco minutos.

Este tipo de lecturas de contadores se refiere habitualmente como “lecturas de cierres” y son para cada tipo de contrato (I, II o III) y para cada punto de medida.

El inicio del período de facturación o de valores en curso, es la fecha/hora de inicio de ese período, que coincidirá con la fecha/hora de cierre del anterior período de facturación, siempre y cuando haya tenido lugar un cierre de facturación previo. Ambas fechas/horas serán múltiplos de la máxima resolución de discriminación horaria anteriores al momento de solicitud de la información. El fin del período de facturación coincide con la fecha/hora de cierre de facturación del período en el caso de Memorias y con la fecha del último período de integración, según máxima resolución de discriminación del Contrato, ya finalizado en el momento de la petición en el caso de Valores en curso. Así por ejemplo en el caso de un período de facturación automático mensual correspondiente a Enero del 2010, las fechas de inicio y fin serían 01/01/10 00:00 y 01/02/10 00:00 respectivamente.

Información de Tarificación		Nº Bytes
Información de Tarificación:	:=CP490[VabA_VinA_CinA_VabRi_VinRi_CinRi_VabRc_VinRc_CinRc_R7_CR7_R8_CR8_VMaxA_FechaA_CMaxA_VexcA_CexcA_FechaIni_FechaFin]	62
VabA = Energía absoluta Activa	:= UI32[1..32] <0..4.294.967.295>	4
VinA = Energía incremental Activa	:= UI32[33..64] <0..4.294.967.295>	4
CinA = Cualificador de Energía Activa	:= UI8[65..72] <octeto cualificador>	1
VabRi = Energía absoluta Reactiva Inductiva	:= UI32[73..104] <0..4.294.967.295>	4
VinRi = Energía incremental Reactiva Inductiva	:= UI32[105..136] <0..4.294.967.295>	4
CinRi = Cualificador de Energía Reactiva Inductiva	:= UI8[137..144] <octeto cualificador>	1
VabRc = Energía absoluta Reactiva Capacitiva	:= UI32[145..176] <0..4.294.967.295>	4
VinRc = Energía incremental Reactiva Capacitiva	:= UI32[177..208] <0..4.294.967.295>	4
CinRc = Cualificador de Energía Reactiva Capacitiva	:= UI8[209..216] <octeto cualificador>	1
R7 = Registro 7 reserva	:= UI32[217..248]	4
CR7 = Cualificador del Registro 7 de reserva	:= UI8[249..256]	1
R8 = Registro 8 reserva	:= UI32[257..288]	4
CR8 = Cualificador del Registro 8 de reserva	:= UI8[289..296]	1
VMaxA = Máximo de las Potencias	:= UI32[297..328] <0..4.294.967.295>	4
FechaA = Fecha del Máximo	:= UI40[329..368] <etiqueta de tiempo tipo a>	5
CMaxA = Cualificador de Máximos	:= UI8[369..376] <octeto cualificador>	1
VexcA = Excesos de las Potencias	:= UI32[377..408] <0..4.294.967.295>	4
CexcA = Cualificador de Excesos	:= UI8[409..416] <octeto cualificador>	1
FechaIni = Inicio del período	:= UI40[417..456] <etiqueta de tiempo tipo a>	5
FechaFin = Fin del período	:= UI40[457..496] <etiqueta de tiempo tipo a>	5

Figura 73. Información de tarificación.

El octeto cualificador de la tabla anterior tiene el mismo formato que el indicado en la Figura 8, con la excepción del bit de reserva 0, denominado ahora como U, que indica las unidades (0=kWh o kVArh; 1= MWh o MVARh).

7.2.3.6.C. Elementos de información: Etiquetas de tiempo

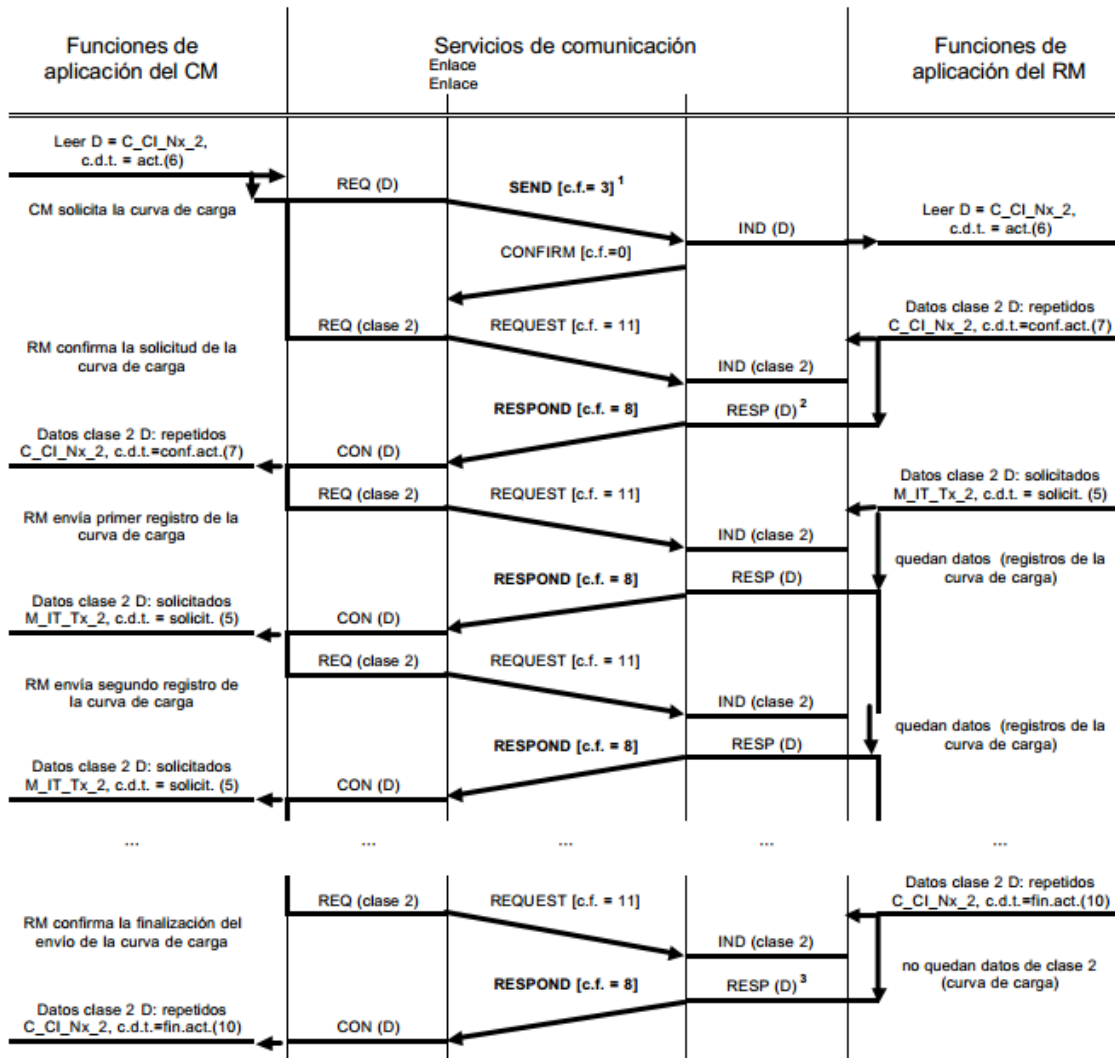
Hay dos tipos de etiquetas de tiempo: tipo a de 5 bytes y tipo b de 7 bytes. El formato de la etiqueta de tiempo del tipo a se indica en la tabla de la página 113. La de tipo b es similar e incluye 2 bytes iniciales adicionales para información de los segundos y milisegundos.

Etiqueta de tiempo tipo a	
Información de tiempo	:=CP40{minuto,TIS,IV,hora,RES1,SU,diares,diasemana,mes,ETI,PTI,año,RES2}
minuto	:=UI6[1..6]<0..59>
TIS=información de tarifa	:=BS1[7]; <0>:=tarifa OFF; <1>:=tarifa ON
IV=Válido	:=BS1[8]; <0>:=válido; <1>:=inválido
hora	:=UI5[9..13]<0..23>
RES1=Reserva 1	:=BS2[14..15]<0>
SU=horario de verano	:=BS1[16]; <0>:=tiempo estándar; <1>:=horario verano
diames	:=UI5[17..21]<1..31>
diasemana	:=UI3[22..24]<1..31>
mes	:=UI4[25..28]<1..12>
ETI=Info tarifa energía	:=UI2[29..30]<0..2>
PTI=Info tarifa potencia	:=UI2[31..32]<0..2>
año	:=UI7[33..39]<0..99>
RES2=Reserva 2	:=BS1[40]<0>

Figura 74. Etiquetas de tiempo tipo a (5 bytes).

7.2.3.6.D. Ejemplo de petición de totales integrados

Aquí podemos ver un ejemplo de cómo sería una petición / respuesta de unos totales integrados.



Abreviaturas empleadas:

- c.d.t.: causa de transmisión
- c.f.: código de función (dentro del campo C)

Figura 75. Ejemplo petición respuesta de totales integrados.

Notas:

1. Las tramas de longitud variable (con datos de usuario de enlace) se indican con negrita (SEND, RESPOND). Las demás tramas son de longitud fija.

2. Se responde con los mismos datos que los enviados en sentido CM => RM. Si se hubiera solicitado un período de integración no disponible, el nivel de aplicación de RM lo indicaría respondiendo con c.d.t. = 18

3. El RM responde con los mismos datos que envió CM para solicitar la curva de carga, y con c.d.t. = 10 (finalización de la activación). El nivel de enlace de CM recibirá una trama NACK en la siguiente petición de datos.

7.2.3.6.E. Integridad y autenticación de datos

Para garantizar que los datos más importantes de un RM que llegan a un CM provienen realmente del equipo que el CM está interrogando (autenticación) y, además, detectar cualquier posible modificación de los mismos por terceras partes (incluso por un CS), se empleará un procedimiento criptográfico de firma electrónica basado en el Digital Signature Standard publicado por el NIST. El algoritmo de clave pública DSA se define para distintas longitudes de claves, en concreto admite claves de longitud $512 + k \cdot 64$ ($k=0..8$). El algoritmo usado en este protocolo empleará claves de 512 bits por consideraciones de velocidad del algoritmo.

El algoritmo descrito en este estándar es del tipo de “clave pública”: el RM dispone de una clave privada (secreta) que le permite elaborar una firma de los datos que envía. Dicha firma es enviada, junto con los datos, al CM, que puede verificar la firma a partir de los datos recibidos y de una clave pública, relacionada con la clave privada. No obstante, dado que es computacionalmente inviable obtener la clave privada a partir de la pública, el CM no puede elaborar la firma a partir de los datos recibidos (o de otros) y de la clave pública; ello permite a un CS leer y verificar los datos de un RM, pero no alterarlos antes de enviarlos al CP sin que el CP detecte la alteración.

El RM elaborará, para cada día completo que transcurra y cada punto de medida que registre, una firma de los datos del día más relevantes, que será enviada al CM a petición de éste. Los datos que se tendrán en cuenta para elaborar la firma serán los de la curva de carga de lecturas de contadores absolutos del día, o los de la curva de carga de energías incrementales del día, tal y como se envían al CM mediante el ASDU

8 (M_IT_TG_2) o el ASDU 11 (M_IT_TK_2), pero incluyendo sólo los siguientes campos de cada registro:

- Tipo de totales integrados (1 octeto): 8 (lecturas) u 11 (energías incrementales).
- Dirección del punto de medida (2 octetos).
- Dirección del total integrado (1 octeto) y total integrado (4 octetos de energía + 1 octeto con cualificadores y número de secuencia), para cada uno de los totales.
- Etiqueta de tiempo del registro (5 octetos).

Los dos primeros datos (tipo de totales integrados y dirección del punto de medida), al ser comunes a todos los registros, se incluirán una única vez al principio de la trama de datos a firmar. A continuación se incluirán los datos específicos de cada uno de los registros (24 para un día completo con período de integración horario), a razón de 11 octetos por magnitud (dirección de objeto, medida, código de calidad y fecha), ordenados por el periodo de integración y, para el mismo periodo de integración, ordenados por la dirección del total integrado. Los registradores que no incorporen las 8 direcciones de total integrado incluirán en la generación exclusivamente aquellas magnitudes que transmitan mediante los ASDU 8 ó 11, con equivalentes de bloques en los ASDU 139 o 140. Esto es, hay tres tipos de firma electrónica: la calculada con 8, la calculada con 6 (sin reservas) y la calculada con 3 magnitudes (consumo sin reservas).

Ejemplo de 24 horas de datos del 1/1/99 firmados por un registrador que transmite 8 magnitudes (tipos 1 y 2):

				Byte 1	Byte 2	Byte 3				
				8 ó 11	Dir.Pnt.Medida					
1	Medida kW activa importada	calidad	(5)vie 01/01/99 01:00 h.invierno-(in)válido							
2	Medida kW activa exportada	calidad	(5)vie 01/01/99 01:00 h.invierno-(in)válido							
3	Medida kVAR reactiva cuadr.1	calidad	(5)vie 01/01/99 01:00 h.invierno-(in)válido							
4	Medida kVAR reactiva cuadr.2	calidad	(5)vie 01/01/99 01:00 h.invierno-(in)válido							
5	Medida kVAR reactiva cuadr.3	calidad	(5)vie 01/01/99 01:00 h.invierno-(in)válido							
6	Medida kVAR reactiva cuadr.4	calidad	(5)vie 01/01/99 01:00 h.invierno-(in)válido							
7	Valor reserva 1	calidad	(5)vie 01/01/99 01:00 h.invierno-(in)válido							
8	Valor reserva 2	calidad	(5)vie 01/01/99 01:00 h.invierno-(in)válido							
1	Medida kW activa importada	calidad	(5)vie 01/01/99 02:00 h.invierno-(in)válido							
2	Medida kW activa exportada	calidad	(5)vie 01/01/99 02:00 h.invierno-(in)válido							
3	Medida kVAR reactiva cuadr.1	calidad	(5)vie 01/01/99 02:00 h.invierno-(in)válido							
4	Medida kVAR reactiva cuadr.2	calidad	(5)vie 01/01/99 02:00 h.invierno-(in)válido							
5	Medida kVAR reactiva cuadr.3	calidad	(5)vie 01/01/99 02:00 h.invierno-(in)válido							
6	Medida kVAR reactiva cuadr.4	calidad	(5)vie 01/01/99 02:00 h.invierno-(in)válido							
7	Valor reserva 1	calidad	(5)vie 01/01/99 02:00 h.invierno-(in)válido							
8	Valor reserva 2	calidad	(5)vie 01/01/99 02:00 h.invierno-(in)válido							
...										
1	Medida kW activa importada	calidad	(6)sab 02/01/99 00:00 h.invierno-(in)válido							
2	Medida kW activa exportada	calidad	(6)sab 02/01/99 00:00 h.invierno-(in)válido							
3	Medida kVAR reactiva cuadr.1	calidad	(6)sab 02/01/99 00:00 h.invierno-(in)válido							
4	Medida kVAR reactiva cuadr.2	calidad	(6)sab 02/01/99 00:00 h.invierno-(in)válido							
5	Medida kVAR reactiva cuadr.3	calidad	(6)sab 02/01/99 00:00 h.invierno-(in)válido							
6	Medida kVAR reactiva cuadr.4	calidad	(6)sab 02/01/99 00:00 h.invierno-(in)válido							
7	Valor reserva 1	calidad	(6)sab 02/01/99 00:00 h.invierno-(in)válido							
8	Valor reserva 2	calidad	(6)sab 02/01/99 00:00 h.invierno-(in)válido							
byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	byte 8	byte 9	byte10	Byte11

Figura 76. Registrador 8 magnitudes

7.3 FORMAS NORMALES

En este apartado veremos la definición de las formas normales,

Primera Forma Normal (1FN)

Una tabla está en Primera Forma Normal si:

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, mínimos. (un contador, un cliente es algo indivisible).
- La tabla contiene una clave primaria única.
- La clave primaria no contiene atributos nulos.
- No debe existir variación en el número de columnas.
- Los Campos no clave deben identificarse por la clave (Dependencia Funcional)
- Debe Existir una independencia del orden tanto de las filas como de las columnas, es decir, si los datos cambian de orden no deben cambiar sus significados
- Una tabla no puede tener múltiples valores en cada columna.
- Los datos son atómicos (a cada valor de X le pertenece un valor de Y, y viceversa).

Esta forma normal elimina los valores repetidos dentro de una BD

Segunda Forma Normal (2FN)

Dependencia Funcional. Una relación está en 2FN si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir que no existen dependencias parciales. (Todos los atributos que no son clave principal deben depender únicamente de la clave principal).

En otras palabras podríamos decir que la segunda forma normal está basada en el concepto de dependencia completamente funcional.

Una dependencia funcional $x \rightarrow y$ es completamente funcional si al eliminar los atributos A de X significa que la dependencia no es mantenida, esto es que $A \in X, X - \{A\} \not\rightarrow Y$. Una dependencia funcional $x \rightarrow y$ es una dependencia parcial si hay algunos atributos $A \in X$ que pueden ser eliminados de X y la dependencia todavía se mantiene, esto es $A \in X, X - \{A\} \rightarrow Y$.

Por ejemplo $\{\text{DNI, ID_PROYECTO}\} \rightarrow \text{HORAS_TRABAJO}$ (con el DNI de un empleado y el ID de un proyecto sabemos cuántas horas de trabajo por semana trabaja un

empleado en dicho proyecto) es completamente funcional dado que ni $DNI \rightarrow HORAS_TRABAJO$ ni $ID_PROYECTO \rightarrow HORAS_TRABAJO$ mantienen la dependencia. Sin embargo $\{DNI, ID_PROYECTO\} \rightarrow NOMBRE_EMPLEADO$ es parcialmente dependiente dado que $DNI \rightarrow NOMBRE_EMPLEADO$ mantiene la dependencia.

Tercera Forma Normal (3FN)

La tabla se encuentra en 3FN si es 2FN y si no existe ninguna dependencia funcional transitiva entre los atributos que no son clave.

Un ejemplo de este concepto sería que, una dependencia funcional $X \rightarrow Y$ en un esquema de relación R es una dependencia transitiva si hay un conjunto de atributos Z que no es un subconjunto de alguna clave de R , donde se mantiene $X \rightarrow Z$ y $Z \rightarrow Y$.

Por ejemplo, la dependencia $SSN \rightarrow DMGRSSN$ es una dependencia transitiva en EMP_DEPT de la siguiente figura. Decimos que la dependencia de $DMGRSSN$ el atributo clave SSN es transitiva vía $DNUMBER$ porque las dependencias $SSN \rightarrow DNUMBER$ y $DNUMBER \rightarrow DMGRSSN$ son mantenidas, y $DNUMBER$ no es un subconjunto de la clave de EMP_DEPT . Intuitivamente, podemos ver que la dependencia de $DMGRSSN$ sobre $DNUMBER$ es indeseable en EMP_DEPT dado que $DNUMBER$ no es una clave de EMP_DEPT .

Formalmente, un esquema de relación R está en 3 Forma Normal Elmasri-Navathe,2 si para toda dependencia funcional $X \rightarrow A$, se cumple al menos una de las siguientes condiciones:

1. X es superllave o clave.
2. A es atributo primo de R ; esto es, si es miembro de alguna clave en R .

Además el esquema debe cumplir necesariamente, con las condiciones de segunda forma normal.

Existen más formas normales pero el exceso es tan negativo como el defecto y consideramos que el rendimiento sería mucho más pesado si hiciéramos mayores divisiones del contenido de nuestras tablas, cosa que acabaría convirtiéndose en más

programas y más modificaciones de BBDD que acabaría penalizando la velocidad de la aplicación.



Figura 77. Diagrama de las formas normales.

7.4 DEFINICIÓN DE LA BBDD

En las siguientes líneas ponemos el código que ejecutamos en el cliente de BBDD para poder cargar nuestra base de datos 'elect',

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `elect` DEFAULT CHARACTER SET latin1 ;
USE `elect`
-----
-- Table `elect`.`datos_clientes`
-----
CREATE TABLE IF NOT EXISTS `elect`.`datos_clientes` (
  `dc_idcliente` CHAR(9) NOT NULL ,
  `dc_nombre` CHAR(200) NOT NULL ,
  `dc_cons_anual` INT(11) NULL ,
  `dc_cons_diario` INT(11) NULL ,
  `dc_cons_noche` INT(11) NULL ,
  `dc_cons_finde` INT(11) NULL ,
```



```

`dc_ultima_mod` TIMESTAMP NULL ,
`dc_pais` CHAR(2) NULL ,
`dc_provincia` CHAR(2) NULL ,
`dc_localidad` CHAR(2) NULL ,
`dc_cp` CHAR(5) NULL ,
`dc_nif` CHAR(15) NULL ,
PRIMARY KEY (`dc_idcliente`),
INDEX `festivo_cliente` (`dc_provincia` ASC, `dc_localidad` ASC) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----
-- Table `elect`.`dias_festivos`
-----
CREATE TABLE IF NOT EXISTS `elect`.`dias_festivos` (
  `df_fecha` DATE NOT NULL ,
  `df_hora` CHAR(10) NOT NULL ,
  `df_provincia` CHAR(2) NOT NULL ,
  `df_localidad` CHAR(2) NOT NULL ,
  `df_precio` INT(11) NULL ,
  `df_desvio` INT(11) NULL DEFAULT NULL ,
  `df_comentario` CHAR(100) NULL ,
  PRIMARY KEY (`df_fecha`, `df_hora`, `df_provincia`, `df_localidad`),
  INDEX `festivos_cliente` (`df_provincia` ASC, `df_localidad` ASC) ,
  CONSTRAINT `festivos_cliente`
    FOREIGN KEY (`df_provincia`, `df_localidad`)
    REFERENCES `elect`.`datos_clientes` (`dc_provincia`, `dc_localidad`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----
-- Table `elect`.`historico_cons`
-----
CREATE TABLE IF NOT EXISTS `elect`.`historico_cons` (
  `hc_cliente` CHAR(9) NOT NULL ,
  `hc_contador` INT(6) NOT NULL ,
  `hc_fecha` DATE NOT NULL ,
  `hc_time` CHAR(10) NOT NULL ,
  `hc_act_imp` DOUBLE NULL DEFAULT NULL ,
  `hc_act_gen` DOUBLE NULL DEFAULT NULL ,
  `hc_v1` DOUBLE NULL DEFAULT NULL ,
  `hc_v2` DOUBLE NULL DEFAULT NULL ,

```

```

`hc_v3` DOUBLE NULL DEFAULT NULL ,
`hc_i1` DOUBLE NULL DEFAULT NULL ,
`hc_i2` DOUBLE NULL DEFAULT NULL ,
`hc_i3` DOUBLE NULL DEFAULT NULL ,
`hc_fp1` DOUBLE NULL DEFAULT NULL ,
`hc_fp2` DOUBLE NULL DEFAULT NULL ,
`hc_fp3` DOUBLE NULL DEFAULT NULL ,
`hc_react1` DOUBLE NULL DEFAULT NULL ,
`hc_timestamp` TIMESTAMP NULL DEFAULT NULL ,
PRIMARY KEY (`hc_cliente`, `hc_contador`, `hc_fecha`, `hc_time`),
INDEX `Ix_2` (`hc_cliente` ASC, `hc_fecha` ASC, `hc_time` ASC),
INDEX `indice_empresas` (`hc_cliente` ASC),
CONSTRAINT `indice_empresas`
  FOREIGN KEY (`hc_cliente`)
  REFERENCES `elect`.`datos_clientes` (`dc_idcliente`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1
COMMENT = 'Tabla con el historico de consumo de los ultimos meses';
-----
-- Table `elect`.`historico_cons_hor`
-----
CREATE TABLE IF NOT EXISTS `elect`.`historico_cons_hor` (
  `hch_cliente` CHAR(9) NOT NULL ,
  `hch_contador` INT(6) NOT NULL ,
  `hch_fecha` DATE NOT NULL ,
  `hch_hora` CHAR(10) NOT NULL ,
  `hch_act_imp` DOUBLE NULL DEFAULT NULL ,
  `hch_act_gen` DOUBLE NULL DEFAULT NULL ,
  `hch_v1` DOUBLE NULL DEFAULT NULL ,
  `hch_v2` DOUBLE NULL DEFAULT NULL ,
  `hch_v3` DOUBLE NULL DEFAULT NULL ,
  `hch_i1` DOUBLE NULL DEFAULT NULL ,
  `hch_i2` DOUBLE NULL DEFAULT NULL ,
  `hch_i3` DOUBLE NULL DEFAULT NULL ,
  `hch_fp1` DOUBLE NULL DEFAULT NULL ,
  `hch_fp2` DOUBLE NULL DEFAULT NULL ,
  `hch_fp3` DOUBLE NULL DEFAULT NULL ,
  `hch_react1` DOUBLE NULL DEFAULT NULL ,
  `hch_timestamp` TIMESTAMP NULL DEFAULT NULL ,
  PRIMARY KEY (`hch_cliente`, `hch_contador`, `hch_fecha`, `hch_hora`),

```

```
INDEX `ix_consumo_h` (`hch_cliente` ASC, `hch_fecha` ASC, `hch_hora` ASC),
INDEX `indice_empresa` (`hch_cliente` ASC),
CONSTRAINT `indice_empresa`
  FOREIGN KEY (`hch_cliente`)
  REFERENCES `elect`.`datos_clientes` (`dc_idcliente`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1
COMMENT = 'Tabla con el historico de consumo horario del contador del c';
```

```
-----
-- Table `elect`.`singularidades`
-----
CREATE TABLE IF NOT EXISTS `elect`.`singularidades` (
  `S_id` INT NOT NULL,
  `S_cliente` CHAR(9) NOT NULL,
  `S_contador` INT(6) NULL,
  `S_fecha` DATE NULL,
  `S_time` CHAR(10) NULL,
  `S_comentarios` CHAR(100) NULL,
  `S_valores_sing` CHAR(10) NULL,
  PRIMARY KEY (`S_id`),
  INDEX `Consulta_sing` (`S_cliente` ASC, `S_contador` ASC, `S_fecha` ASC, `S_time`
  ASC),
  INDEX `singularidad_festivo` (`S_fecha` ASC),
  CONSTRAINT `Consulta_sing`
    FOREIGN KEY (`S_cliente`, `S_contador`, `S_fecha`, `S_time`)
    REFERENCES `elect`.`historico_cons_hor` (`hch_cliente`, `hch_contador`,
    `hch_fecha`, `hch_hora`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `singularidad_festivo`
    FOREIGN KEY (`S_fecha`)
    REFERENCES `elect`.`dias_festivos` (`df_fecha`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
COMMENT = 'Tabla que nos mostrará las singularidades que hayamos detect' /*
comment truncated */;
```

```
-- Table `elect`.`cliente_contador`
-----
CREATE TABLE IF NOT EXISTS `elect`.`cliente_contador` (
  `cc_cliente` CHAR(9) NOT NULL ,
  `cc_contador` INT(6) NOT NULL AUTO_INCREMENT ,
  `cc_consumo_medio` INT NULL ,
  `cc_consumo_nocturno` INT NULL ,
  `cc_ubicacion` CHAR(20) NULL ,
  `cc_longitud` DOUBLE NULL ,
  `cc_latitud` DOUBLE NULL ,
  `cc_descripcion` CHAR(100) NULL ,
  PRIMARY KEY (`cc_contador`),
  INDEX `cliente_contador` (`cc_cliente` ASC),
  CONSTRAINT `cliente_contador`
  FOREIGN KEY (`cc_cliente`)
  REFERENCES `elect`.`datos_clientes` (`dc_idcliente`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
COMMENT = 'Referencia entre los clientes y sus contadores.';

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

7.5 DATOS PREVIOS A LA EJECUCIÓN Y ANÁLISIS DEL CLIENTE.

En los archivos adjuntos de la memoria podremos encontrar los resultados de un día completo, aquí por extensión nos centraremos en la primera hora y su consumo que será suficiente para revisar los datos relevantes.

Varianza

0,027797603

Hora	Potencia circuito pre-medidas	potencia pre-medidas	Energía consumida pre-medidas	Suma Energía pre-medidas	Total	Total
				1660,325	683,37	976,96
				P2	P1	
				0,879166667	7	
8:00:00		211	0,879166667		7	
8:00:15		216	0,9		0,9	
				0,904166666		
8:00:30		217	0,904166667		7	
8:00:45		216	0,9		0,9	
8:01:00		174	0,725		0,725	
8:01:15		204	0,85		0,85	
				0,858333333		
8:01:30		206	0,858333333		3	
8:01:45		198	0,825		0,825	
				0,791666666		
8:02:00		190	0,791666667		7	
				0,966666666		
8:02:15		232	0,966666667		7	
				0,920833333		
8:02:30		221	0,920833333		3	
				0,754166666		
8:02:45		181	0,754166667		7	
				0,608333333		
8:03:00		146	0,608333333		3	
				0,758333333		
8:03:15		182	0,758333333		3	
8:03:30		195	0,8125		0,8125	
8:03:45		183	0,7625		0,7625	
8:04:00		210	0,875		0,875	
				0,895833333		
8:04:15		215	0,895833333		3	
				0,866666666		
8:04:30		208	0,866666667		7	
				0,945833333		
8:04:45		227	0,945833333		3	
				0,841666666		
8:05:00		202	0,841666667		7	
				0,841666666		
8:05:15		202	0,841666667		7	
				0,833333333		
8:05:30		200	0,833333333		3	

8:05:45	205	0,854166667	0,854166666	7
			0,883333333	
8:06:00	212	0,883333333		3
8:06:15	225	0,9375	0,9375	
8:06:30	228	0,95	0,95	
			0,858333333	
8:06:45	206	0,858333333		3
			0,933333333	
8:07:00	224	0,933333333		3
			0,829166666	
8:07:15	199	0,829166667		7
			0,854166666	
8:07:30	205	0,854166667		7
8:07:45	192	0,8	0,8	
			0,691666666	
8:08:00	166	0,691666667		7
8:08:15	186	0,775	0,775	
			0,845833333	
8:08:30	203	0,845833333		3
			0,741666666	
8:08:45	178	0,741666667		7
			0,716666666	
8:09:00	172	0,716666667		7
8:09:15	216	0,9	0,9	
8:09:30	207	0,8625	0,8625	
8:09:45	207	0,8625	0,8625	
			0,795833333	
8:10:00	191	0,795833333		3
			0,879166666	
8:10:15	211	0,879166667		7
			0,795833333	
8:10:30	191	0,795833333		3
8:10:45	189	0,7875	0,7875	
8:11:00	144	0,6	0,6	
8:11:15	177	0,7375	0,7375	
			0,829166666	
8:11:30	199	0,829166667		7
			0,629166666	
8:11:45	151	0,629166667		7
			0,691666666	
8:12:00	166	0,691666667		7
8:12:15	147	0,6125	0,6125	

8:12:30	150	0,625	0,625
8:12:45	157	0,654166667	0,654166666
8:13:00	139	0,579166667	0,579166666
8:13:15	157	0,654166667	0,654166666
8:13:30	153	0,6375	0,6375
8:13:45	149	0,620833333	0,620833333
8:14:00	138	0,575	0,575
8:14:15	142	0,591666667	0,591666666
8:14:30	163	0,679166667	0,679166666
8:14:45	124	0,516666667	0,516666666
8:15:00	118	0,491666667	0,491666666
8:15:15	129	0,5375	0,5375
8:15:30	146	0,608333333	0,608333333
8:15:45	152	0,633333333	0,633333333
8:16:00	148	0,616666667	0,616666666
8:16:15	154	0,641666667	0,641666666
8:16:30	152	0,633333333	0,633333333
8:16:45	188	0,783333333	0,783333333
8:17:00	103	0,429166667	0,429166666
8:17:15	108	0,45	0,45
8:17:30	131	0,545833333	0,545833333
8:17:45	137	0,570833333	0,570833333
8:18:00	148	0,616666667	0,616666666
8:18:15	174	0,725	0,725
8:18:30	151	0,629166667	0,629166666

			0,65416666	
8:18:45	157	0,654166667	7	
8:19:00	177	0,7375	0,7375	
8:19:15	162	0,675	0,675	
			0,59583333	
8:19:30	143	0,595833333	3	
			0,69583333	
8:19:45	167	0,695833333	3	
8:20:00	162	0,675	0,675	
			1,01666666	
8:20:15	244	1,016666667	7	
			0,90833333	
8:20:30	218	0,908333333	3	
8:20:45	207	0,8625	0,8625	
			0,64583333	
8:21:00	155	0,645833333	3	
			0,72083333	
8:21:15	173	0,720833333	3	
8:21:30	174	0,725	0,725	
8:21:45	198	0,825	0,825	
8:22:00	213	0,8875	0,8875	
8:22:15	201	0,8375	0,8375	
			1,00416666	
8:22:30	241	1,004166667	7	
			0,92083333	
8:22:45	221	0,920833333	3	
8:23:00	195	0,8125	0,8125	
8:23:15	186	0,775	0,775	
8:23:30	213	0,8875	0,8875	
			0,75416666	
8:23:45	181	0,754166667	7	
8:24:00	210	0,875	0,875	
8:24:15	189	0,7875	0,7875	
			0,72083333	
8:24:30	173	0,720833333	3	
8:24:45	156	0,65	0,65	
			0,62083333	
8:25:00	149	0,620833333	3	
			0,67916666	
8:25:15	163	0,679166667	7	
			0,76666666	
8:25:30	184	0,766666667	7	
8:25:45	186	0,775	0,775	

8:26:00	152	0,633333333	0,633333333	3
8:26:15	224	0,933333333	0,933333333	3
8:26:30	218	0,908333333	0,908333333	3
8:26:45	193	0,804166667	0,804166667	7
8:27:00	136	0,566666667	0,566666667	7
8:27:15	135	0,5625	0,5625	0,704166667
8:27:30	169	0,704166667	0,704166667	7
8:27:45	200	0,833333333	0,833333333	3
8:28:00	262	1,091666667	1,091666667	7
8:28:15	225	0,9375	0,9375	0,866666667
8:28:30	208	0,866666667	0,866666667	7
8:28:45	222	0,925	0,925	
8:29:00	207	0,8625	0,8625	
8:29:15	189	0,7875	0,7875	0,841666667
8:29:30	202	0,841666667	0,841666667	7
8:29:45	193	0,804166667	0,804166667	7
8:30:00	192	0,8	0,8	

7.6 DATOS POSTERIORES AL ANÁLISIS DE LOS DATOS REGISTRADOS

		Varianza		
			0,002662824	
		Suma Energía pre-medidas		
			238,934375	141,34 97,60
Hora	Potencia post-medidas	Energía consumida pre-medidas	P2	P1
8:00:00	40	0,167361111	0,167361111	

8:00:15	36	0,15	0,15
8:00:30	41	0,171527778	0,171527778
8:00:45	36	0,15	0,15
8:01:00	29	0,120833333	0,120833333
8:01:15	39	0,1625	0,1625
8:01:30	39	0,163888889	0,163888889
8:01:45	33	0,1375	0,1375
8:02:00	32	0,131944444	0,131944444
8:02:15	39	0,161111111	0,161111111
8:02:30	37	0,153472222	0,153472222
8:02:45	30	0,125694444	0,125694444
8:03:00	19	0,080555556	0,080555556
8:03:15	30	0,126388889	0,126388889
8:03:30	33	0,135416667	0,135416667
8:03:45	36	0,147916667	0,147916667
8:04:00	30	0,125	0,125
8:04:15	31	0,128472222	0,128472222
8:04:30	35	0,144444444	0,144444444
8:04:45	38	0,157638889	0,157638889
8:05:00	39	0,161111111	0,161111111
8:05:15	29	0,119444444	0,119444444
8:05:30	33	0,138888889	0,138888889
8:05:45	39	0,163194444	0,163194444
8:06:00	30	0,126388889	0,126388889
8:06:15	38	0,15625	0,15625
8:06:30	43	0,179166667	0,179166667
8:06:45	34	0,143055556	0,143055556
8:07:00	42	0,176388889	0,176388889
8:07:15	33	0,138194444	0,138194444
8:07:30	39	0,163194444	0,163194444
8:07:45	27	0,1125	0,1125
8:08:00	33	0,136111111	0,136111111
8:08:15	26	0,108333333	0,108333333
8:08:30	39	0,161805556	0,161805556
8:08:45	25	0,102777778	0,102777778
8:09:00	34	0,140277778	0,140277778
8:09:15	41	0,170833333	0,170833333
8:09:30	35	0,14375	0,14375
8:09:45	40	0,164583333	0,164583333
8:10:00	32	0,132638889	0,132638889

8:10:15	30	0,125694444	0,125694444
8:10:30	37	0,153472222	0,153472222
8:10:45	32	0,13125	0,13125
8:11:00	19	0,079166667	0,079166667
8:11:15	30	0,122916667	0,122916667
8:11:30	33	0,138194444	0,138194444
8:11:45	20	0,084027778	0,084027778
8:12:00	28	0,115277778	0,115277778
8:12:15	30	0,122916667	0,122916667
8:12:30	20	0,083333333	0,083333333
8:12:45	26	0,109027778	0,109027778
8:13:00	28	0,117361111	0,117361111
8:13:15	31	0,129861111	0,129861111
8:13:30	21	0,085416667	0,085416667
8:13:45	20	0,082638889	0,082638889
8:14:00	23	0,095833333	0,095833333
8:14:15	24	0,098611111	0,098611111
8:14:30	22	0,092361111	0,092361111
8:14:45	21	0,086111111	0,086111111
8:15:00	25	0,102777778	0,102777778
8:15:15	22	0,089583333	0,089583333
8:15:30	24	0,101388889	0,101388889
8:15:45	20	0,084722222	0,084722222
8:16:00	30	0,123611111	0,123611111
8:16:15	26	0,106944444	0,106944444
8:16:30	30	0,126388889	0,126388889
8:16:45	36	0,151388889	0,151388889
8:17:00	17	0,071527778	0,071527778
8:17:15	23	0,095833333	0,095833333
8:17:30	22	0,090972222	0,090972222
8:17:45	28	0,115972222	0,115972222
8:18:00	20	0,081944444	0,081944444
8:18:15	34	0,141666667	0,141666667
8:18:30	30	0,125694444	0,125694444
8:18:45	21	0,088194444	0,088194444
8:19:00	25	0,102083333	0,102083333
8:19:15	32	0,133333333	0,133333333
8:19:30	29	0,120138889	0,120138889
8:19:45	23	0,095138889	0,095138889
8:20:00	22	0,091666667	0,091666667

8:20:15	36	0,148611111	0,148611111
8:20:30	36	0,151388889	0,151388889
8:20:45	30	0,122916667	0,122916667
8:21:00	31	0,128472222	0,128472222
8:21:15	24	0,099305556	0,099305556
8:21:30	24	0,1	0,1
8:21:45	38	0,158333333	0,158333333
8:22:00	41	0,16875	0,16875
8:22:15	34	0,139583333	0,139583333
8:22:30	45	0,188194444	0,188194444
8:22:45	32	0,132638889	0,132638889
8:23:00	38	0,15625	0,15625
8:23:15	26	0,108333333	0,108333333
8:23:30	36	0,147916667	0,147916667
8:23:45	25	0,104861111	0,104861111
8:24:00	30	0,125	0,125
8:24:15	32	0,13125	0,13125
8:24:30	34	0,140972222	0,140972222
8:24:45	21	0,0875	0,0875
8:25:00	30	0,124305556	0,124305556
8:25:15	32	0,134027778	0,134027778
8:25:30	36	0,148611111	0,148611111
8:25:45	36	0,15	0,15
8:26:00	20	0,084722222	0,084722222
8:26:15	37	0,155555556	0,155555556
8:26:30	31	0,130555556	0,130555556
8:26:45	37	0,154861111	0,154861111
8:27:00	28	0,115277778	0,115277778
8:27:15	18	0,072916667	0,072916667
8:27:30	23	0,096527778	0,096527778
8:27:45	38	0,159722222	0,159722222
8:28:00	39	0,161111111	0,161111111
8:28:15	38	0,15625	0,15625
8:28:30	40	0,165277778	0,165277778
8:28:45	37	0,154166667	0,154166667
8:29:00	30	0,122916667	0,122916667
8:29:15	32	0,13125	0,13125
8:29:30	34	0,140277778	0,140277778
8:29:45	32	0,134027778	0,134027778
8:30:00	32	0,133333333	0,133333333

7.7 ÍNDICE Y REFERENCIA DE LAS FIGURAS

Mostramos el índice de las figuras del documento.

- Figura 1. Ejemplo de contadores de energía de diferentes fabricantes (de izquierda a derecha: Landis&Gyr, Círculo, Actaris, ZIV, Orbis) utilizados habitualmente en centrales FV
- Figura 2. Contador de energía con convertidor RS232.
- Figura 3. Modem GSM con convertidor RS232 para lecturas remotas y convertidor USB para lecturas con ordenador local.
- Figura 4. Imagen de nuestro micro controlador
- Figura 5. Puertos de nuestro PIC.
- Figura 6. Características del micro controlador.
- Figura 7. parámetros del microcontrolador.
- Figura 8. Diagrama de bloques del dispositivo.
- Figura 9. Memoria 24LC256
- Figura 10. Parámetros de la memoria 24LC256
- Figura 11. Características de nuestra memoria 24LC256
- Figura 12. Modelo de la memoria 24LC256 que vamos a utilizar.
- Figura 13. Nuestra placa base y micro controlador preparado para recuperar la información y almacenarla en un dispositivo USB.
- Figura 14. Nuestro contador de energía conectado a pc con convertidor RS232 y al PC.
- Figura 15. Imagen de conexión de datos instantáneos.
- Figura 16. Imagen de cómo adquirimos los datos instantáneos con petición desde el PC.
- Figura 17. Trama de datos extraída del contador.
- Figura 18. Consulta del punto de medida
- Figura 19. Causas de transmisión para valores instantáneos.
- Figura 20. Respuesta de valores instantáneos.
- Figura 21. Envío de objetos solicitados en el 163 del ASDU.
- Figura 22. Estructura de los totalizadores de energías.
- Figura 23. Tabla que nos muestra los totales de potencias activas consultadas.
- Figura 24. Valores de diferentes fases de medidas de tensión e intensidad.

- Figura 25. Tabla con las posiciones de los datos en hexadecimal.
- Figura 26. Nuestra solución particular para grabar los segundos en la BBDD
- Figura 27. Nuestro modelo entidad relación.
- Figura 28. Tablas datos_clientes
- Figura 29. Tabla días_festivos
- Figura 30. Tabla singularidades
- Figura 31. Tabla cliente_contador
- Figura 32. Tabla histórico_cons_hor
- Figura 33. La tabla histórico_cons
- Figura 34. Pantalla de inicio
- Figura 35. Alta de cliente.
- Figura 36. Pantalla de consulta de datos de cliente.
- Figura 37. Datos de un cliente.
- Figura 38. Vista de consulta de los contadores del cliente
- Figura 39. Vista de contador de cliente.
- Figura 40. Localización del contador
- Figura 41. Consulta histórico contador
- Figura 42. Salida Ok de la consulta.
- Figura 43. Contador vacío.
- Figura 44. Mensaje de error, por la no especificación del fichero de salida.
- Figura 45. Alta de contador cliente.
- Figura 46. Pantalla de descarga de datos
- Figura 47. Fichero Excel de salida parte 1
- Figura 48. Fichero Excel de salida parte 2
- Figura 49. Los métodos de nuestra aplicación dentro de sus clases
- Figura 50. Interfaz y clase que implementa la interfaz.
- Figura 51. Clase Application Data Object
- Figura 52. Objeto Data
- Figura 53. Lectura del objeto data.
- Figura 54. Contenido del fichero de propiedades
- Figura 55. Clase ExcelUtils que genera la Excel de salida
- Figura 56. Métodos de la clase ExcelUtils
- Figura 57. Consulta desde BBDD para generación de la salida.
- Figura 58. Método de escritura del histórico.

- Figura 59. Clase utilidades con los cálculos necesarios para procesar los datos entre radianes y grados.
- Figura 60. Método que nos devuelve la latitud y longitud de una posición del canvas.
- Figura 61. El panel de dialogoNasa nos muestra el mapa con la información del WorldWindCanvas.
- Figura 62. Esquema de conexionado para comunicación con de los contadores de energía
- Figura 63. Formato de los comandos (tramas) de (a) longitud variable o (b) de longitud fija.
- Figura 64. Estructura del campo de control C de la trama en los dos casos de dirección de comunicación, de maestro a esclavo, PRM=1, o de esclavo a maestro, PRM=0.
- Figura 65. Formato de las tramas de longitud variable, indicando la estructura completa de los datos de aplicación (ASDU).
- Figura 66. Identificadores de tipo, referencia.
- Figura 67. Cualificador de estructura variable.
- Figura 68. Causa de transmisión.
- Figura 69. Causa de transmisión.
- Figura 70. Direcciones de registro.
- Figura 71. Direcciones de objeto.
- Figura 72. Byte de cualificadores
- Figura 73. Información de tarificación.
- Figura 74. Etiquetas de tiempo tipo a (5 bytes).
- Figura 75. Ejemplo petición respuesta de totales integrados.
- Figura 76. Registrador 8 magnitudes
- Figura 77. Diagrama de las formas normales.

7.8 REFERENCIAS

En cuanto a programación:

- La Biblia de Mysql – Ian Gilfillan – Editorial Anaya, 2003

- Java 2 – Curso de programación – Fco. Javier Ceballos - Editorial RA-MA, 2003
- PROGRAMACIÓN EN JAVA – Ángel Esteban - Grupo Eidos, 2000
- Java – Fundamentos de Programación – Judy Bishop - ADDISON-WESLEY, 1999
- Programación orientada a Objetos con Java – F.Duran F. Gutierrez E.Pimentel - Editorial Paraninfo, 2007

Control de protocolos de red eléctrica y contadores:

- ERA SOLAR número 156
- RD 1110/2007, BOE 224 de 18 de septiembre 2007. Regulación unificada de medida del sistema eléctrico.
- RD 1110/2007, BOE 224 de 18 de septiembre 2007. Regulación unificada de medida del sistema eléctrico.
- RED ELÉCTRICA ESPAÑOLA. Reglamento de puntos de medida. Protocolo de comunicaciones entre registradores y concentradores de medidas o terminales de medidas o terminales portátiles lectura. Revisión 10.04.02, 10 de Abril de 2.002.
- RD 1110/2007, BOE 224 de 18 de septiembre 2007. Regulación unificada de medida del sistema eléctrico.
- Transmission Code 2007. Network and System Rules of the German Transmission System Operators. VDN, August 2007

Páginas web:

- En <http://www.ree.es> se puede acceder a los perfiles históricos y en tiempo real de la demanda energética en España.
- <http://www.aperca.org/temp/pdf/Articulo%20Contadores.pdf>
- JAVA <https://www.java.com/es/>
- <http://www.ree.es/es/actividades/operacion-del-sistema/medidas-electricas>
- Librerías Nasa World Wind <http://ifgi.uni-muenster.de/worldwind-tutorial/>
- Librerías POI Apache <http://poi.apache.org/>

- MYSQL desarrollo BBDD www.mysql.com