

Proyecto Fin de Carrera

Cálculo de la eficiencia energética en centros de procesos de datos (Calculation of energy efficiency of data centers)

Para acceder al Titulo de

INGENIERO EN INFORMÁTICA

Autor: Manuel Vila López

Octubre – 2012

INGENIERÍA EN INFORMÁTICA

CALIFICACIÓN DEL PROYECTO FIN DE CARRERA

Realizado por: Manuel Vila López

Director del PFC: Roberto Hidalgo Cisneros

Ponente: Fernando Vallejo Alonso

Título: "Cálculo de la eficiencia energética en centros de procesos de datos"

Title: "Calculation of energy efficiency of data center"

Presentado a examen el día:

para acceder al Título de

INGENIERO EN INFORMÁTICA

Presidente (Apellidos, Nombre): González Harbour, Michael Secretario (Apellidos, Nombre): Martínez Fernández, María del Carmen Vocal (Apellidos, Nombre): Menéndez de Llano Rozas, Rafael Vocal (Apellidos, Nombre): Sánchez Barreiro, Pablo Vocal (Apellidos, Nombre): Sanz Gil, Roberto

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: Vocal

Fdo.: Vocal

Fdo.: Vocal

Fdo.: El Director del PFC

Resumen

El objetivo del presente Proyecto Fin de Carrera es crear una herramienta software para la medida y monitorización de la eficiencia energética en centros de procesos de datos. Esta herramienta se realizará en la empresa CIC Consulting Informático, realizando las medias y las pruebas sobre el data center de la citada empresa.

La métrica estándar para obtener la eficiencia energética es el PUE que es una variable definida por el consorcio 'The Green Grid'. El PUE se calcula como la división de la potencia total de entrada en el centro de procesos de datos entre la potencia suministrada a los equipos IT.

La herramienta realizará la medición del consumo energético en cada uno de los puntos clave que son necesarios para calcular el llamado PUE en sus diferentes categorías y la representación de los valores instantáneos e históricos de los mismos. Así como información relevante del estado de los componentes del data center.

Para la captura en los puntos clave utilizaremos el protocolo a nivel de aplicación SNMP y sondas de medida basadas en toroides con tarjetas de adquisición de la información. Mientras que para la visualización y almacenamiento de los datos se realizara a través de una herramienta web llamada "IDbox", propia de la empresa CIC Consulting Informático.

Las herramientas comerciales existentes actualmente te permiten realizar esta medición de la eficiencia energética pero separada de la monitorización de la situación de los elementos que actúan en esta eficiencia energética y viceversa.

Por lo que la finalidad del proyecto es crear una herramienta que calcule y monitorice el PUE y a su vez nos devuelva información de la situación de los componentes que actúan y afectan al cálculo. Obteniendo información adicional que nos servirá para modificar el entorno físico y así mejorar el PUE.

Palabras Clave: CPD, PUE, SNMP, toroides, eficiencia energética, IDbox, SVG, sinópticos.

Preface

This Master Project aims to create a software application for measurement and monitoring of energy efficiency in data processing centers. This tool will be making in CIC Consulting Informatic Company, realizing the averages and test on the data center of that company.

The standard metric to obtain energy efficiency is PUE that is a variable defined by the consortium ' The Green Grid '. PUE is calculated as division of total power of entry in data center between power supplied IT equipments.

This tool will realize the measurement of energetic consumption in each of the key points that are necessary to calculate PUE in his different categories and representation of instantaneous and historical values of the same ones. As well as relevant information of components of data center.

For capture of key points we will use the protocol to level of application SNMP and probes of measure based in toroides and data acquisition cards. Whereas for visualize and store information it will be execute across a web tool called "IDbox", own of the company CIC Consulting Informatic.

Comercial Tools actually is allowed to realize that energy efficient measurement but it is separated of monitoring and vice versa. Thereby the objective is creating a tool for calculate and monitor PUE and it returns to us information of the situation of the components that act and concern to calculation. Obtaining additional information that will serve us to modify his environment and improve the PUE.

Índice general

Capítulo 1 Introducción	1
1.1. Introducción	1
1.3. Objetivo del proyecto	2
1.3. Estructura del documento	3
Capítulo 2 Tecnología usada	5
2.1. PUE	5
2.2. Sondas de medida	8
2.3. Protocolo SNMP	8
2.3.1. Arquitectura SNMP	9
2.3.2. Formato de mensaje	. 10
2.4. IDbox	. 13
2.4.1. Arquitectura IDbox	. 13
2.5. Lenguaie C#	. 14
2.6. Gráficos SVG	. 14
2.7. Equipos Usados	. 14
Capítulo 3 Planificación del provecto	. 16
3.1 Ámbito funcional del provecto	. 16
3.2. Metodología de desarrollo	. 20
3.2.1. Modelo iterativo incremental	. 21
3.3. Iteraciones y requisitos a nivel de la aplicación	. 22
3.4. Herramientas utilizadas en el desarrollo de la aplicación	. 23
3.4.1. Microsoft Visio	. 23
3.4.2. PDF2svg	. 24
3.4.3. Google SketchUp Pro	. 24
3.4.4. Adobe Illustrator	. 24
3.4.5. Microsoft Visual Studio 2010	. 24
3.4.6. MG-SOFT MIB Browser 2010	. 24
3.4.7. Herramientas IPMI	. 25
3.4.8. Wireshark	. 25
3.4.9. Librería SNMPSharpNET	. 26
Capítulo 4 Preparación del entorno de trabajo	. 27
4.1. Objetivos	. 27
4.2. Ingeniería de Requisitos	. 27
4.2.1. Puntos de medida	. 28
4.2.2. Definición del cálculo PUE	. 29
4.2.3. Refinamiento de requisitos	. 29
4.2.4. Instalación de Sondas de medida	. 31
4.2.5. Prototipo SNMP	. 33
4.2.6. Driver de consultas SNMP	. 34
4.2.7. Casos de uso	. 37
4.3. Implementación	. 38
4.4. Pruebas	. 38
4.5. Sumario	. 39
Capítulo 5 Cálculo del PUE	. 40
5.1. Objetivos	. 40
5.3. Implementación	. 41
*	

5.4. Pruebas	. 42
5.5. Sumario	. 42
Capitulo 6 Creación sinópticos	. 43
6.1. Objetivos	. 43
6.2. Ingeniería de Requisitos	. 44
6.2.1. Refinamiento de requisitos	. 44
6.3. Implementación	. 44
6.3.1. Proceso de creación de sinópticos	. 45
Capítulo 7 Despliegue y aceptación	. 49
7.1. Despliegue de la Aplicación	. 49
7.2. Aceptación de la Aplicación	. 50
7.2.1. Instalación en un equipo del servicio Windows	. 50
7.2.2. Uso Usuario	. 50
7.3. Sumario	. 50
Capítulo 8 Conclusiones y trabajos futuros	. 51
8.1. Conclusiones	. 51
8.2. Trabajos Futuros	. 52
Bibliografía	. 53
Anexo 1. Creación de servicio Windows	. 56
Anexo 2. Uso del MIB Browser	. 63
Anexo 3. WireShark	. 71

Índice de figuras

Figura 1: PUE	7
Figura 2: red SNMP	9
Figura 3: SNMPv1	10
Figura 4: mensaje Get/Set	11
Figura 5: MIB	12
Figura 6: Arquitectura IDbox	13
Figura 7: Layout CPD	17
Figura 8: Pasillo caliente/frío	17
Figura 9: Layout partes CPD	18
Figura 10: Modelo iterativo incremental. Actividades concurrentes	21
Figura 11: Modelo iterativo incremental. Iteraciones	22
Figura 12: Layout CPD sondas completas	28
Figura 13: Layout CPD	32
Figura 14: Diagrama clases driver	36
Figura 15: Casos de uso	38
Figura 10: Proceso SVG 2	46
Figura 17: Proceso SVG 3	46
Figura 18: Proceso SVG 4	47
Figura 19: Proceso SVG 5	47
Figura 20: Sinoptico SVG	48
Figura 21: Anexo I. Paso I servicio Windows	56
Figura 22: Anexo I. Figura estructura servicio windows	50
Figura 23: Anexo 1. Paso 2 servicio Windows	5/
Figura 24: Anexo 1. Paso 5 servicio Windows	50 50
Figura 25: Anexo I. Estructura ServiceI	50 50
Figura 20: Anexo 1. Configuración serviceFrocessinstaller	50
Figura 27: Anexo 1. Configuración servicensianer	50
Figura 20: Anexo 1. Estructura Sotun	5) 60
Figura 29: Anexo 1. Estructura Setup	60
Figura 31: Anexo 1. Fstructura final setun	61
Figura 37: Anexo 1. Proniedades setun	61
Figura 32: Anexo 1. Instalar servicio	62
Figura 34: Anexo 1. Proceso Instalación	62 62
Figura 35: Anexo 2 MIR Browser	65
Figura 36: Anexo 2: Rarra de herramientas	65
Figura 30: Anexo 2: Barra estado	66
Figura 38: Anexo 2. Botion SNMP Protocol Preferences	66
Figura 39: Anexo 2: SNMP Protocol Preferences	67
Figura 40: Anexo 2. Botón Contact Remote SNMP Agente	68
Figura 41: Anexo 2. Querv results	68
Figura 42: Anexo 2. MIB.	69
Figura 43: Anexo 2. Query results	70
Figura 44: Anexo 3. WireShark	72
Figura 45: Anexo 3. Paquete request	72
Figura 46: Anexo 3. Paquete request puerto	73
Figura 47: Anexo 3. Paquete response	73
0 I I I I I I I I I I I I I I I I I I I	

Capítulo 1 Introducción

En este apartado de la memoria se hace una breve introducción al problema que pretende solventar este Proyecto fin de carrera. Tras la introducción se expone la estructura del documento.

Índice

Capítulo 1 Introducción	1
1.1. Introducción	1
1.3. Objetivo del proyecto	2
1.3. Estructura del documento	3
	-

1.1. Introducción

En la situación actual en la que nos encontramos, el gran lastre del ser humano es la necesidad de electricidad para llevar a cabo su vida con satisfacción y comodidad.

Este lastre también se amplia a su vida laboral, necesitando la electricidad en la actualidad para la realización de cualquier trabajo, ya sea desde un carpintero que necesita luz artificial para dar forma a la materia prima y convertirla en un producto manufacturado, hasta el hombre que se sienta delante de un ordenador y monitoriza el rendimiento de un data center [Arrg03], centro de proceso de datos (CPD) o supercomputador.

Este último hombre necesitará mayor consumo de energía eléctrica para llevar a cabo su trabajo ya que se le añadirá el gasto necesario para el data center, CPD o supercomputador.

Este tipo de infraestructuras están ya muy extendidos por todo el mundo, suponiendo el 2% del consumo mundial de energía eléctrica, gastando enormes cantidades de energía eléctrica.

Para ver la importancia de este consumo energético solo ver un ejemplo del gasto que puede hacer un supercomputador. El supercomputador "K-Computer", el segundo mejor supercomputador según el TOP500[TP500] en el año 2012, llega a gastar 436, 52 KW/h que equivale a la descomunal cantidad del consumo de 1034 hogares medios.

Para hacer una aproximación de lo que supondría el gasto del anterior supercomputador he tomado datos de referencia de "IBERDROLA"[IBER12], teniendo un precio de 0,170783 euros el kW/h en el año 2012 para clientes que necesiten una potencia eléctrica mayor a 10 kW. Dando como resultado 74.550,2 euros/h.

1. Introducción

Como vemos, es un precio que ninguna empresa puede soportar. Con motivo de estos consumos tan elevados y el precio cada vez mayor de la energía eléctrica (aumento del 40% desde 2002) se buscan formas de reducir el gasto en energía eléctrica. Por suerte hay formas de reducir el consumo eléctrico, ya sea invirtiendo en componentes más eficientes energéticamente, innovando en nuevas formas de refrigeración, mejorando el gasto en iluminación, haciendo virtualización en los servidores, etc. En el caso de los CPDs se conoce que los grandes gastos en ellos se hacen en la refrigeración y los servidores e equipos. Y es sobre estos dos puntos sobre los que se harán esfuerzos para reducir el gasto.

Pero antes de saber que componentes o recursos se pueden o hay que mejorar, se tendrá que conocer el nivel de eficiencia energética de estos supercomputadores, data centers o centros de proceso de datos (CPD). Y es ahí donde se sitúa este proyecto, en conocer la eficiencia energética de los centros de proceso de datos.

Han sido varias las maneras surgidas para calcular la eficiencia energética, pero para evitar la diversificación en la medición surgió un consorcio de empresas llamado 'The Green Grid'[GGrid], entre las que se encuentran empresas como IDM, AMD, Dell, HP, Intel o Microsoft, como vemos todas ellas con peso dentro del mundo tecnológico de la informática.

'The Green Grid' [Gree12] plantea unas doctrinas para la medida de esta eficiencia energética o como lo hacen llamar 'PUE'[Rec12]. Estas doctrinas han sido aceptadas como el estándar a seguir para realizar las mediciones de la eficiencia energética de un centro de datos. Y se ve con malos ojos no realizarlo siguiendo esas doctrinas.

La eficiencia energética o PUE se calcula como la división de la potencia total de entrada en el centro de procesos de datos entre la potencia suministrada a los equipos IT, este PUE será explicado con mayor detalle y profundidad, junto con los métodos para el cálculo del mismo, en un apartado posterior dentro del "*Capítulo 3 Tecnologías Usadas*".

Este proyecto se realizará en la empresa CIC Consulting Informático, realizando las medidas y las pruebas sobre el data center de la citada empresa. El proyecto se implementará utilizando el lenguaje C# para la parte de captura de los datos necesarios para la medición del PUE y posteriormente para la monitorización y visualización de los datos se utilizará una herramienta llamada 'IDbox'[IDbox], propia de la empresa donde se llevará a cabo el proyecto.

1.3. Objetivo del proyecto

El objetivo del proyecto es desarrollar una herramienta de medición de la eficiencia energética en centros de procesos de datos (CPD), para ayudar a resolver el problema del gasto económico sufrido por un mal uso de la energía o funcionamiento de los componentes (servidores, SAIs,PDUs, etc) de un CPD. Para resolver este problema habrá que elegir los puntos del CPD que den información relevante para realizar el cálculo de la eficiencia energética y para conocer las perdidas que se sufren en el CPD. A la vez que se necesita saber los puntos del CPD que dan información de su consumo, se necesita conocer la información relevante al estado de cada componente del CPD para ver si son rentables energéticamente y decidir si tienen que ser sustituidos.

Una vez decidido los puntos de medida dentro del CPD, para conocer los valores aportados por estos puntos, habrá que utilizar una serie de dispositivos que nos aporten la información del consumo e información de la situación de cada componente del CPD.

Cuando se tengan los valores anteriormente requeridos del consumo energético, estos permitirán realizar las operaciones del cálculo de la eficiencia, atendiendo a la categoría de esta eficiencia apropiada para el CPD.

Con los datos de la información de la situación de cada componente se realizará una aplicación dentro de la herramienta "IDbox" que permita indicar la importancia e información del estado de cada componente.

Los datos calculados de la eficiencia energética e información de cada componente se enviarán a la herramienta "IDbox" para ser visualizados y se enviarán a la vez a una base de datos (BBDD) para tener constancia de sus valores históricos que permitirán conocer el progreso de estos valores.

1.3. Estructura del documento

Tras esta introducción, el resto de la presente memoria se estructura tal como se describe a continuación:

- **Capítulo 2: Tecnología usada**. Describe las tecnologías usadas para llevar a cabo el proyecto. Estas tecnologías son la ya mencionada eficiencia energética o PUE, el lenguaje C# que se utiliza para realizar parte de la herramienta final que realiza la captura de los datos de medición para el calculo del PUE, el protocolo SNMP en el que se fundamenta la captura de los puntos de medición, sondas de medida usadas para capturar el consumo energético en los puntos donde no se puedan obtener por el protocolo SNMP, la herramienta "IDbox" que se utiliza para monitorizar y visualizar los datos capturados para el calculo del PUE, los gráficos SVG que es el modelo a seguir para la visualización de los datos y la librería SNMPSharpNET que se utiliza para implementar la herramienta que consulte mediante el protocolo SNMP en lenguaje C#.
- **Capitulo 3: Planificación del Proyecto**. Describe el ámbito funcional del proyecto y en que casos resulta apropiado su uso. Se indicaran los requisitos que existen, la metodología usada y la planificación del proyecto, definiendo las iteraciones que habrá. También se describirán las herramientas utilizadas para el desarrollo de la aplicación.

- **Capítulo 4: Preparación del entorno de trabajo.** Describe la primera iteración del proyecto, donde se muestra la construcción del primer prototipo cumpliendo las finalidades que se marcaron para esta iteración. De esta manera vemos la elección de los puntos de medida del layout del centro de proceso de datos para realizar el calculo del PUE, la definición del propio calculo del PUE y el driver que nos permitirá realizar consultas mediante el protocolo SNMP a los componentes del CPD, así como la instalación de las sondas de medida en aquellos puntos en los que no se pueda consultar consumos mediante el protocolo SNMP. Se observaran también los casos de uso del driver y el refinamiento de requisitos, para posteriormente diseñar los componentes y finalmente implementarlos y probarlos.
- Capítulo 5: Calculo del PUE. Describe la segunda iteración del proyecto, donde se muestra la finalización del proceso de captura y calculo de información necesaria para el proyecto. Se utilizara una herramienta "Administrador" proporcionada por la empresa CiC Consulting Informático que permitirá definir las operaciones que se desean realizar entre los datos provenientes de los puntos de medida del layout y componentes SNMP. Estas operaciones nos permitirán definir los cálculos necesarios para llevar a cabo el cálculo del PUE.

Se observaran también el refinamiento de requisitos para posteriormente diseñar los cálculos a realizar y finalmente implementarlos y probarlos

Capítulo 6: Creación sinópticos. Describe la tercera iteración del proyecto, donde se realiza la parte visual del proyecto que consta de una mejora para la visualización del cálculo del PUE y los datos de relevancia de los componentes del centro de proceso de datos.

La visualización se hace a través de sinópticos SVG que permitirán ver la disposición física del centro de proceso de datos junto con la información del cálculo del PUE y datos de interés de cada componente del mismo.

Se observará también la aparición de nuevos requisitos encontrados y modificaciones de los existentes, las decisiones que se tomaron durante la fase de implementación y la realización de pruebas.

Capítulo 7: Despliegue y Aceptación. Muestra el proceso de despliegue de la aplicación en el que se realiza la creación del instalador de un servicio Windows que nos permite hacer uso del driver SNMP desarrollado para capturar datos de forma ininterrumpida.

También se habla de la aceptación de la aplicación, incluyendo la instalación del anterior servicio Windows mencionado y el uso de usuario. Así como la configuración de un driver OPC para obtener los datos en la herramienta "IDbox" procedentes de las sondas de medida.

Capítulo 8: Conclusiones y trabajos futuros. Concluye la presente memoria y comenta posibles trabajos futuros.

Capítulo 2 Tecnología usada

El presente capítulo describe las tecnologías sobre las que se fundamenta el presente proyecto. Dentro del capítulo se describirá con detalle el concepto de PUE, la herramienta "IDbox" que nos servirá para llevar a cabo el proyecto, el lenguaje de programación con el que se ha realizado la aplicación y las diferentes tecnologías que me han servido para capturar la información necesaria para el cálculo del PUE y su visualización.

Índice

С	apítulo 2 Tecnología usada	5
	2.1. PUE	5
	2.2. Sondas de medida	8
	2.3. Protocolo SNMP	8
	2.3.1. Arquitectura SNMP	9
	2.3.2. Formato de mensaje	10
	2.4. IDbox	13
	2.4.1. Arquitectura IDbox	13
	2.5. Lenguaje C#	. 14
	2.6. Gráficos SVG	. 14
	2.7. Equipos Usados	. 14

2.1. PUE

El PUE (Power usage effectiveness) es una medida de eficiencia energética en un data center, el cual nos indica que porcentaje de potencia se está utilizando realmente en los equipos de cómputo. El PUE es la relación entre la cantidad de potencia utilizada por un data center y la potencia suministrada a los equipos informáticos de computación. Todo lo que no se considera un dispositivo de computación, se sitúa dentro de una categoría para componentes que gastan energía sin aportar trabajo productivo en el cómputo del CPD.

PUE=Consumo eléctrico en el data center / Consumo eléctrico en cargas IT

Analizando más en profundidad la expresión anterior del PUE, entendemos por "**Consumo eléctrico en el Data center** como todo el consumo de energía de toda nuestra instalación, servidores, máquinas de climatización, iluminación, etc; y donde se dice "**Consumo eléctrico en cargas IT**", se correspondería con el consumo de energía realizado por el equipamiento IT (Information Tecnology) únicamente, servidores, equipos de networking, etc.

Se entiende como carga IT, todo aquel componente que sirve en el proceso de realizar computo. En este proyecto las cargas IT que se van a medir son servidores y SAIs.

Así pues el PUE nos proporcionará un indicador que relaciona el consumo total de energía de la instalación, con la cantidad de ella que se dedica a alimentar el equipamiento IT. Como se puede apreciar el mejor valor de PUE posible (e inalcanzable en la práctica) es 1, ya que esto significaría que cada KW consumido en nuestra instalación se ha dedicado a alimentar servidores y no se ha empleado nada en refrigeración, iluminación, etc.

Ahora, indicar que dependiendo del tipo de data center que se tenga, el PUE se calculará de una manera u otra, los tipos de data center posibles son:

- los data center dedicados, donde se incluye las fuentes de energía en el punto de transferencia del servicio público al operador del data center pero que no tiene en cuenta la climatización, iluminación y todo lo necesario para el soporte del funcionamiento del centro de datos
- los data center mixtos que si tienen en cuenta estos datos.

En el caso de este proyecto el data center es de categoría mixta, puesto que tendremos en cuenta todos los mecanismos de soporte para el funcionamiento (ventilación, iluminación,...) del data center.

Además se establecen cuatro categorías de PUE, que establecerán distintos niveles de precisión o calidad a la hora de medir la eficiencia energética de un data center, siendo estas categorías las siguientes:

- Categoría "0": Esta categoría establece un ciclo de medición de 12 meses, en los cuales el consumo de los sistemas IT se medirá a la salida del sistema SAI (si tenemos varias SAIs, sumaremos la salida de cada una de ellas) y donde se tomarán en cuenta los valores pico (máximos) medidos en KW. La medida correspondiente al consumo total de potencia se realizará en la acometida eléctrica a la infraestructura. No se considera una categoría válida para data centers que no se alimenten exclusivamente de la red eléctrica (usen gas natural, solar, etc.)
- **Categoría "1"**: Similar a la 0, pero en lugar de tener en cuenta sólo los valores máximos, se tendrá en cuenta el total del consumo. El ciclo de medición es también de 12 meses y las medidas se tomarán en KW. La energía total incluirá todos los medios de alimentación disponibles (usen gas natural, solar, etc.)
- **Categoría "2"**: Las medidas se realizarán en ciclos de 12 meses y se realizarán en KW. El punto de medida en este caso ya no será la SAI, sino que se realizará en la salida de la PDU (Power Distribution Unit) que soporte la infraestructura IT (o nuevamente la suma de la salida de las distintas PDUs).La energía total se calculará de la misma manera que en la categoría 1, pero al situar el punto de medida del consumo IT en las PDU, eliminaremos (sacaremos a relucir) las perdidas asociadas a la transformación de las PDUs, switches estáticos, etc.
- Categoría "3": Con un ciclo de medida de 12 meses y realizada en KW, los puntos de medida se situarán en el punto de conexión del equipamiento IT al sistema eléctrico. Esta categoría es la más precisa de todas y engloba a las anteriores, siendo capaz de ofrecernos datos relativos al consumo de nuestra infraestructura IT y de la infraestructura que la soporta.

Hay muchos factores, sobre todo ambientales (*no es lo mismo refrigerar un Datacenter en New York (USA) que en Sevilla (España))* que influyen en el cálculo del PUE y en lo bueno o malo que resulta el índice resultante en relación a instalaciones en condiciones ambientales similares, así que sólo a modo de referencia incluyo una gráfica y una tabla de clasificación presentes en la *Figura 1*:



PUE	DCiE	Level of Efficiency
3.0	33%	Very Inefficient
2.5	40%	Inefficient
2.0	50%	Average
1.5	67%	Efficient
1.2	83%	Very Efficient

Figura 1: PUE

Hay que indicar también que existen problemas a la hora de calcular el PUE, algunos de éstos se muestran a continuación:

- Algunos subsistemas no están presentes en el centro de datos (por ejemplo la iluminación al aire libre).
- Algunos subsistemas de apoyo son de uso múltiple y comparten funciones (por ejemplo, las plantas de refrigeración), por lo que, la energía atribuida a dichos subsistemas no se puede medir directamente, por lo que tendremos que medir el porcentaje que recibe el data center.
- Algunos subsistemas son extremadamente imprácticos o costosos para el consumo de energía (por ejemplo, la perdidas de las power distribution units (PDU) debido al número de conexiones de salida, o de conmutación).
- Algunos puntos prácticos de medición de energía incluyen cargas que no están relacionadas con el centro de datos pero no se pueden separar durante la medición, con lo que tendremos que tomar el porcentaje que recibe del data center.

La conclusión es que el PUE es una medida orientativa y relativamente fácil de obtener, que nos permitirá conocer le eficiencia energética de nuestro CPD desde un punto de vista estrictamente de consumo energético.

Finalmente recordar, que medir es la primera actividad del proceso de mejora de un data center, por tanto, invertir en conocer y medir que ocurre en nuestro data center será una de las mejores inversiones que podamos realizar.

2.2. Sondas de medida

Las sondas de medidas están compuestas por dos componentes. Estos componentes son los toroides y las tarjetas de adquisición de medidas. Estas sondas de medida se encargan de obtener el consumo en los puntos del CPD donde no se pueden obtener mediante el uso del protocolo SNMP. El proceso consiste en instalar un toroide alrededor de un cable de alimentación y así proporcionar una corriente proporcional a la que circula por el cable de alimentación para conocer su consumo o energía. La salida del toroide irá conectada a una tarjeta de adquisición, la cuál se encarga de obtener la medida obtenida del toroide y enviarla a un servidor OPC. Este servidor OPC estará comunicado con la herramienta "IDbox" a través de un driver que consultará este servidor, permitiendo consultar los datos obtenidos.

Los toroides utilizados son de la marca "circuitor", de la gama TCH [CIRC12] que nos permite una mayor precisión, los cuales son unos transformadores de corriente. Estos transformadores de corriente se utilizan para medir la corriente basándose en utilizar el campo magnético natural del conductor para determinar la corriente. Físicamente se trata de una bobina con forma de toroide por la que hacemos pasar un cable conductor de electricidad que nos permitirá conocer la corriente de ese cable. Sabiendo la diferencia entre el campo magnético de la bobina y el propio del cable se conoce la corriente de este cable.

Las tarjetas de adquisición son de la marca "Beckhoff" de la serie CX9010 [BECK12] que se compone de un módulo para conexiones Ethernet y una serie de módulos auxiliares para la captura de datos. Estas tarjetas nos permiten montar un PLC y consultar los datos a través de Ethernet de los toroides conectadas a ellas. La conexión entre las tarjetas de adquisición y el equipo que las monitorice se hará a través de un software propio de la marca, llamado "TwinCat" que vuelca los datos obtenidos a un servidor OPC (la propia tarjeta de adquisición) donde pueden ser consultados por multitud de usuarios haciendo uso del protocolo OPC [OPC12].

La selección y compra de los toroides y tarjetas de adquisición necesarios para llevar a cabo la medición del consumo energético en los puntos de medición, se hará por cuenta de la empresa CiC, que ha decidido el mejor tipo de sonda para este trabajo y que se encargará de la instalación de las mismas.

2.3. Protocolo SNMP

El protocolo SNMP (Simple Network Management Protocol) [Mau05] [Stal99], es un protocolo de nivel de aplicación para realizar consultas a los diferentes elementos que forman parte de una red (routers, switches, hubs, hosts, módems, impresoras, servidores, PDUs, SAIs, etc). Utilizado para sondear datos de consumo y información de interés de los elementos de la red.

Para que un equipo pueda ser consultado por el protocolo SNMP tienen que estar conectado a una red SNMP, y para esto el equipo tiene que pertenecer a una "comunidad" (que representa una red entre componentes) que será la misma para los equipos que se quiera consultar. Cada equipo conectado a la red SNMP ejecuta una serie de procesos (convirtiéndolo así en agentes SNMP) para de esta manera poderse realizar la administración de este equipo de manera local o remota en la red (normalmente utilizando UDP).

Con los procesos anteriormente mencionados ejecutándose se crea una MIB (la MIB es una base de datos relacional organizada por objetos con una serie de atributos, que contiene información del estado del agente SNMP) que puede ser consultada remotamente.

El tipo de información que se obtiene de los distintos equipos de la red varia en función de su tipo, algunos ejemplos son:

- En los routers, su MIB contendría información de sus interfaces activos, velocidades de sus enlaces, número de errores,...
- En los servidores, su MIB contendría información de sus puertos IP activos, memoria usada, número de discos en uno, espacio en los discos, estado del servidor,...
- En los equipos de alimentación o SAI, la MIB contendría información de sus puertos IP activos tensiones de entrada y salida, potencias de salida, factor de potencia, estado de la batería,....

El propósito básico del protocolo SNMP es la comunicación entre la estación administradora y el agente SNMP, permitiendo a los agentes transmitir datos pedidos desde la estación de administración.

En la Figura 2 que muestra a continuación se observa un esquema de una red SNMP.



Figura 2: red SNMP

2.3.1. Arquitectura SNMP

La arquitectura SNMP se compone de 3 componentes principales:

- estación de administración

- agente SNMP
- MIB

Como se ve en la *Figura 2*, hay una estación de administración o consola de administración de SNMP que realiza consultas sobre los distintos agentes SNMP, para ello todos está en la misma "comunidad". Y recibirá de la MIB de cada agente SNMP la información solicitada.

2.3.2. Formato de mensaje

Para la comunicación entre la estación de administración y el agente SNMP se hace uso de una serie de mensajes, que en este proyecto serán de la versión 1 del protocolo SNMP, estos mensajes se tratan mediante pooling. El problema de este pooling es que puede llegar a saturar la red dependiendo de número de nodos o agentes SNMP que tengas en uso.

2.3.2.1. Mensajes SNMP v1



Figura 3: SNMPv1

Los mensajes SNMP se agrupan en mensajes de pregunta y mensajes de respuesta.

- 1. Mensajes pregunta: la estación administradora envía una solicitud a un agente, pidiéndole información o mandando actualizar su estado, se le conoce como sondeo a este método.
- 2. Mensajes respuesta: la información recibida del agente es la respuesta o la confirmación de la acción solicitada.

Otro método es la interrupción (trap): de esta manera un agente puede mandar la información al nodo administrador puntualmente, ante una anomalía en este.

A continuación se muestran algunos de los diferentes tipos de mensajes que soporta SNMPv1, los cuales son los que aparecen en la *Figura 3*:

- Get-request: Solicita el valor de una o varias variables.
- Get-next-request: Solicita el valor de la siguiente variable o grupo de variables:

- Set-request: Da valor a una variable.
- Get-response: Devuelve el valor de una o varias variables.
- Trap: El agente notica al gestor de alguna situación especial que sucede en el elemento de red.

2.3.2.2. Formato de los mensajes Get y Set

Los mensajes de pregunta o respuesta en SNMP tienen un formato fijo, dividido en siete tipos de campos que se distribuyen según se muestra en la *Figura 4*. La descripción del significado de cada campo se muestra a continuación:

Versión:	0 para SNMPv1, 1 para SNMP v2, 2 para SNMP v3.
Comunidad:	nombre que deberá ser igual en la estación de administración y agente para poder comunicarse.
Tipo de PDU:	indica el tipo de mensaje (set, get, next, etc.).
Req ID:	indica el ID del agente SNMP que responde al mensaje.
Status Error:	dará cero si no hay errores y otro valor diferente cuando ocurra un error.
Índice Error:	indica el índice de la variable que dio el fallo dentro del agente SNMP.
Lista Objeto-Valor:	valores de respuesta en mensajes "get" o valores de escritura en variable para mensajes "set".

Formato de mensajes											
	Versión	Comuni.	Tipo PDU	Req. ID	Status error	Índice error	Nombre	Valor	Nombre	Valor	
< Cabecera SNMP >			< Cabo	ecera ge	t/set >	≺ ^{Var}	iables	para get/	set	->	

Figura 4: mensaie Get/Set

Para establecer la comunicación entre la consola de administración y un agente SNMP se utilizan los puertos de comunicación estándares en SNMP para su uso, los cuales son los puertos UDP 161 y 162. El puerto 161 se utiliza para la transmisión normal de comandos SNMP y el puerto 162 se utiliza para los mensajes de tipo interrupción.

Para conocer las características de cada variable de un agente SNMP y saber si se pueden consultar o modificar tenemos la MIB, la cual es una BBDD relacional organizada por variables de los agentes SNMP, que contiene información del estado y actualizada por los agentes. Esta MIB esta organizada en forma de árbol. En la *Figura 5* se muestra una captura de una MIB de un agente SNMP, donde se ve un nodo raíz que es obligatorio para todos los agentes SNMP y de él nacen ramas que representan variables o conjuntos de variables, en la que cada una de ellas tiene una descripción de la variable y un valor para cada una de ellas. Indicar que si no es un nodo hoja no se tendrá valor para el nodo, lo que significa que al consultarlo nos devolverá null.



Ahora para explicar como obtener el valor de cada una de las etiquetas tenemos que hacer referencia a las OIDs.

2.3.2.3. Formato OID

La OID es una representación en forma numérica de la ruta que se sigue dentro de la MIB para llegar desde la raíz de la MIB a un nodo hoja de la misma. Cada nodo de la MIB tiene un identificador, el cual es el número de hijo que ocupa ese nodo respecto a su padre.

De tal manera que al ir pasando por los sucesivos nodos la OID va cogiendo ese valor para que forme parte de su representación.

Para entender mejor la idea de la OID, se seguirá la formación de las OID por medio de un ejemplo, usaremos la *Figura 5* que corresponde a un router de la marca "*Cisco*".

Si se quiere acceder a la etiqueta "*atLocal*" de la MIB que se muestra en la *Figura 5*, primero se tendría que acceder a la MIB, lo que implica que se entra al nodo raíz (iso) y éste pasa a formar parte del OID, siendo la OID equivalente a ".1".

Como no se ha alcanzado la etiqueta deseada habrá que seguir recorriendo la MIB. Para seguir avanzando y llegar al destino hay que pasar por la etiqueta *"identified organization"*, pasando ahora a valer la OID *".1.3"* y así sucesivamente hasta alcanzar el nodo final, quedando la OID de esta manera *"1.3.6.1.4.1.9.3.2"*.

2.4. IDbox

La aplicación software "IDbox" ha sido desarrollada por la empresa CIC Consulting Informático, la cual nos permite recoger, integrar, procesar y visualizar grandes volúmenes de información. El objetivo final es permitir mediante herramientas analíticas y gráficas, la realización de cálculos y análisis de información (en tiempo real o histórico). Esta aplicación ha sido utilizada para monitorizar, procesar y visualizar los datos capturados en los puntos de adquisición necesarios.

2.4.1. Arquitectura IDbox

Como se muestra en la *Figura 6*, "IDbox" está compuesto por cinco partes diferenciadas:

- i) Origen de datos: Se encarga de la captura de información. Las magnitudes físicas o digitales serán capturadas mediante diferentes tipos de sondas para su posterior transferencia.
- ii) Adquisición de datos: codificados los drivers como servicios de Windows. Se conectará a las sondas anteriormente mencionadas y enviará los datos de estas mismas a la Topológica mediante UDP.
- iii) Procesamiento: Se encarga de recibir la información de la adquisición de datos y manipularla. Mediante la "Topología" se realizarán operaciones aritméticas básicas sobre los datos y se enviarán los datos "en bruto" y los resultados de dichas operaciones hacia los agentes de datos (dentro de la "Topología" puede no realizarse ninguna operación aritmética). La definición de la "Topología" se realiza a través de un software llamado "Administrador" que se encarga de definir sobre que datos se realizan dichas operaciones. El agente de consultas ofrecerá la información a diferentes servicios web y el agente de almacenamiento enviará los datos a una BBDD para su almacenamiento.
- iv) Almacenamiento: se encarga de realizar la escritura de los datos enviados desde el agente de almacenamiento en una BBDD.
- v) Visualización: Mediante un servicio web se podrán consultar los datos del agente de datos. Este agente de datos tiene una jerarquía de memoria de tal manera que tiene cacheado los datos más actuales, y si fuera necesario obtener información histórica de ellos se accedería a la BBDD para consultarlos.



Figura 6: Arquitectura IDbox

2.5. Lenguaje C#

El Lenguaje C# [Stel10] [Trol10] o C Sharp es un lenguaje de programación orientado a objetos, desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270).

La sintaxis básica deriva de C y C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma.

Este lenguaje ha sido utilizado para realizar la parte del proyecto que conlleva la creación de un driver para la consulta a los agentes SNMP.

2.6. Gráficos SVG

Los Gráficos Vectoriales Escalables o SVG [Eis11] es una especificación para describir gráficos vectoriales bidimensionales. Estos gráficos representan imágenes vectorial que están formadas por objetos geométricos independientes (segmentos, polígonos, arcos, imágenes de mapa de bits, texto, etc.), cada uno de ellos definido por distintos atributos matemáticos de forma, de posición, de color, etc.

Por ejemplo un círculo de color rojo quedaría definido por la posición de su centro, su radio, el grosor de línea y su color.

El interés principal de los gráficos vectoriales es poder ampliar el tamaño de una imagen a voluntad sin sufrir la pérdida de calidad que sufren los mapas de bits. De la misma forma, permiten mover, estirar y retorcer imágenes de manera relativamente sencilla. Estos gráficos SVG se utilizarán para visionar la misma información desde diferentes dispositivos (computadores, móviles, tablets, etc) sin sufrir problemas de resolución debido a que los gráficos SVG se autoescalan al tamaño de la pantalla del dispositivo.

2.7. Equipos Usados

En este apartado indicaré los equipos que ha proporcionado la empresa CIC Consulting Informático para llevar a cabo este proyecto, que son:

- PDUs.
- Racks.
- SAI.
- Servidores.

Una vez conocidos los equipos con los que tendré que trabajar, paso a analizar cada uno de ellos para ver el uso que le puedo dar a cada uno.

Una PDU o unidad de distribución de energía es un dispositivo equipado con múltiples salidas diseñados para distribuir energía eléctrica (parecido a una regleta de enchufes), sobre todo a bastidores de las computadoras y equipos de red ubicados en un centro de datos. Estos elementos pueden incluir monitoreo mediante protocolos como ModBus y SNMP. Las PDU utilizadas por la empresa son de la marca DELL y vienen montadas dentro de los racks con lo que no se conoce el modelo.

Un rack es un soporte metálico destinado a alojar equipamiento electrónico, informático y de comunicaciones. Las medidas para la anchura están normalizadas para que sean compatibles con equipamiento de cualquier fabricante. Los racks son útiles en un centro de proceso de datos (CPD) para alojar equipos como servidores, conmutadores, enrutadores, paneles de parcheo (centralizan todo el cableado de la planta), cortafuegos o estaciones de monitorización como monitores. Los Racks utilizados por la empresa son de la marca DELL y cuyos modelos son Poweredge 4220 [PW20] y Poweredge 4210 [PW10].

Una SAI o sistema de alimentación ininterrumpida es un dispositivo que gracias a sus baterías u otros elementos almacenadores de energía, puede proporcionar energía eléctrica por un tiempo limitado. Otra función que puede tener una SAI es mejorar la calidad de la energía eléctrica que llega a las cargas, filtrando subidas y bajadas de tensión y eliminando armónicos de la red en el caso de usar corriente alterna. La SAI utilizada por la empresa es de la marca EATON y modelo 9355/30N7 [EAT12].

Un Servidor puede una de las descripciones siguientes:

- Una computadora que provee servicios a otras computadoras.
- Una aplicación informática o programa que realiza tareas en beneficio de otras aplicaciones.
- Una computadora en la que se ejecuta un programa para realizar una tarea en beneficio a otras aplicaciones.
- Un proceso que envía información a otro proceso.

Los servidores empleados por la empresa son de la marca DELL y cuyos modelos son Poweredge 1950, Poweredge R300, Poweredge R410, Poweredge R610, Poweredge 1850, Poweredge 850, Poweredge 2850, Poweredge 1800 y MD1000 [DELLS12].

Capítulo 3 Planificación del proyecto

En este capítulo se describe la planificación que tendrá el proyecto, mostrando en líneas generales su ámbito funcional, la metodología de desarrollo utilizada, los requisitos de alto nivel de la aplicación, las iteraciones programadas, el diseño de los artefactos base, y las herramientas utilizadas para la construcción de prototipos.

Índice

Capítulo 3 Planificación del proyecto	16
3.1 Ámbito funcional del proyecto	16
3.2. Metodología de desarrollo	20
3.2.1. Modelo iterativo incremental	21
3.3. Iteraciones y requisitos a nivel de la aplicación	22
3.4. Herramientas utilizadas en el desarrollo de la aplicación	23
3.4.1. Microsoft Visio	23
3.4.2. PDF2svg	24
3.4.3. Google SketchUp Pro	24
3.4.4. Adobe Illustrator	24
3.4.5. Microsoft Visual Studio 2010	24
3.4.6. MG-SOFT MIB Browser 2010	24
3.4.7. Herramientas IPMI	25
3.4.8. Wireshark	25
3.4.9. Librería SNMPSharpNET	26

3.1 Ámbito funcional del proyecto

En esta sección se describe el ámbito funcional de la herramienta software, la cual deberá realizar el cálculo de la eficiencia energética de un centro de proceso de datos (CPD) y su posterior visualización.

Un CPD se describe como una ubicación en la que se encuentran una serie de recursos para el procesamiento de información de una organización. Estos recursos son dependencias acondicionadas, computadores, sistemas de alimentación ininterrumpida y redes de comunicación.

Para observar de una manera clara como se lleva a cabo el cálculo de la eficiencia explicaré la disposición del layout del CPD de la empresa donde se ha llevado a cabo el proyecto, que se observa en la *Figura 7*.



El CPD es alimentado desde la empresa Eon y reparte esta alimentación entre los componentes del CPD. Los recursos del CPD pueden agruparse en tres tipos:

- Servidores y SAI
- Ventilación
- PLC, alarmas, accesos y extinción

A los servidores les llega la alimentación de manera doble, esta alimentación doble es aportada por la empresa suministradora de electricidad y por una SAI [Smit11]. La SAI [Will00] es un componente que permite aportar electricidad a los servidores si el suministro de electricidad se interrumpe, permitiendo así el funcionamiento continuo de los servidores ante perdidas del suministro de electricidad. Los servidores están dispuestos en una serie de racks, los cuales están distribuidos con el ideal de pasillo caliente/pasillo frío que se muestra en la *Figura 8*[Arreg03].



Figura 8: Pasillo caliente/frío

La ventilación esta diseñada para implementar el ideal de pasillo caliente/pasillo frío. Este sistema de ventilación consta de un sistema de extracción de aire caliente y suflación de aire frío, de tal manera que aporta aire frío al pasillo entre los racks para refrigerar los servidores y recoge el aire caliente que emiten los servidores al estar en funcionamiento. El PLC, alarmas, accesos y extinción son sistemas que controlan cada uno una parte dentro del edificio de la empresa. Estos sistemas son propios de la empresa y pueden variar dependiendo del CPD que se analice. En la *Figura 9* se muestra detalladamente las partes en las que se separa el CPD.



Una vez conocida las partes que componen el CPD, indicar que para llegar al cálculo del PUE necesitaremos una serie de medidas en distintas partes o puntos del CPD. Estos puntos de medida tendrán que aportarnos información de la potencia o consumo que pasa o llega a los puntos que se quieren.

Como se ha indicado ya antes, el PUE es una relación entre el consumo energético del CPD y el consumo energético de las cargas IT. Por lo que necesitaremos conocer el consumo en estas dos partes. Como complemento de estos dos puntos de medición básicos para realizar el cálculo del PUE, se ha propuesto otros puntos, que nos servirán para conocer el PUE en sus diferentes categorías y conocer detalladamente las perdidas sufridas en el CPD. Los puntos propuestos son la entrada y salida de la SAI, la salida de las PDUs, el cuadro eléctrico de la ventilación y PLC, accesos, alarmas y extinciones, que se muestran en la *Figura 12*.

A parte de los puntos de medición necesarios para el cálculo del PUE en sus diversas categorías, se ha decidido aportar información de cada servidor del CPD y de la SAI del CPD. Esta información tiene como propósito aportar conocimiento sobre el estado de los servidores o la SAI, como puede ser el consumo energético, el estado de las baterías, el espacio libre en su memoria, etc.

Los puntos de medición referentes al consumo del CPD, el consumo de cada carga IT, el consumo en la entrada y la salida de la SAI, el consumo de la ventilación y PLC, accesos, alarmas y extinciones se harán mediante sondas de medida. El conocimiento del consumo en los puntos de medición de las PDUs y el conocimiento de la información de cada servidor y la SAI se realizara mediante consultas SNMP.

Una vez capturados los datos serán manipulados mediante una serie de operaciones aritméticas y se enviarán tanto los datos como los resultados de dichas operaciones a

una BBDD para su almacenamiento y poder tener los históricos de las medidas, y a un servicio web para poder ser consultados a través de una web y conocer en tiempo real las medidas.

Una vez se puedan observar los datos a través de una web, estos datos se podrán, ver en tiempo real, mostrar sus históricos, representar gráficas de sus valores y exportarlos en hojas de cálculo. Para complementar esta visualización de los datos, se crearán sinópticos SVG para mostrar una imagen o representación del componente que queremos consultar, juntos con datos consultados a él, en tiempo real.

De tal manera que si visualizamos el sinóptico de un servidor, lo que veremos es una representación física del servidor (propia para cada marca y modelo) y sus datos en tiempo real.

Una vez analizada la funcionalidad que se desea que tenga la herramienta, pasaremos a detallar las funciones que deberán implementarse:

- 1. Obtención del PUE. En todas sus categorías.
- 2. Visualizar los valores de los puntos de medición:
 - i. Dentro de la aplicación cada punto de la adquisición aparecerá representado con una serie de atributos, los cuales son un identificador, una descripción, un tipo de dato (digital o analógico), un valor, un estado (que indica si se esta consultando en tiempo real), comentarios y el origen del mismo.
 - ii. A partir de este momento el usuario ya puede interactuar con ese punto, la forma que tiene el usuario de interactuar con los puntos es seleccionar los puntos que quiere visualizar y decidir de que manera quiere visualizarlo. Las formas que hay de visualizar un punto son:
 - mostrar su valor en tiempo real
 - mostrar su histórico de valores
 - representar una grafica con ese histórico
 - mostrar un intervalo de valores de su histórico
 - mostrar secuencialmente el histórico de un punto
 - exportar los mismos en una hoja de cálculo.
- iii. Búsqueda rápida de "puntos" a través de sus atributos. Ya sea a través del identificador, la descripción, el origen, el tipo de dato o el comentario.
- 3. Visualizar los valores de los puntos de medición de manera interactiva:
 - i. El usuario podrá ver una colección de "puntos" dentro de un sinóptico. Los sinópticos muestran una representación de la distribución física del CPD, datos de sus componentes, PUE calculado y los componentes de medición instalados en el CPD.
 - ii. Para mayor comodidad para el usuario la navegación entre sinópticos se realiza a través de enlaces. Dentro de los sinópticos, se puede seleccionar alguno de los componentes y mostrar el sinóptico que tiene asociado ese

componente, mostrando individualmente ese componente y omitiendo el resto en la representación.

iii. La secuencia que se ha decidido es primeramente mostrar el sinóptico que representa el CPD, seleccionar un rack y mostrar el sinóptico de ese rack, una vez dentro del sinóptico del rack seleccionado, seleccionar un servidor y mostrar el sinóptico de ese servidor.
También se podrá visualizar un sinóptico de manera directa sin necesidad de navegar desde el sinóptico de la representación del CPD.

Tras describir a grandes rasgos el funcionamiento de la aplicación, la siguiente sección proporciona una visión de la metodología que utilizaremos para su desarrollo, así como las justificaciones para la elección de la misma.

3.2. Metodología de desarrollo

Esta sección muestra detalladamente la metodología de desarrollo que será utilizada durante la construcción de la herramienta software.

La Metodología [MET12] es el modelo de proceso o modelo de ciclo de vida utilizado para el desarrollo y define el orden de las tareas o actividades involucradas que deben realizarse para el desarrollo de software. Por tanto una metodología representa el camino a seguir para desarrollar software de manera sistemática.

Entre las más conocidas se puede mencionar: modelo en cascada o secuencial, modelo espiral, modelo iterativo incremental, etc. Comparando las diferentes metodologías, se ha decido realizar la planificación de este proyecto siguiendo la metodología iterativo incremental.

Indicar que los modelos iterativos se utilizan cuando los proyectos sufren alguna de estas tres situaciones:

- Si los requisitos cambian conforme el desarrollo avanza.
- Si las fechas de mercado hacen imposible tener un producto completo y hay que introducir una versión limitada.
- Si los requisitos centrales están bien definidos pero todavía hay que definir detalles de las extensiones del producto.

Conociendo estas características del modelo iterativo y teniendo en cuenta que el proyecto realizado ha tenido una serie de requisitos claramente definidos y otros que no se han definido claramente o que no se han tenido en cuenta cuando se ha realizado el análisis de requisitos o surgido nuevos, el <u>modelo iterativo incremental</u> es idóneo para guiar el proyecto ya que se podrán hacer versiones e incrementar en sucesivas iteraciones estas versiones si aumentan los requisitos (de hecho, ha sucedido mientras se desarrollaba el proyecto).

En definitiva, el modelo elegido facilita la incorporación de nuevos requisitos durante el desarrollo de manera rápida y poco costosa en tiempo y dinero.

Por ultimo indicar que al realizar el análisis de una metodología que guiase el proyecto se tomaron en cuenta una serie de metodologías como el modelo cascada realimentado, el modelo espiral y el modelo de prototipos, las cuales fueron desechadas por sus desventajas sobre proyectos que necesiten de modificaciones en sus requisitos sin costes elevados por ello, inconvenientes sobre la experiencia en desarrollos de proyectos que puede llevarnos a realizar más fallos iniciales y al peligro de realizar prototipos experimentales que estén realizados de manera incorrecta para su posterior mejora y aun así se opte por usar para seguir por ese camino.

3.2.1. Modelo iterativo incremental

El modelo se basa en tener una descripción del sistema, que dará lugar a la especificación y confección de los sucesivos incrementos de las versiones hasta llegar al producto global y final.

Después de haber definido la descripción esencial del sistema, se definirán una serie de actividades concurrentes (especificación, desarrollo y validación), de tal manera que se puedan incrementar las versiones sucesivas y dar así una versión final del proyecto.

De esta manera, se permiten entregas de versiones parciales a medida que se va construyendo el producto final. Es decir, a medida que cada incremento definido llega a su etapa de operación y mantenimiento, incorporará a los anteriores incrementos las funcionalidades y requisitos que fueron analizados como necesarios.



Figura 10: Modelo iterativo incremental. Actividades concurrentes.



Figura 11: Modelo iterativo incremental. Iteraciones

Como se muestra en la *Figura 10*, se aplican sucesivas que produce un incremento y que se entregara un producto con mayor funcionalidad que el previo, hasta alcanzar el software final.

Con este modelo se entrega cada vez un producto parcial pero completamente operacional en cada incremento.

Aplicable cuando los requisitos son medianamente bien conocidos pero no son completamente estáticos y definidos, cuestión esa que si es indispensable para poder utilizar un modelo Cascada.

3.3. Iteraciones y requisitos a nivel de la aplicación

En esta sección se muestran las iteraciones planificadas junto con los requisitos que ha de satisfacer nuestra aplicación en cada iteración, de acuerdo a la descripción del ámbito funcional proporcionada en la Sección 3.1.

Por cada iteración pueden darse, si fueran necesarias cada una de las siguientes fases:

- Análisis de los requisitos de esa iteración,
- Diseño
- Implementación
- Prueba del correcto funcionamiento.

Los requisitos definidos para cada uno de las iteraciones son los siguientes:

- 1. Fase 1. Preparación del entorno de trabajo
 - a. Definición de la ubicación de sondas pasivas en diferentes puntos del layout eléctrico del centro de procesamiento de datos.
 - Instalación de las sondad de medida, que nos den la medida instantánea de las diferentes variables eléctricas (V,A, fase,...), para obtener el consumo.
 - c. Conectar las sondas de adquisición con el PLC de gobierno del edificio.

- d. Recoger las medidas instantáneas de las diferentes sondas de adquisición por medio de un driver basado en el protocolo OPC.
- e. Recogida de consumo eléctrico en elementos activos (SAI y servidores) por medio de un driver basado en el protocolo SNMP.
- 2. Fase 2. Cálculo y representación de PUE
 - a. Realizar el cálculo de PUE instantáneo mediante las funcionalidades de los operadores matemáticos incluidos en "IDbox".
 - b. Representar las medidas históricas de PUE.
 - c. Calcular el PUE por mes y sostenido a lo largo del año, comparándolo con la temperatura exterior del CPD.
- 3. Fase 3. Creación sinópticos
 - a. Representar las medidas instantáneas de consumo eléctrico en un sinóptico dentro de la herramienta de gestión del edificio (basada en la herramienta de desarrollo propio de CIC "IDbox").
 - b. Representar las medidas históricas de los diferentes puntos de adquisición en el CPD:
 - i. Las tarjetas de adquisición y los elementos activos.
 - ii. Las sondas de temperatura (ya existentes) en los diferentes puntos del CPD y del exterior.
 - c. Representar el cálculo del PUE en el diagrama sinóptico del CPD.

3.4. Herramientas utilizadas en el desarrollo de la aplicación

En esta sección se describen las herramientas utilizadas en la creación de la aplicación.

3.4.1. Microsoft Visio

Microsoft Visio [Rot11] es un software de dibujo vectorial para Microsoft Windows. Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, SVG, y más, que permiten iniciar al usuario en los lenguajes de programación.

Aunque originalmente apuntaba a ser una aplicación para dibujo técnico para el campo de Ingeniería y Arquitectura, Microsoft decidió que el futuro del programa residía en el mundo corporativo de los negocios y no en las *mesas de dibujo* de Arquitectos e Ingenieros compitiendo con productos como AutoCad, DesignCad, Microstation, etc. Dicha herramienta se ha utilizado para la creación de gráficos SVG.

3.4.2. PDF2svg

EL programa PDF2svg [PS212] que convertir documentos en formato PDF a archivos SVG. PDF2svg permite una alta calidad de conversión de PDF a SVG, manteniendo el diseño original del documento y conservando los hipervínculos, colores y fuentes. Los archivos resultantes SVG se pueden distribuir, visualizar, editar, almacenar, imprimir, y publicado en sitios web y CD-ROMs. PDF2svg se puede configurar para crear miniaturas de página, así como un resumen XML que describe los componentes del documento, tales como los metadatos, marcadores, anotaciones, etc usando XSLT o cualquier otro procesador de XML.

3.4.3. Google SketchUp Pro

Google SketchUp Pro [Gasp11] es una herramienta para modelado 3D. Donde se pueden importar dibujos, planos de CAD, fotos, imágenes aéreas y otros datos. También permite exportar los modelos 3D realizados en diversos formatos 2D y 3D, como son pdf, jpg, png, bmp, AutoCAD, 3Ds y otros tipos de formatos._Esta herramienta ha sido utilizada para realizar los modelos 2D de los componentes del CPD para su posterior conversión a gráficos SVG.

3.4.4. Adobe Illustrator

Adobe Illustrator [AdoCS11] se trata esencialmente de una aplicación de creación y manipulación vectorial en forma de taller de arte que trabaja sobre un tablero de dibujo.

Es desarrollado y comercializado por Adobe Systems Incorporated y constituye su primer programa oficial de su tipo en ser lanzado por ésta compañía definiendo en cierta manera el lenguaje gráfico contemporáneo mediante el dibujo vectorial. Uso en impresión, vídeo, publicación en la Web y dispositivos móviles. Dicha herramienta fue utilizada para la modificación los gráficos SVG.

3.4.5. Microsoft Visual Studio 2010

Microsoft Visual Studio [May10] [Cag02] es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomuniquen entre estaciones de trabajo, páginas web y dispositivos móviles.

Esta herramienta ha sido utilizada por mi parte para la realización del software de gestión del protocolo SNMP, el cual está escrito en C#.

3.4.6. MG-SOFT MIB Browser 2010

El MG-SOFT MIB Browser es una herramienta que se utilizada para analizar agentes SNMP y observar sus MIBs. Utilizándose dicha herramienta para observar los datos devueltos por consultas SNMP hacia diferentes agentes SNMP.

Hay muchos MIB Browser en el mercado, ya sean gratuitos o de pago. He optado por usar un MIB Browser gratuito ya que la información necesaria para el proyecto me la aporta perfectamente uno gratuito.

Decidí probar dos herramientas y elegir una de ellas para su uso, el criterio para decidirme por uno u otro fue la claridad en la presentación de la información y la facilidad de uso.

Los dos MIB Browser seleccionados fueron:

- iReasoning MIB Browser [iR12]
- MG-SOFT MIB Browser 2010 [MG12]

El elegido fue MG-SOFT MIB Browser 2010 debido a que los dos te dan la misma información pero el uso de este es más sencillo y la interfaz es más amigable, también nos ofrece una ventaja que es cargar MIBs que te puedas bajar de internet y poder darte la OID en nombre y no solo en número.

3.4.7. Herramientas IPMI

El protocolo IPMI es similar al SNMP pero su gran diferencia es que se comunica directamente con un modulo de la placa base llamado BMC. Este modulo BMC es un microcontrolador que gestiona la interfaz entre el software y el hardware, teniendo control sobre los sensores integrados en la placa base. Pudiendo obtener valores de temperatura, alimentación, refrigeración, consumo, etc. Las herramientas IPMI se encargan de realizar consultas IPMI sobre distintos agentes IPMI y obtener sus valores a dichas consultas.

Para poder comprobar este protocolo he utilizado varias herramientas, las cuales son las siguientes:

- IPMIutil Lynux [IPMU12]
- IPMITool Lynux [IPMT12]

3.4.8. Wireshark

Wireshark [Chap12], antes conocido como Ethereal, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica para educación. Cuenta con todas las características estándar de un analizador de protocolos.

Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete. Wireshark incluye un completo lenguaje para filtrar lo que queremos ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP.

Se ha usado para analizar los paquetes de datos del protocolo SNMP, mediante WireShark se capturan los paquetes que salen o van hacia el agente SNMP requerido, seleccionando solo los de tipo UDP y comprobando que la información era la deseada.

3.4.9. Librería SNMPSharpNET

La librería SNMPSharpNet [SSN12] es una librería desarrollada para la plataforma .NET, la cual tiene implementado en lenguaje C# el protocolo SNMP. La librería nos permite utilizar desde una plataforma .NET (Windows) el protocolo SNMP, con toda las funcionalidades de sus versiones, para realizar consultas a diferentes componentes que tengan implementado en él este protocolo y conocer datos de ellos.

3.5. Sumario

Durante este capítulo se ha descrito la planificación del proyecto fin de carrera, indicando el ámbito funcional en el que se encuentra, así como la metodología que se usará. También se describieron los requisitos de alto nivel y las distintas iteraciones de las que se compondrá el proceso de desarrollo.

A su vez se describió el diseño de los componentes más básicos o importantes del proyecto así como las herramientas que se utilizarán para llevarlo a cabo.

Capítulo 4 Preparación del entorno de trabajo

Este capítulo trata sobre la primera iteración del proyecto. En ella se decidirán los puntos de medida del CPD necesarios para el cálculo del PUE y su propio cálculo, se realizará la instalación de las sondas de medida en los puntos de medida que lo requieran, se realizará un prototipo del driver SNMP y su posterior implementación completa. Una vez realizado el driver SNMP de manera completa el resto del proyecto se apoyará en él. También se realizará un refinamiento de requisitos debido a los problemas encontrados a la hora de llevar a cabo la primera iteración.

Por último se mostrarán los casos de uso del driver SNMP y se mostrará el proceso de implementación y pruebas realizadas para comprobar el correcto funcionamiento y cumplimiento de requisitos de esta iteración.

Índice

Capítulo 4 Preparación del entorno de trabaio	
4.1. Objetivos	
4.2. Ingeniería de Requisitos	
4.2.1. Puntos de medida	
4.2.2. Definición del cálculo PUE	
4.2.3. Refinamiento de requisitos	
4.2.4. Instalación de Sondas de medida	
4.2.5. Prototipo SNMP	
4.2.6. Driver de consultas SNMP	
4.2.7. Casos de uso	
4.3. Implementación	
4.4. Pruebas	
4.5. Sumario	

4.1. Objetivos

El objetivo de esta primera iteración es construir la base del proyecto. En ella se decidirán los puntos de medida necesarios para el cálculo del PUE y el propio cálculo del PUE, así como la creación y comportamiento del principal componente de la aplicación, el driver de consultas SNMP.

4.2. Ingeniería de Requisitos

Esta sección engloba todo aquello necesario para la obtención de información del consumo para realizar el cálculo de PUE y conocer el estado de los componentes del CPD. Así como de los requisitos surgidos en esta primera iteración del proyecto

4.2.1. Puntos de medida

En este apartado se indicarán los puntos del CPD donde necesitaré conocer el consumo para el posterior cálculo del PUE. A continuación se mostrarán los puntos de medida elegidos que podemos ver en la *Figura 12*.

Puntos de medida:

- Entrada "GENERAL" en el cuadro eléctrico general del CPD.
- Entrada en el cuadro eléctrico de la ventilación del CPD.
- Entrada de la SAI del CPD.
- Salida de la SAI del CPD.
- Salida de las PDUs del CPD.
- Cargas IT presentes en el CPD.

Para hacer una medición lo más exacta posible del consumo energético se debería conocer este consumo en todos los puntos anteriormente mencionados. Ya que la formula para sacar el valor del PUE se obtendría de la siguiente manera.

PUE= Consumo en Entrada "CPD" / Cargas IT

Conociendo además de esta manera las perdías que se producen en cada punto de medida, ya que tendremos la diferencia entre el consumo total del CPD y lo que se consume en cada punto.

Además de los puntos de medida para llevar a cabo el cálculo del PUE he consultado por SNMP los distintos componentes del CPD que soportan este protocolo (SAI y servidores), de tal manera que obtengo datos de interés de los mismos como por ejemplo el estado de estos o el estado de sus baterías.



4.2.2. Definición del cálculo PUE

En este apartado se indicarán los diferentes cálculos de PUE que existen para un CPD (y cuales que dentro del apartado 4.3 Implementación se utilizan en el proyecto), los cuales varían en función el punto de medición del CPD donde se consulte, del tipo de refrigeración del CPD y del tipo de PDUs que tenga el CPD.

Definición de cálculos ha realizar:

- PUE de categoría 0, el cual se obtiene dividiendo el consumo eléctrico en la entrada GENERAL contando con el sistema de refrigeración por ventilación entre el consumo eléctrico en la salida de la SAI.

PUE0= Consumo eléctrico de GENERAL / Consumo eléctrico de salida SAI.

- PUE de categoría 1, el cual se obtiene dividiendo el consumo eléctrico en la entrada GENERAL contando con todos los sistemas de refrigeración (gas, agua, ,etc) entre el consumo eléctrico en la salida de la SAI.

PUE1= Consumo eléctrico de GENERAL / Consumo eléctrico de salida SAI.

- PUE de categoría 2, el cual se obtiene dividiendo el consumo eléctrico en la entrada GENERAL entre el consumo eléctrico en la salida de la PDU.

PUE2= Consumo eléctrico de GENERAL / Consumo eléctrico de salida PDU.

- PUE de categoría 3, el cual se obtiene dividiendo el consumo eléctrico en la entrada GENERAL entre el consumo eléctrico en la conexión con la cargas IT.

PUE3= Consumo eléctrico de GENERAL / Consumo eléctrico de cargas IT.

4.2.3. Refinamiento de requisitos

Como consecuencia del análisis de los equipos y el layout del CPD surgieron una serie de refinamientos de los requisitos.

Al analizar por separado cada uno de los equipos utilizados dentro del CPD (servidores, racks, PDUs y SAIs) surgieron inconvenientes que no se tuvieron en cuenta a la hora de realizar los requisitos del proyecto, estos inconvenientes surgieron por el desconocimiento de las características propias de cada equipo que se iba a utilizar.

Analizando las PDUs (unidad de distribución) he observado que se tratan de "PDUs tontas", lo que significa que no puedo consultar mediante ningún protocolo la potencia eléctrica que pasa por ellas. Con lo que no podre obtener información en este punto del circuito, que conlleva la imposibilidad de obtener el PUE de categoría 2. Y decido olvidarme de este punto para obtener medidas.

Seguidamente después de analizar las PDUs, analice la SAI. La cual soporta todas las consultas requeridas a ella, pero se decide instalar sondas de medición a su entrada y
salida, al dar datos "poco fiables" en su entrada con relación a la salida (se obtenían valores de potencia mayores a la salida de la SAI que a su entrada).

Al analizar por último los servidores mediante la herramienta MG-SOFT MIB Browser para comprobar la posibilidad de realizar las consultas SNMP necesarias sobre el consumo en cada servidor, compruebo la imposibilidad de recibir datos de consumo a través del protocolo SNMP sobre esos servidores.

Investigando en el sitio web de Dell [DELL12] se indican los modelos y generaciones de los mismos que soportan la consulta del consumo energético. Una vez comprobado los modelos de servidores que soportan la consulta del consumo energético y tiene la empresa, solo los modelos Poweredge 1950, Poweredge R410 y Poweredge R610 nos darán dicha información, pero ninguno cumple la generación que devuelve la información requerida.

Al verme incapacitado para conocer el consumo a través del protocolo SNMP busco protocolos similares que me puedan aportar dichos datos. Como posibles alternativas encuentro el ICMP y el protocolo IPMI.

El protocolo ICMP quedo descartado desde el primer momento porque trabaja igual que el protocolo SNMP y está menos extendido entre los fabricantes.

Solo quedaba la opción del protocolo IPMI. Este protocolo IPMI es similar al SNMP pero su gran diferencia es que se comunica directamente con un modulo de la placa base llamado BMC. Este modulo BMC es un microcontrolador que gestiona la interfaz entre el software y el hardware, teniendo control sobre los sensores integrados en la placa base. Pudiendo obtener valores de temperatura, alimentación, refrigeración, etc. Pero una vez analizado el protocolo en los servidores mediante las herramientas IPMITool y IPMIUtil descubro la imposibilidad de obtener el consumo de los servidores. Por lo que necesitaré otra forma de obtener su consumo, y esta forma es mediante el uso de sondas de medida.

Una vez conocidos los impedimentos surgidos por los equipos del CPD, que nos indican que eliminemos el punto de medida en las salidas de las PDUs y que añadamos una sonda de medida en cada carga IT para conocer su consumo, ahora surge otro inconveniente sobre los puntos de medida del CPD que necesitan una sonda de medida para conocer sus valores. Este inconveniente es el espacio en el cuadro eléctrico y la negativa de la empresa a instalar una sonda en cada servidor para medir el consumo eléctrico de cada servidor del CPD, (*se quiere renovar en un futuro el equipamiento para que bien los servidores nos den los datos de consumo mediante el protocolo SNMP o renovar las PDUs para que sean inteligentes y nos den el consumo en cada entrada de ella*), desechadon la idea de medir el consumo en cada carga IT.

Pasando a medir lo que llamare "perdidas del CPD", estás perdidas nos permitirán obtener "una aproximación" de lo que se consume por las cargas IT en su conjunto. La manera para obtener lo que se consume por las cargas eléctricas será la siguiente:

Ya que el suministro energético de las cargas IT lo proporciona la salida de la SAI, la potencia de las cargas IT se puede calcular de la siguiente manera:

Potencia Cargas IT= Potencia Salida SAI - Potencia "perdidas del CPD"*.

*"Peridas del CPD" son alarmas, extinción, PLC, accesos.

Quedando el requisito de los puntos de medida del CPD, en los siguientes puntos de medida, como se ve en la *Figura 13*:

- Entrada "GENERAL" del cuadro eléctrico del CPD. .
- Entrada del cuadro eléctrico de la ventilación del CPD.
- Entrada SAI.
- Salida SAI.
- "Perdidas del CPD", puntos de gasto de energía de no computo (alarmas, extinción, PLC, accesos).

Para el cálculo del PUE se ha analizado la refrigeración y PDUs del CPD, dando cuenta de que no todos las categorías de PUE pueden ser calculados debido a la disposición del CPD de la empresa CIC Consulting Informático. Pasando así a indicar que el PUE de categoría 1 no se puede calcular, porque es para CPDs con refrigeración por agua u otros métodos, donde el CPD de esta empresa no se encuentra. Y como ya se indico en este mismo apartado, el PUE de categoría 2 necesita PDUs inteligentes que nos den su consumo y no las tenemos, por lo que no calcularé la categoría 2 del PUE.

4.2.4. Instalación de Sondas de medida

La instalación de las sondas de medida se ha guiado de la distribución de los puntos de medida del layout del CPD que se muestra en la *Figura 13*, que corresponde con los requisitos refinados en la *sección 4.2.3 Refinamiento de requisitos* de este mismo capítulo.



Figura 13: Layout CPD

Antes de indicar el proceso de instalación de las sondas de medida, indicar que los servidores del CPD tienen doble línea de alimentación, lo que significa que una línea de alimentación se recibe directamente desde la compañía eléctrica y la otra desde una SAI. Al tener esta doble línea de alimentación el cuadro eléctrico del CPD se puede dividir en dos zonas, la "zona protegida" y la "zona no protegida".

Esta zona protegida es la que se encarga de recibir la electricidad desde la SAI, haciendo posible que si se cae la luz, los componentes conectados a ella no dejen de funcionar. Permitiendo desconectar distintas partes del CPD para instalar las sondas de medida sin necesidad de apagar el CPD por completo.

De la *Figura 13* indicar que la "sonda 3", perteneciente a la salida de la SAI ya está instalada antes de comenzar el proyecto.

La "sonda 1" será modificada cambiando el toroide existente por el tipo de toroide indicado en la *sección 2.2 dentro del capítulo 2*, por razones internas de la empresa.

La "sonda 2" nos ofrece un nuevo problema, este problema es debido al poco espacio dentro del cuadro eléctrico para instalar los toroides. La zona donde irían instalados estos toroides esta muy congestionada de cables y debemos cambiar estos cables para sustituirlos por unos más largos que nos permitan cambiar la ubicación de los toroides a una zona menos congestionada.

La "sonda 4" se sitúa dentro de un cuadro eléctrico externo al cuadro eléctrico del CPD. Este cuadro eléctrico es exclusivo para la ventilación y las sondas, como ocurre en

la sonda 1 se sustituirán por unas con el nuevo modelo de toroide debido al mismo problema.

La "sonda 5" ha sido la más problemática debido a la distribución del cuadro eléctrico. La alarma, PLC, extinción y accesos están controlados por dos relés eléctricos, acceso y PLC por uno y alarmas y extinción por el otro. Cada relé se alimenta de una fase diferente y dificulta la medición de las mismas, con lo que se buscará cambiar esta distribución para que los cuatro puntos reciban de la misma fase y así instalar un toroide que nos mida los cuatro puntos a la vez

Todos estos toroides van conectados a una tarjeta de adquisición que nos permite posteriormente su consulta a través de él como se indica en la *sección 2.2 dentro del capítulo 2.*

4.2.5. Prototipo SNMP

En esta sección se detalla la construcción de un primer prototipo del driver de consultas SNMP que se utilizará para obtener los datos de los componentes a través del protocolo SNMP, donde se investigará que solución tomar para realizar las consultas de la mejor manera y si todos los agentes SNMP soportan las consultas deseadas.

4.2.5.1. Prototipo driver SNMP

Para la creación del prototipo que nos servirá para realizar las consultas SNMP, necesitaremos hacer uso de la clase *PeticionSNMP (la cual se explica su funcionamiento en el Capitulo 4 dentro del apartado 4.2.6 Driver de consultas SNMP),* el uso de esta clase se hará a través de una clase *Prueba* que consiste en tener un método Main que creará una lista con las OIDs que queremos consultar y obtendrá el resultado de las consultas sobre cada OID en una lista para su posterior visionado por pantalla.

Para el aprendizaje de la asociación entre la OID y el valor que devuelve (saber que OID preguntar para que devuelva el dato requerido) ha sido una labor de aprendizaje a través de la herramienta MIB Browser y mediante la especificación de las OID del fabricante del componente que se quiere consultar.

El siguiente código muestra un ejemplo de uso de la clase Prueba:

Para la comprobación de las consultas sobre las OIDs requeridas utilizaré el programa WireShark el cual nos permite capturar paquetes de red que salgan o vayan hacia el componente donde se realizó la consulta. Este programa WireShark nos permitirá ver el contenido de los paquetes y de esta manera ver si la información contenida en él es la correcta, este proceso se detalla dentro del *Anexo 3 Uso de WireShark*.

Otra manera de comprobar las consultas realizadas es mediante la consulta WALK, esta consulta realiza un barrido de toda la MIB del componente que estamos consultando a partir de una OID indicada, de esta manera veremos toda la información que devuelve el componente y si es capaz de devolver dicha petición. El siguiente código muestra cómo se realiza una consulta WALK:

4.2.6. Driver de consultas SNMP

En esta sección se explicará uno de los componentes más complicados de entender del proyecto. En la explicación se detallan los componentes que forman parte del driver y del proceso de actuación entre ellos.

Antes de comenzar la explicación, indicar que el driver está realizado en la empresa CIC Consulting Informático y implementado para ser ejecutado como un servicio Windows. El código está realizado mediante el lenguaje C# y utilizando la librería SNMPSharpNet. Con este driver se obtendrán datos de interés del estado de cada componente del CPD que soporte el protocolo SNMP.

El driver s se componen de tres partes diferencias, las cuales son:

- I. SIDP.Core.Driver.SNMP
- II. SIDP.Service.SNMP
- III. WindowsServiceSetup.SNMP

La primera de las partes, *SIDP.Core.Driver.SNMP* (que se considera como la parte básica del driver), se encarga de establecer una comunicación con el sistema de adquisición de datos. En este caso se utiliza el protocolo SNMP para realizar las consultas a los diferentes componentes (agentes SNMP). Esta parte se encarga de gestionar la conexión, solicitud, recepción de datos y el posterior tratamiento para almacenar y enviar los datos a la "Topología" y a los agentes de datos de la herramienta "IDbox".

La segunda de las partes, *SIDP.Service.SNMP*, permite que el driver realizado en la parte SIDP.*Core.Driver.SNMP* sea un servicio de Windows. Y por último *WindowsServiceSetup.SNMP* se encarga de realizar un archivo instalador con extensión "*.msi*" encargado de instalar el servicio Windows en la maquina donde se ejecute.

El comportamiento de cada servicio del driver, depende de unos ficheros de configuración en formato XML, que deben ser editados e incluidos durante la instalación, para ello tienen que estar ubicados en una subcarpeta "*Conf*" de cada servicio. Estos ficheros especifican la configuración de la conexión entre el Driver y los agentes SNMP, así como la configuración del grabador de transitorios (el agente de

datos de IDbox contiene una caché en la que se guardan datos, a los caules llamamos transitorios) y la configuración de los paquetes UDP que se envían.

Para entender globalmente el driver tendré que pasar a explicar previamente estos ficheros de configuración, los cuales se llaman:

- *I. SNMP.xml*, se encarga de indicar la configuración de la conexión con el agente.
- *II. PIDConfig.xml, se encarga de indicar las OID de cada agente que se quiere consultar.*
- *III. UDP.xml, se encarga de indicar la configuración de los paquetes UDP.*
- *IV. Transitorio.xml, se encarga de indicar la configuración del grabador de transitorios.*
- V. DriverService.xml, se encarga de establecer la conexión entre el Driver y el UPDsender.

Como ejemplo de la configuración de uno de estos archivos paso a especificar en detalle el fichero de configuración SNMP.xml, teniendo una estructura similar a la siguiente:



Donde los parámetros significan lo siguiente:

SNMPConfig es la etiqueta padre e indica quien será el componente principal indicado con un identificador id. Bajo la etiqueta SNMPConfig se deben realizar la declaración de todos los elementos que vayan a ser consultados, donde los parámetros significan lo siguiente:

Para realizar la declaración del tiempo entre consultas a los elementos se debe poner entre las etiquetas Periodo></periodo> el periodo deseado. A continuación para realizar la configuración de la información necesaria para establecer una conexión con el agente SNMP se realiza dentro de las etiquetas PLCS></plcS>. Cada conexión se declara dentro de las etiquetas <PLCSNMPConfig></PLCSNMPConfig> y dentro de ellas se declararán los atributos de cada conexión, las cuales están definidas por seis campos: Enable, SNMPTimeOut, SNMPRetry, SNMPComunit, SNMPVersión y Endpoint. Donde Enable indica si está activa o no dicha configuración del elemento, SNMPTimeOut indica el tiempo d esperar para declarar que ha habido un fallo en la conexión, SNMPRetry que indica los reintentos ante fallo en la comunicación, SNMPComunit que indica la comunidad del agente SNMP, SNMPVersión que indica la versión del protocolo a utilizar y Endpoint que indica la ip y puerto del agente SNMP al que realizar las consultas.

4.2.6.1. Diagrama de Clases

En este apartado se describirá el diagrama de clases pertenecientes a *SIDP.Core.Driver.SNMP*, que es la parte del driver que contiene clases utilizadas para la captura de datos, es el que se muestra en la *Figura 14*.



Figura 14: Diagrama clases driver

En el diagrama se tienen una serie de clases que nos permiten como anteriormente he mencionado la consulta a los agentes SNMP y enviar los datos devueltos a la "Topología" y agentes de datos de la herramienta "IDbox".

Para un mejor entendimiento pasaré a explicar el proceso global del conjunto de clases y posteriormente la información más reseñable de cada clase.

El proceso para la adquisición de datos requeridos y el posterior envío hacía la "Topología" y los agentes de datos comienzan con la creación de la clase *PeticionSNMP*.

La clase *PeticionSNMP* nos permite realizar la parte más importante del driver. Nos permite conectarnos, consultar y desconectarnos al agente SNMP mediante los métodos

conectarUDP(), obtenerValoresOID() y desconectarUDP() respectivamente. Esta clase utiliza la librería SNMPSharpNet para gestionar todas las consultas SNMP.

La conexión con el agente SNMP se realiza a través del protocolo SNMP utilizando paquetes UDP(IPv4). Para realizar una conexión con un agente SNMP se necesita una serie de valores entre los cuales destacamos la ip y puerto del agente.

Para la consulta de las OIDs requeridas para cada agente SNMP se creará un paquete SNMP de consulta, creando una consulta GET para cada OID. Y se recibirá el resultado de todas estas consultas en una lista para tener todos los valores recogidos.

La clase *SNMPBaseDriver* se encarga de hacer la configuración del driver, de esta manera leyendo datos de una serie de archivos de configuración conseguiremos obtener la información necesaria para la clase *PeticionSNMP* y la información de que OIDs consultar.

Ahora como ya tenemos la información sobre como conectarnos al elemento y que preguntar, pasaremos a utilizar el método conectarUDP() de la clase *PeticionSNMP* dentro de la clase *SNMPBaseDriver* a través del método conectar().

A través del método LeeDatos() de la clase *SNMPBaseDrier*, que llama internamente al obtenerValoreOID() de la clase *PeticionSNMP*, devolverá un bloque con el instante de obtención del dato y el valor del dato. El método desconectar() desconecta la conexión con el elemento mediante el método desconectarUDP() de la clase *PeticionSNMP*.

La clase *SNMPDriver* se encarga de crear un objeto de la clase *SNMPBaseDriver* (la clase previamente explicada) para la obtención del valor de las OIDs deseadas. Esta clase contiene una llamada para enviar los datos a la "Topología" y poder trabajar con ellos para después ser enviados a una BBDD para almacenarlos o mostrarlos vía web.

La clase *SNMPMultiDriver* se encarga de realizar una inspección sobre una lista de agentes SNMP (presentes en el fichero de configuración SNMP.xml), a los que se quiere consultar alguna OID, creando para cada agente un Thread que consultara para cada agente sus OIDs requeridas. Esta clase contiene una llamada para enviar los datos a la "Topología" y poder trabajar con ellos para después ser enviados a una BBDD para almacenarlos o mostrarlos vía web.

Para realizar la lectura de los archivos de configuración se utiliza una serie de clases, las cuales llamo "clases auxiliares" porque no interactúan directamente con las consultas SNMP. Las cuales son *LoadConfig, SNMPConfig* y *DriverSNMPPIDAttribute*.

4.2.7. Casos de uso

Los casos de uso de la primera iteración corresponden a las acciones de obtener valores de las OID requeridas de un agente SNMP y obtener la MIB de un agente SNMP mediante el driver de consultas SNMP.



Figura 15: Casos de uso

En el primer caso de uso el usuario debe poder elegir el agente SNMP y las OIDs a las que quiere consultar sus valores. Para realizar esto el Usuario deberá conocer la comunidad del agente SNMP, su dirección IP, puerto por el cual consultar y la versión SNMP que soporta el agente SNMP y elegirá el tiempo de espera ante fallos y el número de reintentos de la consulta.

En el segundo caso de uso el usuario debe poder elegir el agente SNMP sobre el que quiero saber su MIB y la OID desde la cual empiezo a consultar la MIB del agente. Para realizar esto el usuario deberá conocer la comunidad del agente SNMP, la dirección IP y la versión SNMP que soporta el agente SNMP.

4.3. Implementación

En esta sección se describe el proceso de implementación de la primera iteración, de manera que se satisfagan los requisitos descritos en el apartado 4.2.4. *Refinamiento de requisitos*.

La instalación de las sondas de medida ha llevado mi supervisión, lo que quiere decir que he sido el encargado de decidir los puntos de medición del CPD que van a ser monitorizados mediante el uso de sondas de medida. Y el supervisor de su configuración.

Mediante las sondas de medida podemos obtener los datos de la potencia energética en los puntos del CPD descritos en los requisitos. Estas sondas de medida volcarán sobre un servidor OPC los valores obtenidos y un driver OPC se encargará de leer estos valores y enviarlos a la herramienta "IDbox" para su posterior uso. El driver OPC no ha sido desarrollado por mí y su configuración la han realizado empleados de la propia empresa poniendo a mi disposición los datos dentro de la herramienta "IDbox".

Para la implementación del driver de consultas SNMP se crearán las clases descrita en el diagrama de clases que se muestra en la *Figura 14* que nos permitirá tener instalo en la máquina que deseemos un servicio Windows que consultará ininterrumpidamente, y con el período de consulta que deseemos a los agentes SNMP que decidamos, obteniendo el valor de las OIDs solicitadas. Y enviándolos a la herramienta "IDbox" para su posterior manipulación.

4.4. Pruebas

En esta sección se relatan las pruebas realizadas para la comprobación del cumplimiento de los requisitos descritos.

Una vez concluida la fase de implementación, realizamos las pruebas necesarias para comprobar que los requisitos que se marcaron en la primera iteración están cumplidos.

Primeramente indicar que el requisito de decidir los puntos de medida del layout del CPD no puede ser comprobado, no hay un correcto o incorrecto funcionamiento. Sólo indicar que los puntos de medida cumplen los requisitos para poder obtener el PUE. Sucediendo de igual manera con la definición del cálculo del PUE.

Pasaremos ahora a la ejecución de la aplicación para comprobar el cálculo del PUE y el correcto funcionamiento del driver. Para ello ejecutamos la aplicación y observamos si los resultados al ejecutar distintas acciones corresponden con los esperados.

 Acción: Elegir la función Obtener valores OID. Resultado Esperado: Aparición de los resultados de las consultas a las OID en un archivo LOG. A esta escritura de los resultado de las consultas OID se le añade información sobre la conexión y desconexión con los agentes SNMP. Posteriormente se envía esta información de los resultados de las consultas a las OIDs a la "Topología" y Agentes de datos de la herramienta IDbox.

Resultado Obtenido: El Resultado Esperado.

- Acción: Elegir la función Obtener valores MIB. Resultado Esperado: Aparición de los resultados de la consulta sobre las MIB en un archivo de texto(LOG). Resultado Obtenido: El resultado esperado.
- 3. Acción: Comprobar la correcta recepción de los valores medidos con las sondas de medida.

Resultado esperado: Aparición de los resultados en un archivo de texto(LOG). A esta escritura de los resultados de las sondas de medida se le añade información de la conexión OPC. Posteriormente se envía esta información a la "Topología" y Agentes de datos de la herramienta "IDbox".

Resultado obtenido: El resultado esperado.

4.5. Sumario

En este capítulo se habló sobre las acciones que se realizaron durante la primera iteración del proyecto. Se indicaron los puntos de medida del CPD, los cálculos necesarios para el cálculo del PUE, la creación del driver de consultas SNMP los casos de uso del driver de consultas SNMP, la configuración del driver OPC y los nuevos requisitos encontrados.

También se indicó la forma de implementar los componentes y funcionalidades correspondientes a esta iteración y finalmente se muestran las pruebas realizadas para comprobar el correcto desempeño de las acciones.

Capítulo 5 Cálculo del PUE

Este capítulo trata sobre la segunda iteración del proyecto. Con ella se realiza la configuración del elemento necesario para indicar la "Topología" necesaria de la herramienta "IDbox", para definir los cálculos a realizar para la obtención del PUE, este elemento es el "Administrador" de "Topología".

A continuación se mostrará el proceso de implementación, y finalmente las pruebas realizadas para comprobar el correcto funcionamiento y cumplimiento de los requisitos.

Índice

Capítulo 5 Cálculo del PUE	
5.1. Objetivos	
5.3. Implementación	
5.4. Pruebas	
5.5. Sumario	

5.1. Objetivos

En esta segunda iteración, la aplicación se encuentra a la mitad de su implementación, con la mitad de sus requisitos ya resueltos. En la que se añadirá la realización del cálculo del PUE en sus categorías cero y tres.

A partir de esta segunda iteración el resto de iteraciones consistirán en ir añadiendo funcionalidades visuales a la aplicación de manera que se vayan cumpliendo los requisitos.

Para realizar el cálculo del PUE necesitaremos una herramienta que nos permita trabajar con los datos capturados por el driver OPC anteriormente mencionado. Esta herramienta es la llamada "Administrador", la cual nos permite diseñar una "Topología" para los datos con los que trabajemos. El "Administrador" nos facilita la tarea de trabajar con los consumos en los puntos de medida del CPD y definir las operaciones a realizar para obtener el PUE en sus categorías cero y tres.

Las operaciones a realizar son:

- Sumar la potencia de las fases de la SAI para obtener su potencia total.
- Restar la potencia obtenida de la SAI con la potencia de "perdidas del CPD". De esta manera obtenemos la potencia de las cargas IT.
- Dividir la potencia del CPD entre la potencia de las cargas IT.

Quedando como resultado las siguientes formulas a resolver.

PUE0 = consumo eléctrico del CPD / (consumo Salida SAI)

PUE3 = consumo eléctrico del CPD / (consumo Salida SAI - consumo "perdidas del CPD)

5.3. Implementación

Para realizar la configuración de los cálculos a realizar para la obtención del PUE y el tratamiento de los diferentes puntos de información que tenemos del CPD, tendremos que utilizar para ello el "Administrador" el cual nos permite realizar estas necesidades.

Para la configuración necesaria, se ha llevado a cabo la creación de un contenedor en el que se detallan todos los elementos necesarios para obtener los datos de los puntos del CPD capturados y el posterior cálculo del PUE con estos valores.

Dentro de este contenedor se sitúan los componentes:

- ReceiverUDP, el cual se encarga de recibir los datos de los puntos de medida que se envían desde el driver OPC.
- Sumador, el cual se encarga de realizar la suma de las potencias de salida en cada fase de la SAI para obtener la potencia total de salida.
- Restador, el cual se encarga de restar la potencia de salida de la SAI con la potencia de "perdidas del CPD". Para así obtener la potencia de las cargas IT.
- Divisor, el cual se encarga de dividir el consumo del CPD y el de las cargas IT.
- GrabadorHistorico, el cual se encarga de grabar la información en una BBDD para tener sus valores históricos.

De tal manera que el proceso que sufren los datos es ser recibidos en el RecevierUDP, pasando a poder ser utilizados a la salida de este componente por cualquier otro componente. Estos datos recibidos tendrán que ser asociados con un Punto, el cual es la asociación de un PID (identificador) a un valor y la indicación del tipo de datos (analógico o digital).

Una vez se tienen los datos disponibles para trabajar con ellos, los datos que no necesiten ser manipulados (los provenientes del driver de consultas SNMP) irán a la entrada del GrabadorHistorico que se encargará de guardar estos valores en una BBDD para su posterior consulta.

Los datos que tengan que se manipulados por otro componente, como es el caso del componente Sumador, Restador y Divisor irán a la entrada de éstos y sufrirán la conversión a la operación requerida dando en su salida el resultado de la misma. Esta salida irá al GrabadorHistoricos que guardará estos datos en una BBDD.

5.4. Pruebas

En esta sección se relatan las pruebas realizadas para la comprobación del cumplimiento de los requisitos descritos en este capítulo.

Una vez concluida la fase de implementación, realizamos las pruebas necesarias para comprobar que los requisitos que se marcaron en la primera y el objetivo de la segunda iteración estén cumplidos. Para ello ejecutamos la aplicación y observamos si los resultados de las operaciones son los esperados.

Para la observación de los resultados de las operaciones se tiene que hacer uso de la herramienta web de "IDbox". Esta herramienta nos permite visualizar tanto los puntos de medida del CPD y cálculo del PUE. Una vez que se hagan los cálculos, la "Topología" se encarga de enviar los datos al agente de datos que a su vez son consultados por la herramienta web para visualizar estos datos.

Las pruebas ha realizar quedarían encuadras en ver los datos "en bruto" y ver el resultado de las operaciones realizadas con ellos y comprobar que sean correctos estos cálculos con lo definido en el 'Administrador'. Dando como resultado unas pruebas satisfactorias a las operaciones realizadas para llevar a cabo el cálculo del PUE a través del "Administrador".

5.5. Sumario

En este capítulo se habló de la segunda iteración del proyecto, así como las decisiones que se tomaron durante la misma. Con esta iteración finaliza el proceso de captura y cálculo de información necesaria para el proyecto. En las siguientes iteraciones se añadirán funcionalidades al proyecto, como son la visualización de los datos a través de sinópticos y la actuación frente a un PUE negativo.

Durante este capítulo se indican las decisiones tomadas durante la implementación de esta iteración y las pruebas realizadas con el objetivo de comprobar el correcto funcionamiento.

Capitulo 6 Creación sinópticos

Este capítulo trata sobre la tercera y última iteración del proyecto. En ella se realiza la parte visual del proyecto dentro de la herramienta web de "IDbox", esta parte visual comprende una mejora para la visualización de los datos por parte del usuario.

Se realizarán sinópticos SVG para mostrar una representación de un modelo 3D en un modelo 2D que permitirá representar la disposición física del CPD junto con información de interés de cada agente SNMP, puntos de medida y los cálculos del PUE.

En este capítulo se muestran los nuevos requisitos. A continuación se mostrará el proceso de implementación, y finalmente las pruebas realizadas, para comprobar el correcto funcionamiento y cumplimiento de los requisitos.

Índice

Capitulo 6 Creación sinópticos	
6.1. Objetivos	
6.2. Ingeniería de Requisitos	
6.2.1. Refinamiento de requisitos	
6.3. Implementación	
6.3.1. Proceso de creación de sinópticos	

6.1. Objetivos

Esta iteración corresponde a la última planificada para el desarrollo de la aplicación. La cual se centra en la visualización de los datos con los que se trabaja en el proyecto, los cuales son los datos de interés de los agentes SNMP, puntos de medida del CPD y cálculos del PUE.

La visualización se realiza dentro de la herramienta web de "IDbox" y se centra en la realización y visualización de sinópticos SVG para mostrar una representación de un modelo 3D en un modelo 2D que permitirá representar la disposición física del CPD junto con información de interés de cada agente SNMP, puntos de medida y los cálculos del PUE.

Con la realización de los sinópticos SVG se busca hacer más amigable al usuario la manera de visualizar la información, de tal manera que los sinópticos aparte de permitirnos ver la disposición física del CPD y de la información del PUE y datos de los agentes SNMP nos ofrezca la posibilidad de mostrar alarmas sobre distintos estados, tanto de los agentes SNMP como de los puntos de medida y cálculos del PUE. Estas alarmas se muestran con un código de colores y texto que nos indican la importancia de esa alarma.

Básicamente esta es una iteración cuya finalidad es cambiar la apariencia básica de la interfaz gráfica, que hasta ahora nos muestra valores numéricos de los distintos puntos

de medida del CPD, por una más atractiva que nos aporte información adicional a los simples valores numéricos.

6.2. Ingeniería de Requisitos

En esta sección se muestra el refinamiento de requisitos, así como la aportación de nuevos. En esta iteración no se añade ninguna nueva funcionalidad, por lo que no se precisa de un diagrama de casos de uso. En cambio se añaden algunos requisitos y se refinan otros para lograr que la parte visual del proyecto sea visualmente atractiva y con ello lograr el objetivo de esta iteración.

6.2.1. Refinamiento de requisitos

Para lograr cumplir con los objetivos de esta iteración se definieron una serie de nuevos requisitos. Anteriormente los requisitos sobre sinópticos se centraban en realizar sinópticos de las medidas instantáneas en el CPD y un sinóptico de los cálculos del PUE.

Con motivo de dar un valor añadido a la herramienta he decidido aportar sinópticos sobre la disposición física del CPD, sinópticos de la SAI, sinópticos de cada rack del CPD y sinópticos de cada servidor del CPD, en los cuales se mostrarán los datos obtenidos por SNMP o sondas de medida que aportan información sobre el estado de esos componentes.

Para hacer más fluida la navegación entre sinópticos se enlazan de tal manera que seleccionando un componente dentro de un sinóptico se accede al sinóptico de ese componente.

6.3. Implementación

En esta sección se describe el proceso de implementación de la tercera iteración así como las decisiones tomadas durante esta etapa, de manera que se satisfagan los requisitos descritos en la sección anterior.

Al comenzar esta iteración ya se dispone de una aplicación completamente funcional pero con una herramienta web que sólo aporta información numérica sin resaltar la importancia de la misma. Por lo que en esta iteración sólo se limita a mejorar el apartado visual de la aplicación como ya se ha expresado en las dos secciones anteriores.

Para la mejora visual he utilizado sinópticos para mostrar la información del PUE y la información de los componentes del CPD. Para ello se ha he decido hacer un modelo 2D de los objetos que se monitorizan y mostrar en este mismo modelo los datos en tiempo real de los puntos de medición.

En la empresa CiC Consulting Informático se tienen una herramienta web llamada "IDbox" que tiene ya implementado el modo de visualización deseado por la empresa y

se seguirá esta política de la empresa. El método consiste en tener gráficos SVG[Cag02] que se visualizarán desde una web, mostrando los datos en tiempo real de los puntos de medida del CPD y agentes SNMP del CPD.

El proceso de creación de los sinópticos consistirá en hacer un modelo 3D que representa el objeto físico del que se van a sacar los puntos de medida y posteriormente convertirlo a formato PDF para su posterior conversión a un archivo SVG.

Indicar que los datos de interés que se mostrarán en los sinópticos para los diferentes componentes son:

Para los servidores: estado general del servidor, estado de su batería, posibilidad de usar su batería, estado de sus discos y espacio libre en disco.

Indicar que no todos los servidores soportan la monitorización de estos valores debido a sus limitaciones. Ya que dependiendo del modelo de servidor nos aporta unos datos u otros.

Para la SAI: temperatura, estado de la batería, porcentaje de la batería cargada, tiempo restante de batería, tiempo sin usar la batería.

Para el CPD: temperatura del CPD (pasillo caliente/pasillo frio) y PUE.

Una vez explicado como será la disposición de los gráficos SVG paso a explicar el proceso de realización de gráficos SVG.

6.3.1. Proceso de creación de sinópticos

El proceso para realizar los graficos SVG consta de tres pasos, en el que el primero es generar un modelo 3D del objeto u objetos mediante la herramienta Google SketchUp Pro y se procederá a exportarlos en formato PDF.



Una vez exportado a PDF se visualizara como se muestra en la Figura 16.



Figura 16: Proceso SVG 2

El segundo paso consistirá en transformar el formato PDF a SVG mediante el uso de PDF2svg. Realizandolo a través de línea de comando y cuyo formato es el siguiente:

> pdf2svg -o "folder_Out" "folder_In\Name_File". > pdf2svg -o "c:\My Output" 1.pdf

Una vez generado el SVG resultante, se pasa a comprobar el resultado, visionándolo en un navegador.



Figura 17: Proceso SVG 3

Cuando ya se ha generado el archivo SVG, se dará el tercero de los pasos manipulando el archivo SVG dentro de la herramienta Adobe Illustrator para añadirle texto y formas geométricas que en un futuro cambiarán de forma y color dependiendo de los valores de las señales que estemos monitorizando.



Figura 19: Proceso SVG 5

Cuando obtenemos el svg cambiado por el Adobe Illustrator, pasamos al visionado del svg y comprobar que está de forma correcta, si el svg no fuese como deseamos, pasaría a cambiarlo manualmente a través del programa "Notepad++" donde veremos el archivo svg, "línea a línea", donde podemos cambiar sus atributos, ya sea su posición en el svg, color, etc.

Una alternativa para crear los gáficos SVG, es usar la herramienta Visio, que tiene modelos de servidores, racks, router, Lan y otros, de forma predeterminada y exporta directamente a SVG. Lo único que necesitaremos es bajarnos el modelo de la red, http://www.visiocafe.com/, y colocar los rack y servidores de la manera que deseamos. La única pega es que no están en los tamaños requeridos (se tendrán que modificar) y solo están disponibles los frontales, con lo que la apariencia de los objetos siempre será en 2D.

6.4. Pruebas

En esta sección se relatan las pruebas realizadas para la comprobación del cumplimiento de los requisitos descritos en la sección "6.2.1. Refinamiento de requisitos".

Una vez concluida la fase de implementación, realizamos las pruebas necesarias para comprobar que todo funciona correctamente. Para ello ejecutamos la aplicación completa y observamos si los resultados al ejecutar distintas acciones corresponden con los esperados.



Figura 20: Sinóptico SVG

Las pruebas realizas para la comprobación se basan en verificar que se muestren las alarmas de los sinópticos en función de la configuración realizada en los archivos svg. Estas alarmas irán acompañadas de un color, texto y valor que indique el estado de un atributo del componente y se comprobará que dependiendo del valor, la alarma asociada es la definida en el archivo svg y por tanto la correcta.

A su vez se ha comprobado el funcionamiento de la navegación entre sinópticos dando como resultado unas pruebas satisfactorias.

6.5. Sumario

Durante este capítulo se trató las acciones realizadas en la tercera y última iteración del proyecto. En ella se cambia la parte visual de la aplicación.

Para ello se mostraron los nuevos requisitos encontrados y las modificaciones sobre los existentes, las decisiones que se tomaron durante la fase de implementación y finalmente las pruebas que se llevaron a cabo para comprobar el correcto cambio visual de la aplicación.

Capítulo 7 Despliegue y aceptación

Este capítulo trata sobre el proceso de despliegue y aceptación de la aplicación desarrollada. Para ello se describe el proceso utilizado para la creación de instaladores para los distintos sistemas operativos, así como los problemas encontrados durante el proceso.

Índice

Capítulo 7 Despliegue y aceptación	49
7.1. Despliegue de la Aplicación	49
7.2. Aceptación de la Aplicación	50
7.2.1. Instalación en un equipo del servicio Windows	50
7.2.2. Uso Usuario	50
7.3. Sumario	50

7.1. Despliegue de la Aplicación

En esta sección se muestra como se realizará el despliegue de la aplicación en una maquina que actuará de esta manera como una estación de administración SNMP y estación de consulta al servidor OPC.

El despliegue de la aplicación consistirá en crear un instalador Windows que nos permita instalar en una computadora de la empresa CIC (la cual ya tiene instalado el driver OPC) un servicio Windows que contiene el uso del driver SNMP y realizar la configuración del driver OPC.

Mediante el uso de un instalador de un servicio Windows se facilitará el uso en distintos computadores y la instalación del mismo pues el propio instalador se encargara de elegir el lugar donde instalar el servicio y de donde obtendrá los datos necesarios para su funcionamiento.

Para que el servicio Windows funcione correctamente tendrá que contener en la carpeta de instalación del mismo las librerías utilizadas por parte del driver SNMP para su correcto funcionamiento y una carpeta "Conf", que contendrá los archivos de configuración escritos en XML. En esta misma carpeta del servicio se creará una carpeta "Logs" que contendrá información del proceso de captura de los datos y del envió de los mismos a la "Topología" y a los agentes de datos.

Para el uso de un agente SNMP por parte del servicio Windows antes mencionado, este agente deberá tener instalado el protocolo SNMP para poder comunicarse con él. Dicha instalación es realizada por el departamento de soporte de la empresa CIC Consulting Informático. Indicar que el driver OPC es un driver aportado por la empresa CIC Consulting Informático el cual se encarga de consultar un servidor OPC y obtener los datos del mismo para su posterior visualización en la herramienta "IDbox". Este driver ha sido probado en la primera iteración del proyecto pero no explicado su funcionamiento debido a que no está realizado dentro del proyecto. Para su uso tendré que configurar los archivos de configuración XML, del cual obtiene los datos de consultas requeridos.

De esta manera los archivos de configuración necesitan el servidor OPC al cual conectarse para realizar las consultas OPC (en este servidor las sondas de medida vuelcan los valores obtenido), el nombre del grupo de los datos y el periodo entre consultas a este servidor.

Al igual que ocurre con el driver de consultas SNMP el driver OPC tiene un archivo de configuración encargado de asociar a cada consulta un PID, el cual se enviará más a la herramienta "IDbox".Y mediante el archivo UDP.xml configurar el paquete UDP que recibirá dicha herramienta.

7.2. Aceptación de la Aplicación

Esta sección trata sobre la implantación del software desarrollado en los equipos finales. El software se instaló en una maquina con Windows server 2003.

7.2.1. Instalación en un equipo del servicio Windows

Para la instalación del servicio Windows simplemente tendremos que ejecutar el instalador de dicho servicio y seguir las indicaciones que se muestran por pantalla para llevar con éxito su instalación. Las indicaciones mostradas serán si se desea cambiar la ruta de instalación del servicio y si estamos de acuerdo con estos cambios. Por defecto el instalador tiene una ruta de instalación "C:*CiC\WindowsServiceSetup.SNMP*"

7.2.2. Uso Usuario

Como se trata de una aplicación interna pero con intención de venderla en un futuro, no se ha podido testear sobre usuarios finales externos a la empresa. Con lo que el encargado de decidir si la aplicación es de agrado y correcta es del director del proyecto en la empresa CIC Consulting Informático, el cual quedo satisfecho con el desarrollo del proyecto así como con la propia aplicación.

La aplicación se adapta a la forma de uso esperada ya que las pautas a seguir para conseguirlo fueron dadas por él. La aplicación responde como se esperaba, ofreciendo una parte gráfica agradable y amistosa para el usuario. Mostrándose contento, por disponer finalmente de una herramienta atractiva y potente, destinada al fin de monitorizar y visualizar la información del cálculo del PUE, información del consumo del CPD y de información del estado de cada agente SNMP del CPD en una misma herramienta a diferencia de la situación actual.

7.3. Sumario

Durante este capítulo se explica cómo se realizó el proceso de creación de un instalador para un servicio Windows y su instalación. La configuración del Driver OPC y finalmente la aceptación de la aplicación por parte de los usuarios finales.

Capítulo 8 Conclusiones y trabajos futuros

Como parte final de la memoria, se muestra las conclusiones del proyecto así como posibles proyectos futuros.

Índice

Capítulo 8 Conclusiones y trabajos futuros	51
8.1. Conclusiones	51
8.2. Trabajos Futuros	52
5	

8.1. Conclusiones

En este proyecto de fin de carrera se ha implementado una aplicación, la cual es capaz de capturar, monitorizar y visualizar los consumos en las distintas partes de un CPD y realizar el cálculo y la visualización del PUE de un CPD, así como la captura, monitorización y visualización de información sobre el estado de cada componente del CPD. Gracias a ello se ha logrado una aplicación altamente útil para la monitorización de la eficiencia energética de un centro de proceso de datos que permitirá ahorrar en el consumo energético en la empresa donde se situé esta aplicación, tal y cómo era el objetivo principal de este proyecto.

La aplicación mide mediante sondas de medida los consumos de las distintas partes de un CPD. Conociendo así las perdidas y consumos energéticos en cada parte de un CPD, que nos servirán para conocer el PUE en dos de sus categorías, la primera de ellas haciendo la diferencia entre la potencia eléctrica del CPD entre la potencia eléctrica de la salida de la SAI y la segunda haciendo la diferencia entre la potencia eléctrica del CPD entre la potencia eléctrica del CPD entre la potencia eléctrica del cPD entre la potencia eléctrica de las cargas IT. A su vez la aplicación muestra información del estado de una serie de componentes del CPD (agentes SNMP) mediante el uso de un driver que implementa este protocolo.

Una vez monitorizado dicha información, esta información será mostrada a través de una herramienta web, pudiendo ser observada de dos formas. La primera mediante una visualización de los valores capturados y la segunda mediante el uso de sinópticos SVG que permiten una visualización más amigable y atractiva para el usuario.

Estos sinópticos nos permitirán mostrar modelos 2D de la disposición física de los componentes en el CPD junto a la información aportada por ellos a la herramienta. Con el uso de estos sinópticos se hace posible realizar una herramienta que muestra alertas de manera visual, indicándolo mediante texto y código de colores para cada estado de los componentes. Con lo que si algún componente falla se muestra una señal roja con el texto "critical" y así con cada estado posible del componente. Haciendo posible de esta manera la sustitución de alguno de los componentes del CPD por mal funcionamiento o fallo del mismo, y consiguiendo de esta manera una mejora en la eficiencia energética de un CPD.

Durante la realización del proyecto fin de carrera he comprobado la importancia que es tener una buena planificación en el proyecto, dedicar el tiempo necesario para el análisis de los requisitos, así como el diseño. Para, de esta manera, poder desempeñar la actividad de implementación lo más rápidamente posible y de una manera fiable, que más tarde supere las pruebas y cumpla con los requisitos esperados.

Gracias a la metodología usada, se ha ido en cada iteración añadiendo más funcionalidades al sistema, de manera que como resultado final de las iteraciones contábamos con un prototipo con en el que analizar el buen rumbo de la aplicación. Y por último tener paciencia y dotes de superación frente a adversidades y fallos en el desarrollo del proyecto.

8.2. Trabajos Futuros

En esta sección se muestran los posibles trabajos futuros que se podrían realizar una vez implantada la aplicación.

En el futuro más próximo el trabajo sería aumentar el alcance de la aplicación, de tal manera que se ampliara el tipo de agentes SNMP que se monitorizan, de está manera podríamos tener toda una red monitorizada. Esta monitorización nos podría permitir conocer el estado de todos los elementos conectados a red y controlar por lo tanto todo un edificio. Incluso podríamos añadir una serie de controladores a elementos como climatizadores, ventanas, etc. Y conectarlos con nuevos drivers para poder monitorizar y controlar por ejemplo, un caso en el que la temperatura de una oficina sea alta y decidir activar el climatizador de esa oficina.

También se podría comprar una centralita Asterisk para configurar el envió de mensajes de voz para comunicar una situación peligrosa en el CPD como puede ser el aumento de temperatura en todo el CPD.

A medio plazo el trabajo que se puede hacer sobre el proyecto es la posibilidad de tener una BBDD en la que se tenga información de configuración de los agentes SNMP y las ODIs que se quieren consultar y generar de manera automática los archivos de configuración XML, así solo teniendo la BBDD al día facilitaríamos la configuración.

Y a largo plazo se ha pensado en cambiar la forma de trabajar con los archivos de configuración, hasta ahora se añadía en los archivos de configuración los agentes SNMP a consultar y las OIDs a consultar. Esto se cambiaria por una interfaz grafica para que el usuario en tiempo real pudiese elegir que agentes SNMP consultar y que OIDs a consultar. De tal manera que el conocimiento sobre OIDs fuera transparente para que el usuario solo tuviese que pensar por ejemplo "Quiero preguntar la temperatura del servidor", y que se hiciese la configuración para obtener ese valor de manera automática.

Bibliografía

- [May10] Joe Mayo. *Microsoft Visual Studio 2010 : A Beginner's Guide*. McGraw-Hill.1° Edición. 2010.
- [Cag02] Kurt Cagle. *SVG Programming: The Graphical Web.* Apress. 1° Edicion. 2002.
- [Eis11] J. Eisenberg. SVG Essentials. O'Reilly Media. 1º Edicion. 2011
- [GGrid] Web official del consorcio "Green Grid", el cual se encarga de declarar la medición de eficiencia energética(PUE) y otras series de métricas en un CPD. Abril 2012.
- [Gree12] The Green Grid advances global harmonization of datacenter energy efficiency metrics.(BUSINESS): An article from: Green Data Centers & Internet Business. Abril 2012.
- [Rec12] Recommendations for Measuring and Reporting Overall Data Center Efficiency. Abril 2012. http://www.energystar.gov/ia/partners/prod_development/downloads/Dat a_Center_Metrics_Task_Force_Recommendations_V2.pdf
- [Arreg03] Mauricio Arregoces, Maurizio Portolani. *Data Center Fundamentals*. Cisco Press. 1º Edición. 2003
- [Smit11]] Hubbert Smith. Data Center Storage: Cost-Effective Strategies, Implementation, and Management. Auerbach Publications. 1° Edición. 2011.
- [Will00] James William, W. Nilsson, Susan A. *Riedel. Electric Circuits, Revised Printing.* Prentice Hall. 6º Edición. 2000
- [Mau05] Douglas Mauro, Kevin Schmdit. *Essential SNMP*. O'Reilly Media, 2° Edicion. 2005.
- [Stal99] William Stallings. *SNMP*,*SNMPv2*, *SNMPv3* and *RMON 1* and 2. Addison-Wresley Professional. 3° Edicion. 1999.
- [Chap12] Laura Chappell, Gerald Combs. *Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide*. Laura Chappell University. 2° Edicion. 2012.
- [Rot11] Chris Roth. Using Microsoft Visio 2010. Que. 1º Edición. 2011.
- [Stel10] Andrew Stellman, Jennifer Greene. *Head First C#, 2E: A Learner's Guide to Real-World Programming with Visual C# and* .NET. O'Reilly Media. 2° Edicion. 2010.

[Trol10]	Andrew Troelsen. <i>Pro C# 2010 and the .NET 4 Platform, Fifth Edition.</i> Apress. 5° Edicion. 2010.
[AdoCS11]	Adobe. Adobe CS6 Design Standard Student and Teacher Edition. 2011.
[Gasp11]	Joao Gaspar. Google SketchUp Pro 8 step by step. VectorPro. 1° Edicion. 2011.
[PS212]	Sitio web de la herramienta PDF2svg donde se encuentra el manual de la herramienta. http://www.pdftron.com/pdf2svg/documentation.html
[TP500]	Web TOP500, esta web nos ofrece una lista de los 500 supercomputadores más potentes del mundo, se rige por flop/s para organizar la lista. Junio 2012.
[SSN12]	Sitio web de la Librería SNMPSharpNET, donde se puede descargar la propia librería y se nos aporta su API y ejemplos de uso de ella. http://www.snmpsharpnet.com
[iR12]	Sitio web de la herramienta MIB Browser iReasoning. http://ireasoning.com/mibbrowser.shtml
[MG12]	Sitio web de la herramienta MG-SOFT MIB Browser. http://www.mg-soft.si/mgMibBrowserPE.html
[IPMU12]	Sitio web de la herramienta IPMIUtil. http://ipmiutil.sourceforge.net/
[IPMT12]	Sitio web de la herramienta IPMITool. http://ipmitool.sourceforge.net/
[IDbox]	Sitio web de la herramienta IDbox idbox.cic.es. Septiembre 2012
[MET12]	Javier Garzás, Emanuel Irrazábal, Juan Enríquez, Mario Piattini. <i>Gestión Ágil de Proyectos Software</i> . Kybele Consulting, 1º Edición. 2012.
[DELL12]	Sitio web donde consultar la posibilidad de aportar información de consumo energético de los modelos de servidores. Mayo 2012.http://support.dell.com/support/edocs/SOFTWARE/smdmc/202/sp/ ug/powermon.htm
[DELLS12]	Sitio web donde consultar las características y manuales de cada servidor de la marca DELL. Mayo 2012. http://www.dell.com/support/troubleshooting/us/en/555/ProductSelector/ Select/FamilySelection?CategoryPath=all- products%2Fesuprt_ser_stor_net%2Fesuprt_poweredge&Family=Power Edge&DisplayCrumbs=Product%2BType%40%2CServers%252c%2BSt orage%2B%2526%2BNetworking%40%2CPowerEdge&rquery=na
[CIRC12]	Sitio web donde consultar las características de toroides de la marca circuitor. Mayo 2012.

	http://circutor.es/m-medida/m7-transformadores-de-medida-y- shunts/transformadores-de-corriente-alta-precision-tch_p_144.aspx
[BECK12]	Sitio web donde consultar las características de las tarjetas de adquisición beckhoff. Mayo 2012. https://logicelectronic.com/BECKHOFF/CPU%20CX9010.htm
[OPC12]	Sitio web donde consultar documentación sobre el protocolo OPC. Mayo 2012. http://es.wikipedia.org/wiki/OPC
[PW20]	Sitio web del rack de marca DELL y modelo Poweredge 4220. Mayo 2012. http://www.dell.com/us/enterprise/p/poweredge-4220/pd
[PW10]	PDF indicando sus especificaciones técnicas del rack de marca DELL y modelo Poweredge 4210. Mayo 2012. http://www.verinet.dk/dell_rack_system.pdf
[EAT12]	Sitio web de la SAI de marca Eaton y modelo 9355/30N7 http://powerquality.eaton.com/EMEA/Products-services/Backup-Power- UPS/9355-UPS/9355-specs.asp

Anexo 1. Creación de servicio Windows

Dentro de este Anexo se explica el proceso de creación de un servicio Windows, este proceso es necesario para convertir el driver SNMP en un servicio Windows y poder ejecutarlo de manera continua desde una maquina Windows y consulte repetidamente la información requerida a los agentes SNMP.

Previamente antes de pasar a la propia creación del servicio Windows tenemos que tener realizado en Visual Studio nuestro proyecto que queremos que se ejecute como un servicio Windows. Una vez tenemos este proyecto pasamos a la propia creación del servicio Windows. A continuación se muestran los pasos para la creación de convertir un proyecto en un servicio Windows.

1. Crearemos un proyecto en C# de tipo Servicio de Windows, en nuestro caso se creara con el Framework 3.5 debido a exigencia de la empresa. Y le otorgamos un nombre el cual será SIDP.Service.SNMP.



Figura 21: Anexo 1. Paso 1 servicio Windows

Así quedará la estructura del proyecto creado.



Figura 22: Anexo 1. Figura estructura servicio Windows

La estructura del proyecto quedaría como se muestra en la Figura Anexo 1. Estructura servicio Windows

Ahora, una vez creado el proyecto, se modificará la clase *Program.cs* para insertar en ella el código que necesitemos para ejecutar el driver creado en *SIDP.Core.SNM*.

Este código se encarga de crear un clase *SNMPMultiDrivers* para ser ejecutada por la clase Service1.cs que creará a su vez un objeto de la clase *Program.cs* y ejecutará el método *setup()* perteneciente al objeto, anteriormente creado, de la clase *SNMPMultiDrivers* y los métodos que nos permiten activar el servicio (*init(), start()*) y para el servicio (*stop(), exit()*). Este código nos permite correr un servicio que lea la configuración de los archivos de configuración y comience la adquisición de datos del CPD.

2. Ahora para crear el instalador, agregaremos al proyecto un nuevo ítem, clic derecho sobre el proyecto SIDP. Service.SNMP→Agregar→Nuevo elemento... Y seleccionamos el objeto Clase Instalador, como aparece en la imagen.



Figura 23: Anexo 1. Paso 2 servicio Windows

Ahora debemos agrega los elementos ServiceInstaller y ServiceProcessInstaller. Para ello haremos que se muestre la barra de herramientas y haremos clic derecho sobre la barra de herramientas y escogemos elegir elementos.

3. Ahora una vez creado la clase Instalador, tendremos que agregar al proyecto los elementos ServiceInstaller y ServiceProcessInstaller. Para agregar estos elementos hacemos click derecho sobre el cuadro de herramientas y seleccionamos la opción "Añadir elemento" y escogemos los elementos mencionados. Como se muestra en la figura a continuación.

SIDP.Service.SNMP - Microsoft Visual Studio (Administr	ador)	the second se				- 0 X
Archivo Editar Ver Proyecto Generar Depurar E	quipo	atos Herramientas Arguitectura Prueba Analizar	Ventana Ayuda			
8 - 9 - 0 - 0 X & A 0 - 0 - A	- 2	Debug v 🕅 responde	- 🔄 🕾 🗟 🕸 😒 🖬 🛍 🖬 - 🗋			
「新臣を引きる軍官法務後」。		응화화학 만만 역망 별 광.				
🟭 Cuadro de herramientas 🔍 📼 🗙		vice1.cs [Diseño] Examinador de objetos		- E	xplorador de soluciones	- ₽ ×
Componentes SIDP.Service.SNMP	C			2		
Todos los formularios Windows Forms		gir elementos del cuadro de herramientas			System Configuration Install	
Secontroles comunes		Componenter Silverlight Componenter de Su	rtem Workflow Componenter de Syste	m Activities	System Core	
Contenedores		Componentes de .NET Framework	Componenter COM Component	ater W/DE	- System.Data	
Menús y barras de herramientas			ites wer	- System.Data.DataSetExtensions		
a Datos		Nombre Espacio de nombres	Nombre del ensambl Directorio	*	- System.Management	=
Componentes		Separator Microsoft Office Tools Ribb	han Microsoft Office Tool Caché glob		- System.ServiceProcess	
>> Impresión		Seguence System Workflow Activities	System Workflow Acti Caché glob		- System.Xml	
P ▷ Cuadros de diálogo		SerialPort System IO Ports	System Caché glob		- System.Xml.Ling	
8. ▷ Interoperabilidad WPF		Server Microsoft Analysis Services	Microsoft AnalysisSer Caché glob		Emlerador de coluci	🔛 Virta da clarar
▷ Informe		Service1 SIDP.Service.SNMP	SIDP.Service.SNMP (1 c:\visual stu		realitexploted de soldel	vista de clases
Visual Basic PowerPacks		ServiceBase System.ServiceProcess	System.ServiceProcess Caché glob		piedades	- 4 ×
a General		ServiceController System.ServiceProcess	System.ServiceProcess Caché glob		italler1 System.Configuration.Install.Installer	r -
	arrá	ServiceInstaller System.ServiceProcess	System.ServiceProcess Caché glob			
un elemento a este texto y agréguelo al cuadro de	ear	ServiceProcessInstaller System.ServiceProcess	System.ServiceProcess Caché glob			
herramientas.		SetState Microsoft.SharePoint.Work	fl Microsoft.SharePoint Caché glob		(Name) Installer1	
		SetState System.Workflow.Activities	s System.Workflow.Acti Caché glob	_	(Name) Installer I	do)
		The second secon			Localizable Falce	10)
		<u>Filtro</u> :		Borrar	Varios	
		ServiceInstaller			HelpText	
		Idioma: Todos los idiomas (todos los p	aíses)	Examinar	Parent (ninguno)	
					(ingato)	
			Aceptar Cancelar	Restablecer		
	U					
	1.17				(Name)	
					Indica el nombre utilizado en el código para ide	entificar el obieto
					and a normal administration of the courgo para fue	interior er objeto.
	in the second					
🥳 Lista de errores						
Listo						

Figura 24: Anexo 1. Paso 3 servicio Windows

Quedando la estructura de Service1.cs como se muestra en la "*Figura Anexo 1*. *Estructura Service1*".

1	Installer1.cs [Diseño]*	х	Service1.cs	Program.cs	Service1.cs [Diseño]	Examinador de objetos 🔹
Explorador de s	🏶 serviceInstallerI	L	servicel 🏶	ProcessInstaller1		
				E :		1 Estave store Construction 1

Figura 25: Anexo 1. Estructura Service1

Una vez añadidos los elementos *serviceInstaller1* y *serviceProcessInstaller1* a Service1.cs, pasaremos a configurar su funcionamiento.

Para el elemento *serviceProcessInstaller1* (el cual realiza trabajo común a todos los servicios de un ejecutable, La utilidad de instalación lo utiliza para escribir valores del registro asociados a los servicios que desea instalar), en sus propiedades configuramos, Account (LocalSystem, obtiene o establece el tipo de cuenta en la que se ejecutará esta el servicio). Parent (Installer1, instalador que contiene la colección a la que pertenece este instalador).

Propiedades	- ∓ ∓ X
serviceProcessInstalle	r1 System.ServiceProcess.ServiceProces 🗸
🔡 24 🔳 🖋 🖻	
▲ Diseño	
(Name)	serviceProcessInstaller1
GenerateMember	True
Modifiers	Private
 Varios 	
Account	LocalSystem 💌
HelpText	
Parent	Installer1

Figura 26: Anexo 1. Configuración serviceProcessInstaller

Para el elemento *serviceInstaller1* configuraremos sus propiedades al gusto que queramos. Las propiedades son Description, DisplayName, ServiceName y StartType. La descripción se mostrara posteriormente como información asociada al servicio Windows y se mostrara cuando se visualicen los servicios en ejecución El DisplayName es el nombre que le queremos dar a nuestro servicio. EL ServiceName es el nombre que se mostrara cuando se visualice el servicio. Y el StartType es la forma en que se ejecutara el servicio cuando encendamos el equipo Windows.

Pro	piedades	▼ ₽	×
sei	rviceInstaller1 System.Servio	ceProcess.ServiceInstaller	•
	<u>2</u> ↓ 🔲 🖋 🖾		
	(Name)	serviceInstaller1	•
	GenerateMember	True	
	Modifiers	Private	
4	Varios		
	Description		
	DisplayName		=
	HelpText		
	Parent	Installer1	
	ServiceName		
⊳	ServicesDependedOn	Matriz String[]	
	StartType	Automatic	-

Figura 27: Anexo 1. Configuración serviceInstaller

4. Se generará un proyecto de tipo Instalador. Sobre el nombre de la solución *"SIDP.Services.SNMP"*, hacemos clic derecho y seleccionamos Agregar→Nuevo proyecto.



Figura 28: Anexo 1. Paso 4 servicio Windows

Quedando la estructura de Setup1.cs como se muestra en la Figura 28.



Figura 29: Anexo 1. Estructura Setup

5. Hacemos clic derecho sobre Carpeta de la aplicación y Seleccionamos Agregar→Resultado del proyecto.



Figura 30: Anexo 1. Paso 5 servicio Windows

Quedando la estructura final de Setup1.cs como se muestra en la Figura 30.



Figura 31: Anexo 1. Estructura final setup

6. Pasaré a configurar las propiedades de *setup1* que indicará información nrelevante sobre el creador del servicio Windows.

Propiedades	→ □ ×
WindowsServiceSetup.SNMP Propied	ades del proyecto de implementación 🔹
2 2	
Varios	
AddRemoveProgramsIcon	(Ninguno)
Author	CiC
BackwardCompatibleIDGeneration	False
Description	
DetectNewerInstalledVersion	True
InstallAllUsers	False
Keywords	
Localization	Spanish (Spain) (International Sort)
Manufacturer	CiC
ManufacturerUrl	
PostBuildEvent	
PreBuildEvent	
ProductCode	{5DC2A6C6-B93F-45FD-9805-9F8C7A52511E}
ProductName	WindowsServiceSetup.SNMP
RemovePreviousVersions	False
RunPostBuildEvent	Si la compilación es correcta
SearchPath	
Subject	
SupportPhone	
SupportUrl	
TargetPlatform	x86
Title	WindowsServiceSetup.SNMP
UpgradeCode	{1B2DCC60-FFAC-45C0-BDB7-61273E6840E3}
Version	1.0.0
Varios	

Figura 32: Anexo 1. Propiedades setup

7. Una vez terminamos nuestro instalador es importante generar el proyecto y posteriormente instalar el servicio. Para ello seleccionaremos con click derecho sobre setup1 la opción *"Install"*

lution Explorer			
1 5 6 7 1			
Solution 'Window	isService1' (2 projects) ervice1		
=- ↓2) Baun - ⊕- ┣= De 🎬 ↓- 🔂 Pri	Build Rebuild		
	View		•
	Add		•
Γ	Install		
G	Uninstall Open Folder in Windows Explorer		
2	Add Solution to Source Control		
×	Cut	Ctrl+X	
×	Remove Rename	Del	
6	Properties	Alt+Enter	r

Figura 33: Anexo 1. Instalar servicio

8. Una vez se ejecute el instalar seleccionamos Next haciendo click. Y configuraremos donde queremos instalar el servicio.

🖥 Setup 1			<u>_ _ ×</u>
Welcome to the Setup1	Setup Wiza	ırd	5
The installer will guide you through the	steps required to inst	all Setup1 on your co	mputer.
WARNING: This computer program is p Unauthorized duplication or distribution or criminal penalties, and will be prosec	protected by copyrigh of this program, or ar uted to the maximum	t law and internation y portion of it, may re extent possible unde	al treaties. esult in severe civil r the law.

Figura 34: Anexo 1. Proceso Instalación

9. Para verificar la instalación. Vamos a Inicio→Ejecutar y escribimos "services.msc". Y buscaremos el servicio instalado entre todos.

Anexo 2. Uso del MIB Browser

La herramienta MG-SOFT MIB Browser nos permite analizar la MIB de un agente SNMP de manera remota. La herramienta ha sido utilizada para analizar los agentes SNMP presentes en el proyecto y así conocer que datos que se pueden consultar en un agente SNMP.

La herramienta nos permite una gran cantidad de funciones con los agentes SNMP y sobre ellos. En este pequeño tutorial solo se van a explicar las funciones utilizadas en el proyecto para conocer los datos que nos devuelve cada agente SNMP.

Las funciones que soporta el MIB Browser:

- Contactar remotamente con un agente SNMP.

- Usando IPv4 o IPv6.

- Usando IPX.

- Seleccionar la interfaz de enlace.

- Especificar los parámetros del protocolo SNMP.

- Usando SNMPv1.

- Usando SNMPv2.

- Usando SNMPv3, creando una password para la conexión.

-Configurar los agentes SNMP y su uso.

- Usando SNMPv1

- Usando SNMPv2

- Usando SNMPv3

- Visualizar estado de agentes SNMP.

- Guardar perfil de los agentes.

- Obtención de la MIB y propiedades de cada nodo.

- Guardar y cargar MIBs.

- Operaciones soportadas.

-Get Request.

-GetNext Request.

-Walk.

-Set.

-Recivir Traps y su configuración.

-Descubrir agentes SNMP remotos.

-Monitorizar agentes SNMP.

-Comparar instantáneos de agentes SNMP.

-Guardar y cargar instantáneos de agentes SNMP.

-Representación grafica de valores instantáneos.

-Guardar y cargar representaciones graficas de valores instantáneos..

-Simular agentes SNMP.

Las funciones que utilizaré para comprobar los agentes SNMP son las siguientes:

Especificar los parámetros del protocolo SNMP

Usando SNMPv1

Contactar remotamente con un agente SNMP.

Usando IPv4

Configurar los agentes SNMP y su uso.

Usando SNMPv1

Obtención de la MIB.
Operaciones soportadas.

Walk

1. Empezando con MG-SOFT MIB Browser

Esta sección presenta los requisitos básicos del computador para instalar y usar MG-SOFT MIB Browser. El MG-SOFT MIB Browser se trata de una herramienta multiplataforma en la que se ha hecho uso de la plataforma Microsoft Windows para utilizarla. Con lo que el tutorial solo indica el uso en esta plataforma.

1.1. Configuración de requisitos

Para que la herramienta pueda utilizarse en Microsoft Windows el sistema operativo debe de ser uno de los siguientes: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, o Windows 7. También se requiere Microsoft internet explorer 4.0 o superior para acceder a ciertas funciones del software.

2 Instalación de MIB Browser

Una vez comprobado que el computador donde se desea instalar la herramienta cumple los requisitos de configuración se instalará la herramienta a través del instalador que nos podemos descargar de la web de la herramienta. Y se seguirán las indicaciones mostradas por pantalla para llevar a cabo la instalación.

3. Comienzo con MIB Browser

Seleccionaremos Inicio → Programas → MG-SOFT MIB Browser → MIB Browser command from the Windows taskbar. Cuando el programa arranque, se mostrara una ventana con información sobre la herramienta. Haremos click sobre Continue para continuar con la herramienta.

3.1 MIB Browser

El diseño de la apariencia del MIB Browser sigue las convenciones sobre las aplicaciones Windows. El MIB Browser tiene una barra de titulo, un menú, una barra de herramientas, una barra de estado y los botones de minimizar, maximizar y cerrar la herramienta.



Figura 35: Anexo 2. MIB Browser

En el escritorio MIB Browser, se puede elegir entre tres pestañas generales, la pestaña de consulta, la pestaña mib y la pestaña ping. Las opciones que nos muestran estas partes constan de paneles, cuadros o pestañas.

La pestaña de consulta es la que se utilizará y consta de tres partes. Las cuales son el acceso remoto al agente seleccionado, la estructura del árbol MIB y los resultados de las operaciones realizadas.

3.1.1 Partes principales

Barra de menú

La barra se sitúa en la parte superior del MIB Browser que contiene menús, tales como Archivo, Editar, Ver, SNMP, etc

Barra de herramientas

La barra de herramientas en la parte superior del MIB Browser, contiene botones que abren la mayor parte de función la mayor parte de MIB Browser o proporciona acceso rápido a algunos comandos de la MIB.

🚳 MG-SOFT MIB Browser Professional Edition	
File Edit View SNMP Action Tools Window Help	
ei ?() Ø 🋠 28 (8) 🗴 🗏 🛄 🙀 🖢 🧯 🚱 🗃 🧐 🚰 🗐	0 🖒

Figura 36: Anexo 2. Barra de herramientas

Área de trabajo
La zona se sitúa entre la barra de herramientas y la barra de estado (zona central de la herramienta), en el que se puede elegir entre la pestaña Query, MIB y Ping. Los paneles ubicados en esta pestaña le permiten cargar módulos MIB, para realizar las operaciones básicas de SNMP, y mostrar los resultados de realizar las operaciones. Por otra parte, permiten hacer un ping a un agente SNMP remoto.

Barra de estado

Es una barra situada en la parte inferior del MIB Browser con cinco campos que muestran los diferentes tipos de información.



Figura 37: Anexo 2. Barra estado

El primer campo que informa sobre el último cambio o acción que se realiza en el browser del MIB (por ejemplo, se le informa si una consulta de un agente SNMP ha tenido éxito o ha agotado el tiempo, o cual nodo MIB ha sido seleccionado, etc.). El segundo campo muestra el número de paquetes SNMP enviados desde un agente, del cual se pone a cero cada vez que el panel de resultados de la consulta es aclarado. El tercer campo muestra la versión de SNMP utilizada actualmente, el cuarto campo son una serie de luces que indica se la conexión ha fallado, se ha reintentado o ha sido correcta. Y el último campo muestra un reloj despertador que con una animación le notifica acerca del recibimiento de Traps.

4. Especificar valores del protocolo

En esta sección, se explicará como se especifica el protocolo SNMP a utilizar en las consultas a los agentes SNMP. En este caso se explicará el protocolo SNMPv1 porque es el utilizado en la totalidad del proyecto.

Para especificar el protocol, seleccionaremos el botón SNMP Protocol Preferences.



Figura 38: Anexo 2. Botón SNMP Protocol Preferences

2. Se abrirá la ventana SNMP Protocol.

SNMP Protocol Preference	ces					
SNMP protocol version						
SNMPv1 ○	SNMPv2c	NMPv2c 💿 SNMPv3				
General	Get-Bulk	Get-Bulk settings				
Read community	🗸 Use G	✓ Use Get-Bulk				
public 🗸	0	Non repeaters				
Set community	10	Max repetitions				
private 👻	SNMPv3 security					
Timeout [s] 5	User sec	User security name				
Retransmits 4] Securitul	Coouritu laural				
Port number 161 👻						
Add to agent profiles	Load user	profile Edit user OK Cancel				

Figura 39: Anexo 2. SNMP Protocol Preferences

3. Haremos click en la opción SNMPv1.

4. En *"Read community"*, se especifica la comunidad de lectura del agente. Este parámetro solo se usara en las consultas Get, GetNext, GetBulk y Walk.

5. En *"Set community"* se especifica la comunidad de escritura del agente. Este parametro solo se usará en las consultas Set.

6. En *"Timeout"*, se especificara el valor en segundos que se espera a la respuesta a una consulta a un agente SNMP.

6. En "Retransmits", se especifica el numero de reintentos sobre la conexión con un agente SNMP.

7. En "Port", se especifica el número de puerto para comunicarse con el agente. El puerto por defecto es 161.

8. Haremos click en el botón OK para cerrar la ventana y aplicar los cambios realizados.

5. Contactar remotamente con un agente SNMP

Para monitorizar o gestionar un agente SNMP previamente se tiene que establecer un contacto con el agente. Después de contactar con el agente, se pueden recibir valores de las consultas sobre él.

5.1 Contacto Remoto con un agente SNMP

1. En la pantalla principal, escogeremos la pestaña Query.

2. En el input "*Remote SNMP Agent*", pondremos la dirección ip del agente al que queremos consultar.

3. Haremos click sobre el botón "Contact Remote SNMP Agent".



Figura 40: Anexo 2. Botón Contact Remote SNMP Agente

4. El MIB Browser contactara con el agente SNMP seleccionado y devolverá los datos en el panel "Query results".



Figura 41: Anexo 2. Query results

5.1 Usando Direcciones IPv4 para contactar con el agente SNMP

Para contactar y consultar a un agente SNMP utilizaremos la opción IPv4 para la configuración de la dirección IP del agente al cual consultar, la cual es la más conocida entre los usuarios de red.

1. Primero, tenemos que seleccionar el puerto para la comunicación. Para ello, abre el "SNMP Protocol Preferences" (View/ SNMP Protocol Preferences command).

2. En la nueva ventana, selecciona el número de puerto al que consultar en el agente SNMP.

5.2 Seleccionar nodos en la MIB

En el MIB Browser, se pueden cargar módulos MIB, organizar y representar la MIB de un agente SNMP en forma de árbol. Se puede ver la estructura en la siguiente figura.

MG-SOFT MIB Browser Professional Edition				-
File Edit View SNMP Action Tools Window Help				
🛃 ?(] Ø 🎌 🕸 🕸 🗕 🧯 🗰 🗓 🔯 🏷 🦉	3 😭	4	0 🖉	3
Query MIB Ping				
Remote SNMP agent Split				
192.168.0.15 👻 🔀 🔲 Vertical				
MIB tree				
🔺 🔯 MIB Tree				*
a 🛅 internet				
irectory				
🔺 🧰 system				
🎐 sysDescr				
sysLopEctID sysLopTime sysContact sysName				11
interfaces				
b in at				
> 🛅 top				
b 💼 udp				
▷ constant			L	
⊳ Construction				
🚞 experimental				
private				÷
Query results				
3420: enterprises.77.1.3.2.0 (counter) 0				
3421: enternrises 77.1.3.3.0 (counter) 13311 <			•	Ŧ
				_
OID 1.3.6.1.2.1.1.1	3426		🖻 SNMPv1 🛛 🔍 🔍 💆	ž

Figura 42: Anexo 2. MIB

Esta MIB se obtiene haciendo una consulta Wak, la cual nos permite recorrer esta MIB en forma de árbol y devolvernos los valores de sus OIDs.

5.3 Operación Walk

Se puede obtener todos los valores de la MIB del agente SNMP mediante la operación Walk.

Cuando el MIB Browser empieza con la operación Walk, primero envía una consulta GetNExt al agente con la OID ".1". En respuesta, le llega la OID y el valor de esta primera OID. Para continuar con la consulta de todos los valores que me puede devolver el agente enviare otra consulta GetNext a la OID siguiente y así sucesivamente recorrer en forma de árbol. Hasta llegar a una OID que no especifica ningún valor. 5.3.1 Proceso operación Walk

1. Seleccionar en el menú principal la opción Walk. Se puede hacer mediante el botón *"Walk Remote SNMP Agent"*.

2. Se devolverán todos las OID con valores en la "Query Results.



Figura 43: Anexo 2. Query results

Los resultados obtenidos serán las OIDs con el valor asociado a ella, este valor puede ser un texto, un valor numérico, una ip, una posición de memoria, etc.

Anexo 3. WireShark

Para comprobar el uso correcto del prototipo del driver SNMP creado, se tendrá que utilizar un analizador de paquetes de red. De esta manera comprobaremos si las consultas al agente SNMP nos devuelve información de manera correcta y fiable. Para ello comprobaremos a través de WireShark si se cumple la configuración del agente SNMP desde el driver SNMP y la respuesta a la consulta de las OIDs requeridas.

Para ello se ha utilizado la herramienta WireShark. La cual es un capturador/analizador de paquetes de red, que permite ver a un nivel bajo y detallado qué está pasando en tu red. La ventaja de Wireshark es que se trata de un herramienta opensource y multiplataforma.

El driver SNMP hace una consulta sobre un agente SNMP especificando la dirección ip del mismo, el puerto al cual conectarse, la versión del protocolo SNMP, la comunidad y el timeout y retrys deseados para la consulta. Esta consulta se realiza a través de paquetes UDP.

La configuración del agente en el driver SNMP es la siguiente: IP: 10.60.2.4 Puerto: 161 Comunidad: cicsnmp Protocolo: SNMPv1 Retry: 6 Timeout: 2000 ms Protocolo para cración de paquetes: UDP

En este caso, para analizar si la consulta se ha llevado a cabo correctamente, solo necesitaremos consultar si la información configurada para la conexión con el agente en el driver SNMP es la correcta, se trata de si el portocolo SNMP es detectado como en uso por el WireShark, la ip, puerto y comunidad del agente SNMP y si los paquetes enviados son pertenecientes al protocolo UDP.

Una vez realizada la consulta sobre el agente SNMP, se ha detectado por parte de la herrameinta WireShark dos paquetes, un paquete que corresponde con la consulta SNMP y otro que corresponde con la respuesta a la consulta. Como se observar en la siguiente figura.

Capturing from Realtek PCIe GBE Family Contro	oller [Wireshark 1.6.7 (SVN Rev 41973 from /trunk-	1.6)]
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> apture <u>A</u> nalyze <u>S</u> tat	tistics Telephony <u>T</u> ools <u>I</u> nternals <u>H</u> elp	
	🔍 🗢 🔿 春 🕹 🔳 🔳 🖯 🔍	2, 0, 🖭 🎬 🗹 畅 % 💢
Filter: ip.addr==10.60.2.4	 Expression Clear 	Apply
No. Time Source	Destination Protoco	l Length Info
295 7.366276 10.60.8.8 296 7.374827 10.60.2.4	10.60.2.4 SNMP 10.60.8.8 SNMP	256 get-request 1.3.6.1.2.1 273 get-response 1.3.6.1.2.
< [III	•
Realtek PCIe GBE Family Controller: <live ca<="" td=""><td>Packets: 100244 Displayed: 2 Marked: 0</td><td>Profile: Default</td></live>	Packets: 100244 Displayed: 2 Marked: 0	Profile: Default

Figura 44: Anexo 3. WireShark

Ahora se pasará a analizar por separado la información que nos proporciona cada paquete y comprobar de esta manera si se ha establecido correctamente la conexión y consulta SNMP.

En el primer paquete, el cual es el request, donde se obtiene la información de conexión con el agente SNMP. Esta información nos aporta si se cumple que el protocolo de internet para los paquetes es UDP, el protocolo de conexión es SNMP y la ip y puerto son los configurados. En la *Figura 44*.

□ Internet Protocol Version 4, Src: 10.60.8.8 (10.60.8.8), Dst: 10.60.2.4 (10.60.2.4)
Version: 4
Header length: 20 bytes
⊟ Differentiated Services Field: 0x00 (D5CP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
0000 00 = Differentiated Services Codepoint: Default (0x00)
00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
Total Length: 242
Identification: 0x2295 (8853)
Fragment offset: 0
Time to live: 128
Protocol: UDP (17)
⊟ Header checksum: 0x0000 [incorrect, should be 0xf8e2 (maybe caused by "IP checksum offload"?)]
[Good: False]
🖃 [Bad: True]
Elexpert Info (Error/Checksum): Bad checksum]
[Message: Bad checksum]
[Severity level: Error]
[Group: Checksum]
Source: 10.60.8.8 (10.60.8.8)
Destination: 10.60.2.4 (10.60.2.4)

Figura 45: Anexo 3. Paquete request

En la *Figura 45* podemos ver el puerto de conexión utilizado, el cual es el 161 y coincide con el configurado en el driver.

```
    □ User Datagram Protocol, Src Port: 64690 (64690), Dst Port: snmp (161)
    Source port: 64690 (64690)
    Destination port: snmp (161)
    Length: 222
    ⊕ Checksum: 0x1f73 [validation disabled]
    Figura 46: Anexo 3. Paquete request puerto
```

En el segundo paquete, el cual es el response, donde se obtiene la información de los valores de las OIDs consultadas y donde vemos que ahora la dirección de origen es el agente SNMP, que envía la respuesta a la estación que le consulta.

En la Figura 46 vemos el resultado de las consultas a las OIDs.

```
Simple Network Management Protocol
    version: version-1 (0)
    community: cicsnmp
 data: get-response (2)
    get-response
       request-id: 326511154
       error-status: noError (0)
       error-index: 0
     wariable-bindings: 10 items
       □ 1.3.6.1.2.1.33.1.4.4.1.2.1:
            Object Name: 1.3.6.1.2.1.33.1.4.4.1.2.1 (iso.3.6.1.2.1.33.1.4.4.1.2.1)
            Value (Integer 32): 230
       □ 1.3.6.1.2.1.33.1.4.4.1.2.2:
           Object Name: 1.3.6.1.2.1.33.1.4.4.1.2.2 (iso.3.6.1.2.1.33.1.4.4.1.2.2)
            Value (Integer32): 230
       □ 1.3.6.1.2.1.33.1.4.4.1.2.3:
           Object Name: 1.3.6.1.2.1.33.1.4.4.1.2.3 (iso.3.6.1.2.1.33.1.4.4.1.2.3)
            Value (Integer32): 230
       □ 1.3.6.1.2.1.33.1.4.4.1.3.1:
           object Name: 1.3.6.1.2.1.33.1.4.4.1.3.1 (iso.3.6.1.2.1.33.1.4.4.1.3.1)
           Value (Integer32): 18
                                  Figura 47: Anexo 3. Paquete response
```

Ahora cabe preguntarnos ¿Cómo saber si la información aportada por las OIDs es correcta?, esto se responde indicando que el proveedor del agente SNMP, en este caso una SAI, nos proporciona una web donde se puede consultar información sobre este agente SNMP. Esta información es obtenida mediante SNMP y nos permite comprobar si las consultas sobre esa información nos devuelven el mismo valor y por lo tanto comprobar el correcto funcionamiento.