

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SOFTWARE DE
INSTALACIÓN Y CONTROL DE SENSORES EN UNA VIVIENDA
BIOCLIMÁTICA AUTOSUFICIENTE**

PROYECTO FIN DE CARRERA

Cecilio Pato Jiménez de Castro

Mayo 2007

PROYECTO FIN DE CARRERA

Título: Diseño e implementación de un sistema software de instalación y control de sensores en una vivienda bioclimática autosuficiente.

Autor: D. Cecilio Pato Jiménez de Castro

Tutor: D. Álvaro Gutiérrez Martín

Departamento: Tecnologías Especiales Aplicadas a la Telecomunicación

TRIBUNAL CALIFICADOR

Presidente:

D. Francisco Javier Jiménez Leube

Vocal:

D. Javier Macías Guarasa

Secretario:

D. Álvaro Gutiérrez Martín

Suplente:

D. José Julián Chaparro Peláez

Fecha de Lectura:de Mayo de 2007

Calificación:

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SOFTWARE DE
INSTALACIÓN Y CONTROL DE SENSORES EN UNA VIVIENDA
BIOCLIMÁTICA AUTOSUFICIENTE**

PROYECTO FIN DE CARRERA

**Autor:
Cecilio Pato Jiménez de Castro**

**Tutor:
Álvaro Gutiérrez Martín**

Mayo 2007

Resumen del Proyecto

En este proyecto se ha diseñado e implementado una aplicación software que permite gestionar la configuración y el funcionamiento del sistema de control de una vivienda bioclimática autosuficiente, Magic Box. Se ha dotado al sistema de control original de esta vivienda de nuevas mejoras y funcionalidades que hacen que el nuevo sistema de control sea genérico, portable a cualquier localización, dinámico, abierto, modulable y de fácil expansión.

La aplicación desarrollada concede al usuario el control sobre la gestión del sistema de control de la vivienda. Se ha automatizado, en la medida en que el hardware lo permitía, la detección y configuración de los elementos de la red de sensorización del sistema. Así mismo se ha dotado al sistema de la lógica necesaria para actuar sobre el sistema de control de la vivienda, automatizando el reconocimiento de la configuración de las distintas funciones de control sin la necesidad de interrumpir la ejecución del mismo.

Palabras Clave

Magic Box, domótica, hogar digital, sistema de control, automatización, instalación, red de sensorización, nodos, sensores, computación ubicua, inteligencia ambiental, vivienda bioclimática, RS- 485.

Agradecimientos

Estas líneas que ahora escribo no son sólo las que significarán el fin de un proyecto fin de carrera, sino que expresan la conclusión de un proyecto de mucha mayor envergadura. Con estas líneas cierro un ciclo vital cargado de experiencias, emociones, recuerdos, momentos de felicidad y no tan felices, lecciones tanto intelectuales como vitales que han acabado de moldear la persona hoy soy. Pero en definitiva la parte más fundamental de este ciclo que ahora concluye son las personas que me han acompañado a lo largo del mismo, y no quisiera poner punto y final sin expresar mi más sincera gratitud por todos y cada uno de los momentos que han compartido conmigo.

No podría empezar estos agradecimientos por otras personas que no fueran mi familia. Por supuesto mis padres, que siempre me han apoyado, que han disfrutado con tanto o más énfasis que yo mis éxitos, y de igual manera han sufrido con mis momentos más difíciles. Mis hermanas y también amigas, que siempre han confiado en mí y en las que siempre he podido confiar. Y mi padrino Alberto por sus sabios consejos.

En mi paso por Teleco no puedo dejar de agradecer a las personas que han sido mis tutores, consejeros, mentores y padrinos en la universidad (Eve, Miguel Ángel, Califa y Murcia) gracias por todo vuestro apoyo y vuestros consejos. A mi fiel compañera Cris por hacerme tan agradable trabajar con ella. A Isabel por tantísimos momentos. A mis compañeros de fatiga Luís y Cesar y también a mis compañeras de fatiga Leti y Leti, por tantas risas y tan buenos momentos. Y a mis amigos más deportistas, a todos (Nuria, Luismi, Rafa, Javi, Orlando, Nacho...) pero especialmente a Carlos y Lara por vuestra amistad, apoyo y vuestras visitas que tanto bien me hacían.

Gracias a mis amigos, tanto a los que comparten conmigo cada día (Sito, Cas, Jordi, Abel, Ati y Obi) como a los que no veo tanto como quisiera (Javi Ros, Hormi, Depi, Champi, Cesar, Marina, Raquel y Carmen) porque sin todos ellos no sería quien soy, porque consiguen sacar lo mejor de mí y soportan todos mis defectos.

Gracias a Álvaro, mi tutor por tanta paciencia conmigo, por su apoyo, ayuda y colaboración durante todo el proyecto. A José por su ayuda y su toque corporativo y también a Félix por prestarme durante tanto tiempo su laboratorio. Tampoco olvido a Lorenzo, Javier y Fernando por sus consejos como gente con experiencia dentro de la escuela.

A todos ellos, y a muchos que sé que no se encuentran aquí.

Muchas Gracias.

Índice

Índice General

Índice General.....	1
Capítulo 1 Introducción y objetivos.....	6
1.1. Encuadre. Hogar Digital	7
1.2. Magic Box.....	12
1.3. Objetivos del Proyecto.....	15
1.4. Organización del Proyecto.....	17
Capítulo 2 Arquitectura del Sistema.....	19
2.1. Introducción.....	20
2.2. Subsistema Hardware.....	20
2.3 Protocolo de Comunicación	23
2.4 Estructura del Software.....	24
2.4.1. Estructura del Sistema Original de Magic Box.....	24
2.4.2. Estructura del Sistema Actual.....	25
2.5 Organización del Software.....	30
2.6. Conclusiones.....	37
Capítulo 3 Software de Control.....	39
3.1 Introducción.....	40
3.1.1. Introducción a los lenguajes de programación	42
3.2 Objetivos.....	46
3.3 Estructura.....	47
3.3.1 Gestión del Sistema.....	47
3.3.2 Detección de Nodos.....	50
3.3.3 Creación de la Red.....	51
3.3.4 Toma de medidas.....	57
3.3.5 Control.....	57
3.3.6 Elementos Auxiliares.....	67
3.4 Conclusiones.....	73

Capítulo 4 Software de Aplicación.....75

4.1 Introducción.....	76
4.1.1 ¿Por qué PHP?	77
4.1.2 ¿Por qué MySQL?	79
4.2 Objetivos.....	81
4.3 Estructura.....	83
4.3.1 Gestión del Sistema.....	84
4.3.2 Configuración de la Red.....	85
4.3.3 Configuración de los Módulos de Control.....	87
4.3.4 Activación de los Módulos de Control.....	99
4.3.5 Visualización del Estado de la Red.....	100
4.3.6 Librerías Auxiliares.....	101
4.4 Conclusiones.....	102

Capítulo 5 Aplicación Práctica.....105

5.1 Introducción.....	106
5.2 Instalación.....	106
5.3 Sistema de Control.....	107
5.4 Análisis de los Resultados de las Pruebas.....	110
5.4.1 Resultados Prueba nº1	110
5.4.1 Resultados Prueba nº2	114

Capítulo 6 Conclusiones y Líneas Futuras.....120

6.1 Introducción.....	121
6.2 Conclusiones.....	121
6.3 Líneas Futuras de Desarrollo.....	124
6.3.1 Expansión del sistema	125
6.3.2 Utilización de Smart Sensors.....	127
6.2.3 Utilización de una arquitectura Ethernet	129
6.2.4 Utilización de una arquitectura Inalámbrica. Wi-Fi	132
6.2.4 Utilización de una arquitectura Inalámbrica. Bluetooth.....	133

Bibliografía.....135

Bibliografía.....	136
--------------------------	------------

Apéndice A. Manual de Usuario.....140

A.1. Introducción.	141
A.2. Instalación.	141
A.3. Manejo del Sistema.	142
A.3.1 Configuración de la Red	143
A.3.1.1 Instalación de nodos y sensores	144
A.3.1.2 Desinstalación de nodos y sensores	146
A.3.2 Configuración de los módulos de control	146
A.3.2.1 Instalación de módulos de control	147
A.3.2.1 Eliminación de módulos de control	154
A.3.3 Activación de los módulos de control	155
A.3.4 Visualización del Estado de la Red.	156

Índice de Figuras

Figura 1.1 Domótica. Integración de Servicios.....	9
Figura 1.2 Magic Box.....	13
Figura 1.3 Arquitectura Centralizada. Magic Box.....	14
Figura 2.1 Arquitectura Centralizada. Magic Box.....	21
Figura 2.2 Nodo RS-485.....	22
Figura 2.3 Protocolo de Comunicación.....	23
Figura 2.4 Solar Decathlon Control Software.....	25
Figura 2.5 Estructura General Sistema.....	26
Figura 2.6 Funcionalidad Software de Control.....	27
Figura 2.7 Estructura General Software de Control.....	33
Figura 2.8 Estructura General Software de Aplicación.....	35
Figura 2.9 Estructura General Software Auxiliar.....	37
Figura 3.1 Diagrama de Flujo Programa Principal.....	49
Figura 3.2 Diagrama de Flujo tarea Detección.....	52
Figura 3.3 Diagrama de Flujo tarea Instalación.....	56
Figura 3.4 Diagrama de Flujo tarea Toma de Medidas.....	58
Figura 3.5 Diagrama de Flujo tarea Control.....	59
Figura 3.6 Esquema Reconocimiento y Ejecución Control.....	68
Figura 3.7 Estructura y Atributos de la Red Virtual.....	73
Figura 4.1 Arquitectura Web.....	78
Figura 4.2 Software de Aplicación.....	84
Figura 4.3 Menú Principal de Aplicación.....	85
Figura 4.4 Esquema Módulo de Control.....	88
Figura 4.5 Esquema Condición General.....	89
Figura 4.6 Esquema Bloque de Respuestas.....	90
Figura 4.7 Selección de Operación.....	92
Figura 4.8 Selección de Sensor.....	93
Figura 4.9 Selección de Valor Umbral.....	93
Figura 4.10 Selección de Comparador.....	94
Figura 4.11 Selección de Concatenador.....	95
Figura 4.12 Selección de Acción.....	96
Figura 4.13 Niveles de Indexación de Respuestas.....	97
Figura 4.14 Selección de Respuestas.....	98
Figura 4.15 Selección del Relay.....	98
Figura 4.16 Eliminación Módulo de Control.....	99
Figura 4.17 Activación Módulos de Control.....	100
Figura 4.18 Selección de Visualización.....	101
Figura 4.19 Visualización sensores nodo 1.....	102

Figura 5.1 Distribución arquitectura sistema.....	108
Figura 5.2 Resultados Temperatura Vivienda.....	112
Figura 5.3 Resultados Humedad Vivienda.....	114
Figura 5.4 Resultados Humedad Vivienda Día 30.....	114
Figura 5.5 Resultados Humedad Vivienda Día 29.....	115
Figura 5.6 Resultados Temperatura Vivienda Prueba 2.....	116
Figura 5.7 Resultados Actuador Rejilla Norte.....	118
Figura 5.8 Resultados Actuador Rejilla Sur	118
Figura 5.9 Resultados Actuador Turbinas	119
Figura 5.10 Resultados Detalle Rejilla Norte	120
Figura 5.11 Resultados Detalle Rejilla Sur	120
Figura 5.12 Resultados Detalle Turbinas	120
 Figura 6.1 Smart Sensors IEEE 1451.4.....	 129
Figura 6.2 Arquitectura Bus-Serie RS-485.....	132
Figura 6.3 Arquitectura Estrella Ethernet	132
 Figura A.1 Menú Principal de Aplicación.....	 143
Figura A.2.1 Detección de Nodos.....	143
Figura A.2.2 Resultado Detección de Nodos.....	144
Figura A.2.3 Configuración de Nodos.....	144
Figura A.2.4 Configuración de Sensores.....	145
Figura A.2.5 Resultado Instalación de Sensores.....	145
Figura A.2.6 Eliminación de Nodos.....	146
Figura A.3.1 Configuración de Módulos de Control.....	147
Figura A.3.2 Instalación de un Módulo de Control.....	147
Figura A.3.3 Selección del tipo de operación.....	148
Figura A.3.4 Selección operación tipo sensor.....	148
Figura A.3.5 Selección número de sensores para operación media.....	149
Figura A.3.6 Selección múltiple operaciones tipo sensor.....	149
Figura A.3.7 Selección tipo comparador.....	150
Figura A.3.8 Selección operación posterior al comparador.....	150
Figura A.3.9 Selección de Valor Umbral.....	151
Figura A.3.10 Selección del concatenador de condiciones.....	151
Figura A.3.11 Selección de Respuesta Cierta o Falsa.....	152
Figura A.3.12 Selección de Acción.....	153
Figura A.3.13 Selección de Actuador	153
Figura A.3.14 Selección de nuevas respuestas.....	154
Figura A.3.15 Eliminación Módulo de Control.....	154
Figura A.3.16 Confirmación Eliminación Módulo de Control.....	155
Figura A.4.1 Activación Módulos de Control.....	156
Figura A.5.1 Selección de Visualización.....	156
Figura A.5.2 Visualización sensores nodo 1.....	157

Capítulo 1

Introducción y objetivos

1.1. Encuadre. Hogar Digital

La “Sociedad de la Información” ya no es un concepto de futuro, sino que vivimos inmersos en ella. Con el desarrollo de las nuevas tecnologías de la información y la comunicación (TIC), la información y el conocimiento han pasado a jugar un papel fundamental en nuestra vida cotidiana y se han convertido en el principal soporte de la economía.

Los cambios propiciados por el avance de la tecnología han revolucionado todos nuestros ámbitos de actividad (finanzas, educación, sanidad, etc.) y fundamentalmente la forma de acceso a la información. Estos cambios permiten, entre otras muchas cosas, hacer más eficiente nuestro tiempo, independientemente de si estamos en el trabajo, de viaje o en nuestra casa.

La tendencia de la evolución de la tecnología es independizar completamente los servicios ofrecidos al usuario de su ubicación. Términos como inteligencia ambiental y computación ubicua son hoy de máxima actualidad. Las principales empresas tecnológicas están realizando una apuesta por desarrollar tecnologías que se adapten al usuario, satisfaciendo sus necesidades sin que el destinatario necesite ordenar conscientemente ese tipo de servicios.

Se puede definir un entorno como inteligente “cuando se encuentra saturado de ordenadores y capacidades de comunicación, integrados fácil y amigablemente con los usuarios, de modo que la tecnología se utiliza tan profusamente que tiende a desaparecer de la conciencia de los usuarios.” [1]

El concepto de ambiente inteligente enfatiza la facilidad de uso y se materializa en un individuo rodeado de interfaces inteligentes e intuitivas que se encuentran integradas en partes y objetos corrientes. Para ello es necesaria la integración de múltiples tecnologías, como redes de sensores, control,

automatización, microservidores, comunicaciones inalámbricas, localización, interfaces multimodales, etc.

Este entorno mencionado no se limita sólo a la vivienda, sino que comprende todo tipo de lugares físicos de la vida de un usuario normal como el coche o el lugar de trabajo. Aún así la vivienda, en particular, está experimentando una inmensa transformación para adaptarse a las nuevas necesidades de las familias, que necesitan vivir en entornos más flexibles para compaginar de la mejor forma posible sus tareas domésticas con el campo profesional o el ocio.

Internet, ha sido un claro impulsor de esta evolución en las viviendas, ha revolucionado el mundo de las comunicaciones y ha propiciado el auge de productos y servicios que permiten la convergencia entre la informática, las comunicaciones y la electrónica de consumo, aportando un valor añadido incomparable para el usuario.

Para gestionar estos cambios en la vivienda y proveer servicios integrando distintas tecnologías aparece una nueva disciplina, la Domótica, pensada principalmente para aumentar el confort y la calidad de vida dentro de la vivienda.

Según la Asociación Española de Domótica, (CEDOM) [2], ésta disciplina se define como “la incorporación al equipamiento de nuestras viviendas de una sencilla tecnología que permite gestionar de forma energéticamente eficiente, segura y confortable para el usuario, los distintos aparatos e instalaciones domésticas tradicionales que conforman una vivienda, como pueden ser la calefacción, la iluminación, etc. La domótica busca el máximo aprovechamiento tanto de la energía como de la luz solar, adecuando su comportamiento a nuestras necesidades”.

De ésta y otras definiciones podemos concluir que la esencia que caracteriza todo sistema domótico es la **integración**, la integración de todos los sistemas eléctricos y automáticos de los que se dispone en una vivienda, bajo un mismo control, permitiendo así la interacción entre ellos. Un sistema domótico debe satisfacer todas las necesidades de forma global y en conjunto. En otro caso no puede hablarse de domótica, sino simplemente de la automatización de diferentes actividades.

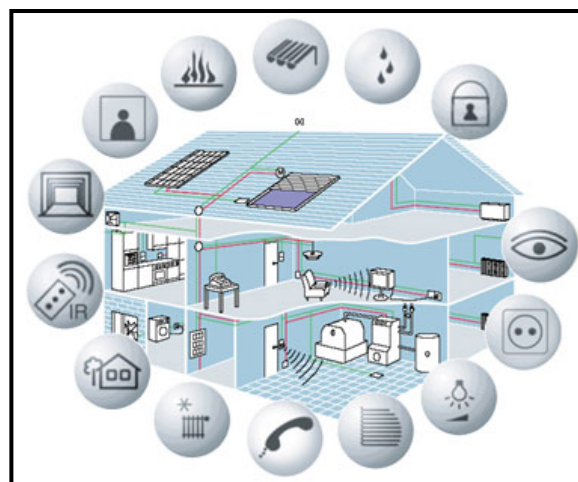


Figura 1.1 Domótica. Integración de Servicios

Algunas de las prestaciones que la domótica puede aportar a las viviendas se resumen en las siguientes [3]:

- Gestión u optimización energética.
- Aumento del nivel de confort y del bienestar en la vivienda
- Aumento considerable de la seguridad, tanto a nivel técnico como personal.
- Teleasistencia, para dar atención a las personas discapacitadas y de la tercera edad en el instante en el que lo precisen.
- Telemantenimiento de sistemas e instalaciones, con el consiguiente ahorro en el desplazamiento de los técnicos al domicilio.
- Teletrabajo, posibilidad de trabajar desde el domicilio particular.

- Control remoto de la vivienda, mediante teléfono, Internet, ordenador de sobremesa o portátil, televisión u otros medios de la instalación domótica dispuesta en nuestra vivienda.
- Posibilidad de optimizar los recursos naturales en su conjunción con técnicas de diseño arquitectónico naturales.
- Potenciación de las comunicaciones, ya que se añaden a la casa servicios telemáticos y acceso a los diferentes servicios que proporciona Internet.

Todas estas prestaciones y posibilidades que nos puede ofrecer un sistema domótico se pueden agrupar dentro de cinco pilares básicos:

- **Seguridad:** como sistemas de alarma, detección de situaciones peligrosas (incendios, gas, inundaciones, etc.), simulación de presencia (luces, persianas, etc.)
- **Confort:** control de variables ambientales como iluminación, calefacción, climatización (aire acondicionado), ventilación, etc.
- **Economía doméstica:** ahorro de energía, gestión de existencias (despensa 'informatizada'), etc.
- **Comunicaciones:** control y gestión de comunicaciones privadas y públicas, conectividad con proveedores de servicios (por Internet u otros canales específicos), control a distancia del entorno doméstico, incluso desde fuera del domicilio.
- **Entretenimiento:** actividades lúdicas diversas (música, juegos, espectáculos, ambientación), tanto localmente como a través de proveedores de servicios (por ejemplo, mediante sistemas "pay per view")

En definitiva, la tecnología está modificando la idea tradicional de hogar, consolidando el concepto del Hogar Digital. La vivienda del siglo XXI será un Hogar Conectado, pero al mismo tiempo deberá suponer una clara apuesta por la sostenibilidad y el medio ambiente.

El Hogar Digital [4] es el lugar físico donde las necesidades de la familia, en materia de seguridad y control, comunicaciones, ocio y confort, integración medio ambiental y accesibilidad, son atendidas mediante la convergencia de

servicios, infraestructuras y equipamientos. El objetivo de todo Hogar Digital [5] es el de cubrir todas y cada una de las necesidades de sus habitantes de manera que estos tengan un control íntegro de la casa, consiguiendo con ello incrementar la seguridad, el confort, mejorar las comunicaciones, ahorrar energía, dinero y tiempo, facilitar la oferta de nuevos servicios, etc.

La vivienda del futuro va a tener una composición estructural en función de las distintas tipologías de usuarios, pero tiene que adoptar unos criterios mínimos de sostenibilidad en el diseño y la construcción, necesidades de accesibilidad y movilidad en el interior, una infraestructura de comunicaciones que constará de una red doméstica y la pasarela de comunicaciones que tiene que permitir la integración y el acceso externo a redes de banda ancha, y todo esto acompañado de la automatización de las instalaciones tradicionales del edificio y la vivienda.

Las instalaciones comunes del hogar son entre otras las de iluminación, suministro eléctrico, antenas de televisión, televisión, telefonía, sistemas de acceso al recinto, alarmas, sistemas de climatización, etc. No es casual que el equipamiento electrónico en los hogares vaya en aumento: informático, audiovisual, de comunicaciones y doméstico, al mismo tiempo que incorpora nuevas características y deja de operar de manera aislada. El objetivo de los nuevos sistemas y equipamientos del hogar, es la integración y el control de varios sistemas a la vez permitiendo su automatización y control. Es por esto que resulta tan importante fomentar la necesidad de conectar los dispositivos electrónicos entre sí, mediante estas redes de hogar, y con el exterior a través de Internet para poder disfrutar de servicios cada vez más avanzados.

La convergencia entre las telecomunicaciones y la informática ha abierto muchas posibilidades con el tendido de banda ancha. Las redes del Hogar (Home Networking), se transformaron poco a poco en redes multipropósito, con

servicios convergentes de voz, datos, audio, video, control y más, promoviendo de este modo ambientes residenciales más interactivos y eficientes.

A continuación se presenta **Magic Box**, una vivienda domótica, bioclimática y autosuficiente que representa un claro ejemplo de la evolución de lo que será la vivienda del siglo XXI y su compromiso con la sostenibilidad y el medio ambiente.

1.2. Magic Box

En otoño del 2005 se celebró en Washington D.C. la segunda edición del concurso internacional Solar Decathlon patrocinado por el Departamento de Energía de los Estados Unidos. Solar Decathlon [6][7] es un concurso dirigido a universidades que persigue difundir la posibilidad de conciliar las buenas prácticas arquitectónicas con un uso racional de la energía, a través del aprovechamiento de la energía solar tanto pasiva como activamente y del uso de tecnologías eficientes actualmente disponibles.

La Universidad Politécnica de Madrid participó en este concurso con un equipo formado por más de 30 alumnos y 7 profesores de centros distintos como la Escuela de Arquitectura, el Instituto de Energía Solar y el Centro de Domótica Integral. El trabajo de este equipo y la integración de las distintas disciplinas se materializó en la propuesta presentada: **Magic Box**.

Magic Box es una vivienda de 70 m² alimentada exclusivamente por energía solar que responde a las necesidades de un hogar de nuestros días y muestra el uso práctico de las energías renovables en la vida cotidiana. Consta de una cocina, un comedor, un dormitorio, un despacho, un baño, un salón, diferentes terrazas y un cuarto exterior de instalación eléctrica y baterías.



Figura 1.2 Magic Box

Magic Box es una vivienda ecológicamente sostenible, bioclimática, autosuficiente, y automatizada. En su construcción se han utilizado materiales sostenibles, algunos de ellos reciclados y la mayoría de ellos reciclables.

Como es un prototipo su precio es mucho mayor que una casa convencional y mayor que si la misma casa fuera desarrollada comercialmente (importe total de construcción 420.000€) [6]. El ahorro de energía es completo puesto que la casa no consume energía de la red, su consumo está completamente basado en la energía solar fotovoltaica y térmica.

En este proyecto la domótica tuvo un papel muy importante para la integración de los automatismos eléctricos, informáticos, electrónicos, de comunicaciones, confort y ahorro energético. La primera decisión importante para el grupo de investigación domótica del proyecto fue la de analizar los diferentes sistemas domóticos existentes en ese momento (X10, Lonworks, Konnex, etc.) para ver si alguno de ellos se adaptaba a las necesidades del proyecto [8][9][10][11][12] .

Un análisis en profundidad de dichas tecnologías reveló que ninguna de ellas encajaba dentro de los requisitos del proyecto. Las razones de esto fueron muy variadas, desde la falta de flexibilidad del sistema y la imposibilidad de encontrar componentes comerciales con la precisión necesaria, hasta

problemas de falta de documentación que imposibilitaban desarrollos adaptados a las necesidades del proyecto, pasando por problemas de licencias o elevados costes de desarrollo de prototipos.

Por todo ello se tomó la decisión de desarrollar un sistema domótico propio. Esta apuesta se tradujo en la necesidad de desarrollar tanto los componentes hardware como los elementos software.

Para el subsistema hardware se escogió [13] una arquitectura centralizada en la que un sistema de control central recibe información de los diferentes sensores de la vivienda y es capaz de modificar el estado de los equipos e instalaciones empleando una serie de actuadores.

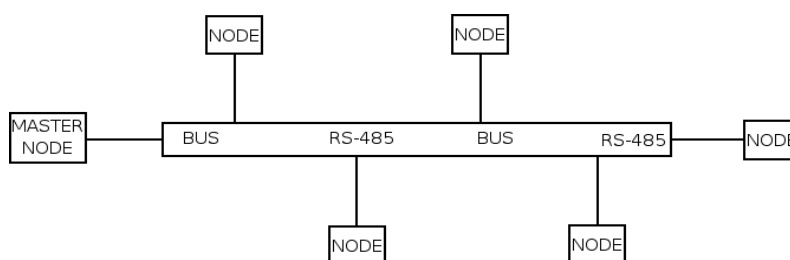


Figura 1.3 Arquitectura Centralizada. Magic Box.

La plataforma del hogar digital desarrollada para Magic Box es el elemento software que da cohesión a los diferentes componentes del sistema domótico desplegado en la vivienda: desde la comunicación con la red de control hasta la presentación e interacción con el usuario.

Se pretendió que la experiencia de la participación en Solar Decathlon no finalizase con el concurso, sino consolidar la colaboración iniciada y continuar las investigaciones con el objetivo de analizar las posibilidades de combinar la arquitectura bioclimática, las tecnologías energéticas renovables y los sistemas de gestión domóticos en el diseño de viviendas modernas y atractivas que

contribuyeran a introducir criterios de sostenibilidad en el sector inmobiliario español.

Con esta intención, en Diciembre de 2003 se presentó al Programa de Construcción del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica 2004-2007 la propuesta **“Heliodomo: nuevo concepto de vivienda autosuficiente”**. Los trabajos que se extenderán hasta finales del 2007, combinarán la realización de análisis teóricos relacionados con las fases de diseño, elección de materiales y sistemas de construcción, con la realización de campañas experimentales sobre prototipos reales, el desarrollo de métodos para garantizar la calidad de los sistemas energéticos renovables y el desarrollo de sistemas de gestión domóticos apropiados.

Éste proyecto trata de continuar con esa línea de investigación utilizando para ello uno de los prototipos de vivienda “Magic Box” instalada en la Escuela Técnica Superior de Ingenieros de Telecomunicación.

1.3. Objetivos del Proyecto

Este Proyecto Fin de Carrera plantea la implementación de una aplicación, que permita manejar de forma abierta la configuración y el diseño de redes de nodos y sensores, que nos ayude a controlar múltiples facetas del hogar. Así, podremos actuar sobre elementos de confort como temperatura o humedad, elaborar funciones de ahorro de energía o configurar a elección del usuario diversos tipos de alarmas.

Se procura también que el código sobre el que funciona esta aplicación, permita modificaciones en la configuración de la red y parametrización de sus sensores sin complicaciones para futuros usuarios y/o programadores, y gestionar la información relacionada con los diferentes componentes del

sistema domótico de tal forma que sea presentada de manera sencilla e intuitiva al usuario.

Por lo tanto los objetivos fundamentales del proyecto son:

- Analizar las características de los sensores y parámetros fundamentales para su configuración y funcionamiento.
- Configurar la comunicación entre los sensores, actuadores y nodos de nuestra red con el ordenador central desde el que ejerceremos el control del hogar.
- Configurar los elementos de comunicación entre el ordenador central que controla la red del hogar y el usuario que proporcionará las instrucciones de control que deben ser interpretadas y ejecutadas.
- Elaborar un software modulable y abierto que permita modificar los distintos elementos de la red de control con facilidad, de una manera intuitiva y dentro de lo posible sin necesidad de paralizar el sistema y de modificar y recompilar código.
- Incluir en la aplicación distintos sistemas de control de los diversos dispositivos. Permitir su selección, configuración y la posibilidad de incrementar el control a través de la introducción de nuevos módulos.

Para la implementación de los objetivos se atenderá a las siguientes condiciones para el diseño:

- **Abierto:** Realizar un desarrollo basado en estándares abiertos para que el sistema pueda ser actuado directamente por el usuario sin ninguna condición o restricción impuesta por la compañía que controle la infraestructura utilizada.
- **Escalable:** El sistema debe ser fácilmente ampliable. De este modo se podrán aumentar sus capacidades cuando el usuario desee expresarlo en mayor medida añadiendo nuevas funciones de control o seguridad.
- **Modulable:** Debido a la extensión del código éste debe estar claramente distribuido para facilitar el acceso a información específica para nuevos usuarios o programadores no familiarizados con él.

En resumen con este proyecto se ha pretendido continuar con la labor de trabajo e investigación realizada para el desarrollo de Magic Box, y se ha centrado el trabajo en dos aspectos:

- Automatizar en lo posible la instalación, detección y configuración de la red de nodos y sensores empleados en Magic Box, teniendo en cuenta las limitaciones del Hardware existente.
- Facilitar la incorporación al código de distintas funciones de control a elección del usuario, de manera que puedan añadirse y seleccionarse distintos sistemas de control sin dificultad y sin paralizar el funcionamiento del sistema.

Del mismo modo se desea que este proyecto sirva en un futuro como base para otros desarrollos. Por ello es sumamente interesante realizar un sistema abierto, siguiendo los estándares de comunicación y representación de información que permitan dichos desarrollos.

1.4. Organización del Proyecto

Este tomo está estructurado en los siguientes capítulos.

El Capítulo 1 corresponde a la introducción. Encuadra el proyecto a través de una breve descripción del contexto en el cual se desarrolla, revisando brevemente términos como Domótica, Hogar Digital y finalmente situándonos en una vivienda real, Magic Box.

En el Capítulo 2 se describe la arquitectura del sistema. Se presentarán los elementos Hardware y el protocolo de comunicación utilizados. Asimismo se explicará el modo en el que se ha decidido organizar el Software y el funcionamiento general de la aplicación desarrollada.

En el Capítulo 3 se aborda la descripción del sistema de control implementado. Se analizará lo que constituye el motor fundamental del

sistema. Este elemento se encarga de la comunicación entre el ordenador central y los elementos de la red, y es el responsable del reconocimiento de las funciones de control y el funcionamiento del sistema completo.

El Capítulo 4 corresponde al sistema de aplicación. Este sistema constituye el interfaz entre el usuario y la máquina. Se encarga de gestionar la configuración de los elementos de control del sistema y el manejo de las distintas bases de datos.

En el Capítulo 5 se presentan las conclusiones finales y las líneas futuras del proyecto.

Capítulo 2

Arquitectura del Sistema

2.1. Introducción.

Este capítulo analiza la estructura del sistema de control que se ha diseñado en este proyecto para la vivienda Magic Box. Se analizarán en primer lugar los elementos físicos que constituyen el Hardware del sistema (bus del sistema, los nodos y los sensores y actuadores) y el protocolo de comunicación utilizado para la correcta comunicación entre el ordenador central y los distintos elementos Hardware.

Posteriormente se analizará la estructura del Software, de manera que se puedan apreciar los cambios y mejoras que ofrece el funcionamiento general del nuevo sistema implementado frente al sistema original.

Una vez se haya proporcionado al lector una idea clara del funcionamiento general del sistema se pasará a detallar la manera en la que se integran los distintos elementos del software desarrollado de una manera modulable y escalable.

Se finalizará el capítulo con unas conclusiones generales del desarrollo de la arquitectura del sistema.

2.2. Subsistema Hardware.

La arquitectura con la que se ha trabajado proviene del subsistema hardware desarrollado para el concurso Solar Decathlon 2005. En esta sección se describen los principales componentes hardware del sistema y su misión a desarrollar, sin entrar en demasiados detalles de la arquitectura ya que puede encontrarse abundante información en [13][14][15].

El subsistema hardware utiliza una arquitectura centralizada en la que un sistema de control central recibe información de los diferentes sensores de la

vivienda y es capaz de modificar el estado de los equipos e instalaciones empleando una serie de actuadores.

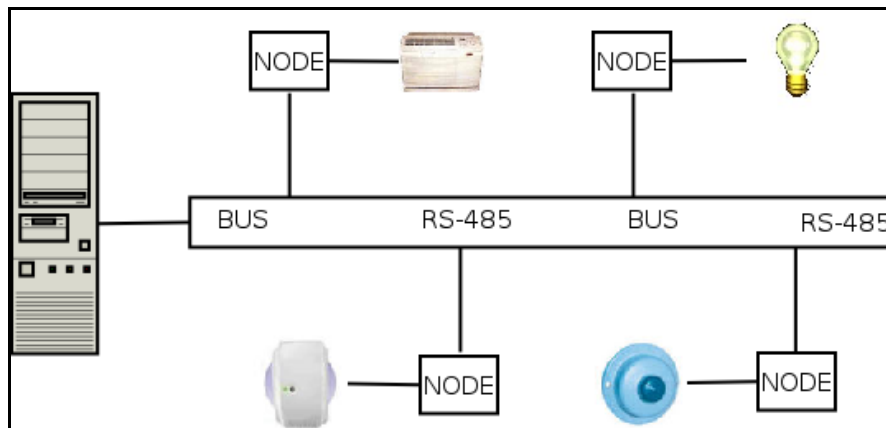


Figura 2.1 Arquitectura Centralizada. Magic Box

Durante la primera fase del diseño de Magic Box se estudiaron diferentes estándares de medida y control remotos, con el objeto de definir la tecnología base sobre la que se basaría el sistema de monitorización y control a incorporar en la vivienda. Este estudio se centró en las posibilidades que ofrecían tecnologías como: Lonworks (Estándar Americano de bus industrial-domótico), EIB-KNX (Estándar Europeo Domótico), PLC (Autómatas programables), X10, u otros sistemas propietarios [10][11][12].

Finalmente se optó por implementar un sistema propietario sobre la capa física de la norma RS485 [13], esta decisión obligó al diseño a medida de los elementos sensores y actuadores, ofreciendo a cambio la versatilidad de desarrollar un sistema de control y actuación adaptado a las necesidades del proyecto.

El bus de control es por lo tanto un sistema diseñado e implementado para Magic Box desde la capa física a la de aplicación y está basado en estándares industriales. Este tipo de bus industrial (RS-485) nos permite disponer de un sistema robusto, sin contar con que, al tener un diseño propio la capacidad de

maniobra frente a diferentes cambios que se quieran realizar dentro de la vivienda es bastante rápida, sin necesidad de depender de terceras personas y disponibilidad comercial.

Centrándonos sobre la arquitectura del sistema, comentar que la transmisión de información se realizará a cuatro hilos. El ordenador central se encargará de realizar las peticiones de información y los nodos escucharán y responderán. Los resultados obtenidos de los sensores serán analizados por el subsistema software de control que determinará las acciones que deberán realizarse. Estas órdenes serán de nuevo enviadas a los nodos del bus que actuarán sobre los elementos físicos de la vivienda.

El nodo utilizado (NOD485) dispone de entradas analógicas para la adquisición de datos a las que podemos conectar sensores (en principio de humedad, luz y temperatura) y es capaz de actuar tanto sobre relés convencionales como sobre triacs, en función de la tipología de los dispositivos en base a su funcionamiento en corriente continua o alterna.

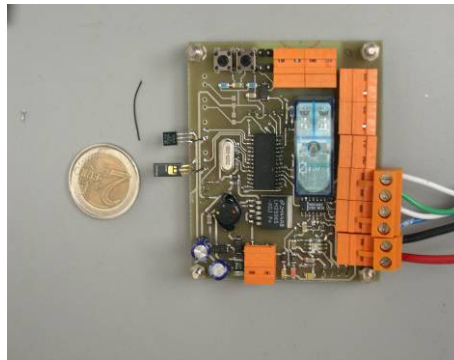


Figura 2.2 Nodo RS-485

De esta manera nuestro sistema se puede expandir de una forma muy sencilla; basta con conectar nuevos sensores a las entradas analógicas, y analizar dichos datos en el sistema central.

2.3. Protocolo de Comunicación

El protocolo de comunicación implementado está basado en caracteres ASCII. Aunque el uso de este tipo de caracteres no sea tan común en estándares informáticos, su uso resulta una práctica muy habitual en estándares industriales orientados a automatismos.

En el protocolo de comunicación se incluyen caracteres de control y codificación de los valores a enviar en ASCII, para evitar que existan ambigüedades en las tramas enviadas. Todos los envíos se realizan mediante tramas cuya estructura está bien definida, como se puede observar en la Figura 2.3:



Figura 2.3 Protocolo de Comunicación

- **STX (0x02):** Inicio de transmisión. Este carácter es el que delimita el comienzo de una trama de datos.
- **DIR:** Son 2 bytes que indican en ASCII el valor hexadecimal de la dirección del nodo que envía los datos cuando la trama es enviada por un nodo, y la dirección de la tarjeta-nodo a la que se dirige la trama cuando ésta es enviada por el PC.
- **Número de trama:** 2 bytes que indican en ASCII el valor hexadecimal del número de la trama que se envía.
- **Datos:** El número de datos enviados en cada trama es variable, y dependerá del tipo de información que contenga (temperatura, humedad, luz, actuador, comando etc.).
- **CRC-16:** 2 bytes que sirven para la detección de errores en la transmisión. El CRC generado corresponde a toda la trama, salvo los caracteres de inicio y final de transmisión y el propio CRC-16.
- **EOT (0x04):** Final de transmisión. Indica el final de la transmisión en curso.

Cualquier tipo de información que se transmita entre los nodos y el PC irá encapsulada de esta forma, la incorporación al sistema de nuevos dispositivos agregando nuevas tramas de datos es trivial, siempre y cuando se mantenga la misma estructura de trama.

2.4. Estructura del Software.

El propósito de este apartado es proporcionar al lector una visión global de la estructura del sistema completo. De éste modo se define el funcionamiento general del sistema y a su vez se acerca al lector la utilidad y necesidad de algunas funciones o elementos del código que se comentarán posteriormente en detalle.

2.4.1. Estructura del Sistema Original de Magic Box.

En el proyecto original se planteó la posibilidad de crear procesos distintos para el control y las comunicaciones con el bus y para la comunicación con el exterior sin embargo se descartó dotar de capacidades de comunicación al módulo de control para no introducir demasiada complejidad al sistema.

De esta manera se desarrolló **sdcs** (Solar Decathlon Control Software), el núcleo de la plataforma del hogar de Magic Box. Se agruparon en este componente todos los elementos que constituían el sistema de control. Las salidas del sistema generaban por un lado las medidas instantáneas de cada sensor, así como la estructura de los nodos con sus sensores respectivos.

Posteriormente con un sistema de plantillas (Smarty), una librería libre (XML-RPC for PHP) y con PHP se implementaba la aplicación web para la visualización de los resultados. La visualización permitía mostrar las direcciones de nodo válidas, la información de un nodo y la estructura de datos con la lista de los nodos y los sensores asociados a cada uno de ellos [14].

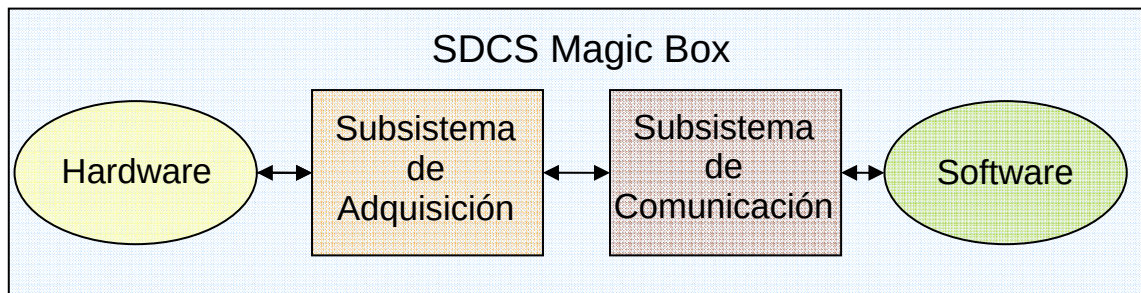


Figura 2.4 Solar Decathlon Control Software

2.4.2. Estructura del Sistema Actual

En la estructura del sistema diseñado para este proyecto se pueden discernir dos bloques claramente diferenciados que forman la mayor parte del software del proyecto. De aquí en adelante nos referiremos a ellas como Software de Control y Software de Aplicación.

El Software de Control se encarga de la correcta ejecución del sistema y la comunicación entre el ordenador central y los elementos del Hardware, este subsistema software está implementado en C y C++. El Software de Aplicación lo forma una aplicación Web, diseñada en PHP y HTML, que se encarga de servir de interfaz con el usuario y generar las instrucciones que sean precisas para que sean ejecutadas por el control del sistema.

Entre estas dos capas principales se sitúa otra de menor envergadura pero igualmente imprescindible, este estrato está formado por un conjunto de archivos XML y varias bases de datos MySQL, que proporcionan el soporte necesario para trasladar la información entre las dos capas fundamentales de una manera estructurada y que ambos sistemas puedan comprender.

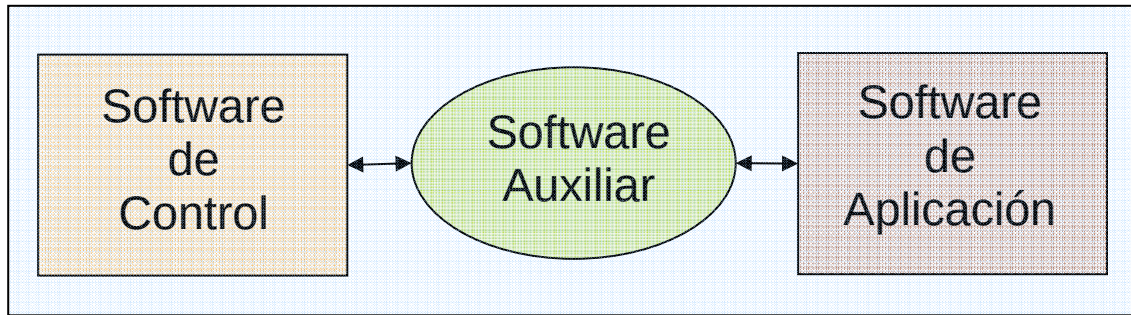


Figura 2.5 Estructura General Sistema

Conviene recordar que los principales objetivos de este proyecto eran automatizar en lo posible la instalación, detección y configuración de la red de nodos y sensores empleados en Magic Box, así como facilitar la incorporación de funciones de control al sistema, de manera que puedan añadirse y seleccionarse distintos módulos de control sin dificultad. Estos objetivos nos van a definir los pasos fundamentales de la operativa del sistema que pasamos a detallar a continuación.

El motor fundamental del funcionamiento de todo el sistema es el componente software de control, que heredando la nomenclatura original llamaremos **sdcs**. Este componente estará en constante ejecución esperando las instrucciones que envíe el usuario por medio del interfaz de la parte de aplicación. Dichas instrucciones pueden pertenecer a cuatro tipos de operaciones:

- Configuración de nodos y sensores.
- Configuración de módulos de control.
- Activación de módulos de control.
- Visualización del estado de medidas de los sensores.

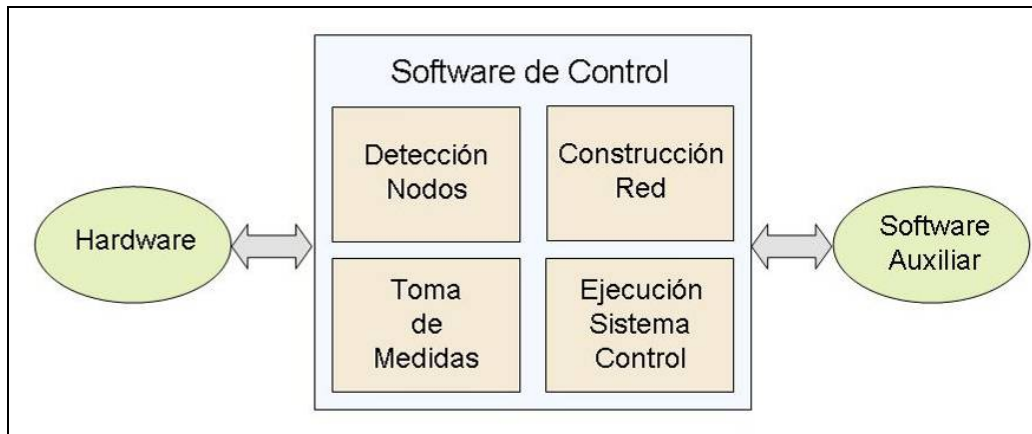


Figura 2.6 Funcionalidad Software de Control

- **Configuración de nodos y sensores.**

El programa principal espera a que el usuario decida crear la configuración de nodos y sensores. En ese momento el programa principal detecta los nodos del sistema que están en el bus y se encuentran activos. La aplicación se lo notifica al usuario informándole del estado de cada uno de los nodos detectados en la red. En caso de ser un nodo nuevo permite la opción de instalarlo y si el nodo estaba previamente instalado permite su desinstalación.

En el caso en que se desee instalar un nuevo nodo la aplicación solicitará al usuario la información sobre el nodo y sobre los sensores pertenecientes a él.¹ Si la opción deseada fuera desinstalar un nodo, el programa obligará a confirmar la operación advirtiéndole de sus posibles consecuencias.

Cada una de estas acciones introducirá o eliminará la información del nodo de una base de datos y comunicará al programa principal que la configuración de la red ha variado para actualizar su comportamiento.

En el momento en que el programa principal detecta un cambio en la configuración del sistema de nodos revisa su propia estructura de red y la

¹ El Hardware no permite la detección de los sensores.

actualiza. Mientras el programa principal tenga una estructura de red creada se realizan medidas en todos los sensores constantemente. El programa recorre los sensores de cada uno de los nodos, detecta el tipo de sensor que es y opera para obtener la medida correspondiente. Esta medida se introduce en la red virtual creada por el sistema de control.

Una vez finalizadas todas las medidas se almacenan en diferentes archivos XML para su posterior utilización. Un archivo guarda la medida instantánea, otro guarda la medida anterior realizada y en otro se guarda un histórico de todas las medidas realizadas. Se han creado estos archivos con vistas a su utilización en posibles sistemas de control en los que interese evaluar además del estado actual, la evolución con el tiempo de las medidas de los sensores, así como la evaluación del sistema de control.

- **Configuración de módulos de control.**

Uno de los principales objetivos del proyecto es incorporar la posibilidad de incluir todo tipo de funciones de control sin tener que detener la ejecución del programa principal y lo que es más interesante sin la necesidad de tener que modificar el código subyacente y tener que volver a compilarlo.

Se ha creado un sistema mediante el cual el usuario puede introducir la función de control que desee con la única limitación de los tipos de sensores y tipos de funciones que en ese momento haya configurados. De forma automática el programa es capaz de analizar la instrucción deseada y ejecutarla.

Para ello la aplicación irá guiando al usuario en la configuración de su propia función de control introduciendo paso a paso cada uno de los elementos. Así se logra estructurar un módulo de control que estará formado por una o varias condiciones de valores de sensores, de funciones o numéricos. La parte final de este módulo de control estará formada por una o varias respuestas o

acciones que llevar a cabo en función del cumplimiento de dichas condiciones. Estas respuestas a su vez pueden incluir algún condicionante para su ejecución.

La aplicación almacena cada elemento del módulo de control de manera que el programa principal pueda interpretarlo correctamente, fabricarse su propia función, evaluarla y comunicar a los elementos de red necesarios las acciones a llevar a cabo.

Como medida de seguridad dentro de la información de cada módulo de control se añade también una lista de dependencias de los nodos que el módulo necesita que estén activos para su correcto funcionamiento. Esta lista se revisa en el momento de realizar la función de control y se comprueba que efectivamente todos y cada uno de los nodos responden, procediendo a su ejecución sólo en caso de que esto se cumpla.

De la misma manera que la aplicación permite añadir funciones de control también ofrece la posibilidad de eliminar módulos de control que ya no se desean. En este caso muestra al usuario toda la información disponible de todos los módulos con un panel para seleccionar cual eliminar. Una vez realizada la selección exige una confirmación antes de borrar la información permitiendo volver al menú principal en caso de que no se desee continuar con la operación.

- **Activación de módulos de control.**

Para hacer efectiva la ejecución de los módulos de control todos se inicializan desactivados. Así se evita que pudieran aparecer problemas si se empezara a ejecutar un módulo de control sin que estuviera completamente configurado.

El usuario puede alterar, en cualquier momento, el conjunto de módulos de control que desea que se ejecuten y todo ello sin modificar el funcionamiento

normal del programa principal. Se le presentan los módulos de control instalados con una descripción de su funcionamiento, sus dependencias y su estado actual. Con esta información se facilita al operador la tarea de seleccionar qué módulos activar y cuales dejar inactivos.

El programa principal revisa constantemente cuales son los módulos de control que tiene configurados y cuales están activos. Antes de ejecutar cada uno de ellos comprueba el estado de los nodos implicados en sus funciones de manera que si uno de los nodos de los que el módulo depende no se encuentra activo el módulo de control no se ejecuta. De esta manera evitamos que el sistema general deje de funcionar si hay un problema con un nodo o si se elimina uno por error.

- **Visualización del estado de medidas de los sensores.**

En esta parte de la aplicación se da la oportunidad al usuario de comprobar el estado de la red visualizando las medidas de los sensores. Para ello se le ofrece presentar la información de los sensores agrupados de diferentes maneras:

- Toda la red.
- Agrupados por tipo de sensores.
- Agrupados por nodo al que pertenecen.
- Agrupados por zona en la que el nodo está instalado.

Esta acción no interactúa con el programa principal solo muestra los datos obtenidos de la última medida realizada.

2.5. Organización del Software.

Para la realización de este proyecto siempre se han tenido como principales objetivos realizar un sistema abierto y escalable de manera que cualquier otro

investigador interesado pudiera manejar y expandir la aplicación desarrollada sin dificultad. Por este motivo se ha procurado hacer una implementación modulable diferenciando los distintos documentos por su tipo y su funcionalidad.

La estructura de este proyecto mantiene básicamente el esqueleto del original, aunque ampliando las funcionalidades y expandiendo el sistema para darle una mayor flexibilidad, introduciendo algunas variaciones importantes.

Para lograr que la configuración de la red deje de ser estática, y dotarla de ciertas propiedades de instalación, desinstalación y diseño, se requiere la interacción entre el sistema de control y el usuario. Esta circunstancia es también aplicable al funcionamiento de la configuración de los módulos de control. Por lo tanto ha sido necesario incorporar al sistema una capacidad de comunicación entre la aplicación de control y el usuario.

Se ha optado por una organización en capas para cubrir cada una de las necesidades del sistema. Conviene recordar la distinción entre el Software de Control y el Software de Aplicación introducidos en el apartado anterior, y la capa intermedia auxiliar que nos facilita la comunicación entre los dos bloques principales.

Pasamos a un análisis un poco más detallado de la organización de estas capas:

- **Software de Control**

Este componente constituye la capa más inferior del software elaborado. Se encarga del sistema de adquisición de datos y en general de la comunicación con los elementos del hardware a través del bus de comunicación.

Como se presentó en el apartado anterior se han mantenido el hardware y el protocolo de comunicación originales. Por esta razón se han conservado

básicamente aquellas funcionalidades relacionadas con toda la operativa del bus de comunicaciones, con pequeñas adaptaciones a las características del nuevo software.

Este software se ha desarrollado mayoritariamente en C, aunque para algunos de sus elementos se ha optado por utilizar C++ aprovechando así las posibilidades que ofrecen los objetos, clases, polimorfismo y herencia de clases. De esta forma se ha utilizado la capacidad de crear objetos para generar una red virtual de nodos y sensores para la realización de todos los procesos requeridos.

El Software de Control está agrupado en una carpeta de nombre *sdcs* para continuar con la terminología heredada. Dentro de ella a su vez se han separado los elementos de código C y C++ de sus respectivos archivos cabecera. De este modo se facilita la localización y el acceso a cada uno de estos archivos, organizándolos de una manera estructurada y a la vez intuitiva.

Esta intención se ha mantenido a la hora de distribuir el código en los diferentes archivos que componen este bloque, de tal manera que tenemos el motor fundamental del sistema (*main.c*) que en su ejecución puede lanzar hasta cuatro hebras distintas, cada una corresponde a una funcionalidad y están definidas en archivos distintos. Por un lado tenemos la tarea encargada de la detección de los nodos activos cuando se desea configurarlos (*bus.c*) y otra se ocupa de reconocer la estructura de la red deseada y la construcción de la red virtual con objetos en C++ (*instal.c*). Siempre que haya una red virtual creada se ejecutará la tarea encargada de recorrer la red y obtener las medidas de todos los sensores (*measures.c*). La última tarea realizable corresponde a la interpretación y ejecución de las funciones de control definidas y activas (*control.c*).

También se han separado otros archivos con utilidades muy claras. Así las funciones específicas de las preguntas y las acciones en los sensores se engloban en *asking.c*. Las distintas funciones implementadas para operaciones en la parte de control se encuentran en *function.c*. Los elementos necesarios para operar con la red virtual se distribuyen en *red.cpp*, *nodo.cpp* y *sensor.cpp*. Y por último hay utilidades generales para las comunicaciones con el bus en *communications.c* y para el manejo de los archivos auxiliares de XML en *xmlutil.c*.

Todos estos archivos se agrupan en el bloque **sdcs** junto con el archivo ejecutable del programa *sdsp*, el archivo *Makefile* para la correcta compilación de todos los componentes y la carpeta *lib* que contiene todos los archivos cabecera correspondientes a los ya mencionados del código fuente. (Ver apéndice Manual de Usuario).

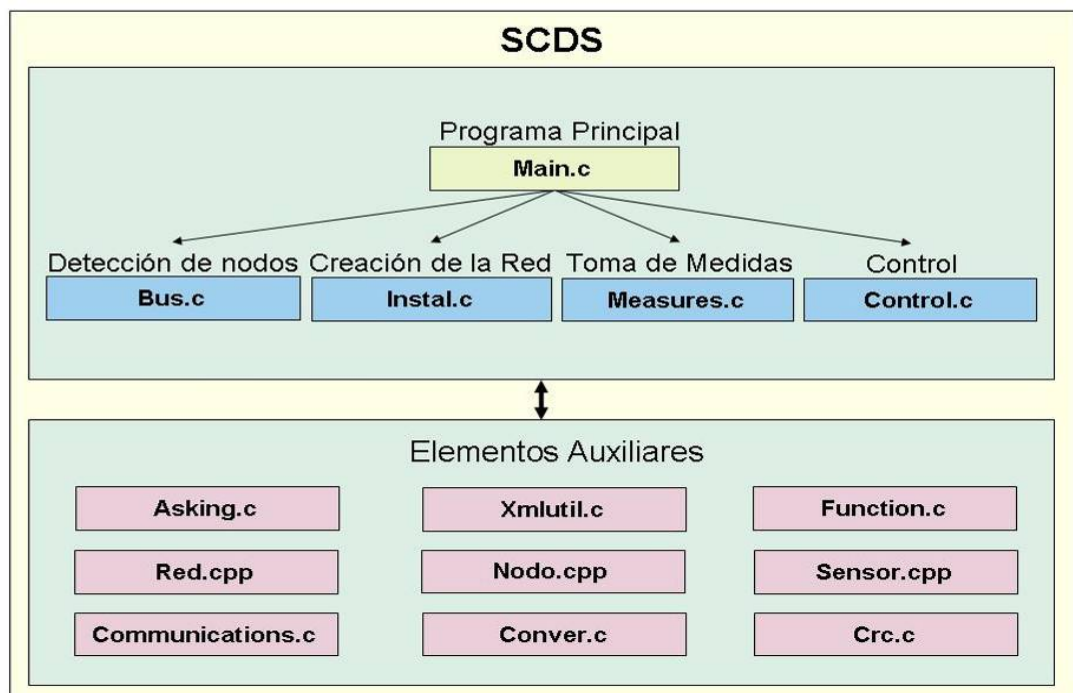


Figura 2.7 Estructura General Software de Control

- **Software de Aplicación**

El Software de Aplicación constituye la interfaz hombre-máquina del sistema. Debido a las necesidades de comunicación con el usuario se ha realizado una aplicación web con toda una estructura de formularios para las configuraciones de los nodos, sensores y los módulos de control del sistema. Esta aplicación web se ha implementado con PHP, para la visualización se ha utilizado un sistema de plantillas css y para el acceso a los datos se pueden utilizar dos fuentes distintas. Una de las fuentes serían los archivos auxiliares generados en XML para el intercambio de información. Para algunas utilidades se han utilizado varias bases de datos realizadas sobre MySQL, estas bases de datos constituyen la segunda fuente de obtención de datos para la aplicación.

Al igual que en la organización del Software de Control se consideró de máximo interés estructurar los distintos elementos del código de manera que el manejo de esta aplicación se pueda hacer de manera fácil e intuitiva. El conjunto de este bloque está integrado en la carpeta *www*. En esta carpeta se encuentran el *index.php* que será el menú principal de nuestra aplicación y *style.css* que será nuestra plantilla para dar formato a nuestra aplicación web el resto de los elementos se reparten en carpetas según su funcionalidad.

Para la configuración de la red de nodos y sensores los archivos se engloban en la carpeta *Network*. Los archivos de la configuración de los módulos de control se encuentran en *Configmodule*. Los correspondientes a la activación de los módulos de control pertenecen a *Activemodule*. La última posibilidad de acción que permite el programa principal es la visualización del estado de la red cuyos elementos están incluidos en *Visualization*.

A estas carpetas que constituyen cada uno de los bloques de la funcionalidad de la aplicación se añaden otras dos de propósito general. Una de ellas contiene las librerías necesarias para el funcionamiento de todo el

sistema (*Library*). En la otra se incluyen los elementos visuales de la aplicación web (*img*).

Puesto que el funcionamiento detallado del Sistema de Aplicación se explicará en un capítulo posterior no se procederá a analizar aquí los numerosos elementos de estas carpetas. Basta enfatizar la idea de que cada una de las funcionalidades del sistema está estructurada de manera que resulte sencillo acceder a la información que se desee en cada momento.

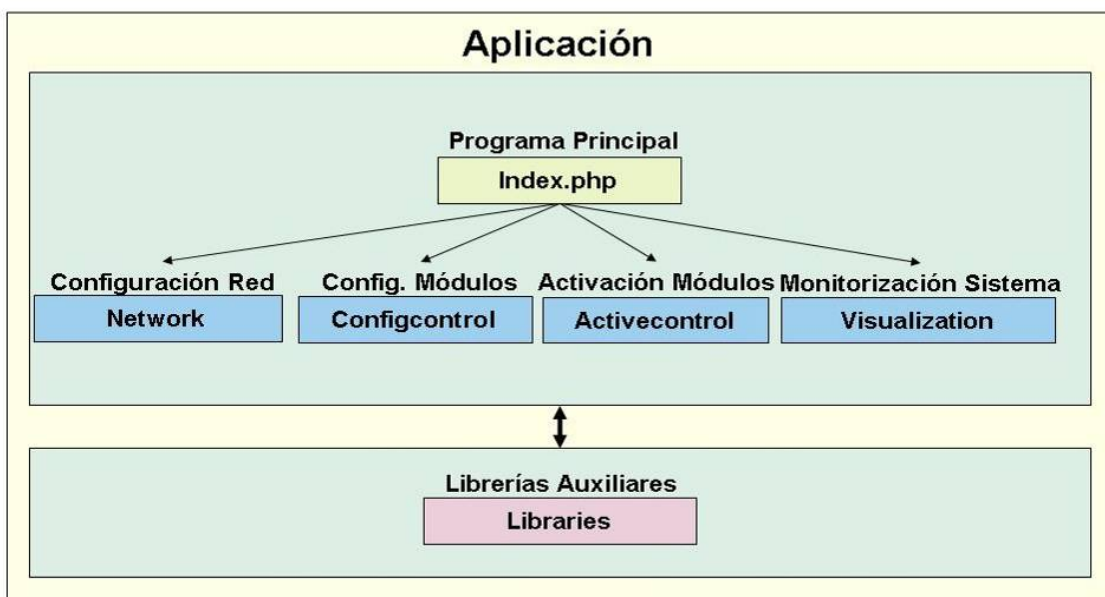


Figura 2.8 Estructura General Software de Aplicación

- **Software Auxiliar**

Para el correcto funcionamiento del sistema general y en particular para la buena comunicación entre los dos subsistemas principales del software son necesarios ciertos elementos que constituyen el software auxiliar del proyecto. Dentro de este término podemos englobar dos elementos distintos, el conjunto de archivos XML para el manejo de los datos y la información entre sistemas, y las bases de datos utilizadas.

El conjunto de archivos XML que se generan bien en el software de control, bien en el software de aplicación, se reúnen en una carpeta de nombre *xml*. Cada uno de estos elementos tiene diversas funciones.

Se utiliza *flag.xml* como manejador de eventos, dando instrucciones bien a C o a PHP de cuando pueden proseguir con cierta ejecución para instrucciones que dependen del estado de ejecución del otro sistema.

Para la configuración de la red tenemos dos elementos distintos *config.xml* contiene las direcciones de nodos disponibles para la instalación. Una vez que se ha configurado la red esta información se almacena en *instal.xml* conteniendo ya la red con la información de los nodos y sus respectivos sensores.

El almacenamiento del estado de las medidas de la red se efectúa en tres archivos distintos. Se guarda la medida actual en *instantmeasure.xml*, la medida realizada anterior a la global en *oldmeasure.xml* y se almacena un histórico de las medidas realizadas cada hora en archivos cuyos nombres llevan la información del momento en que fueron realizados. El formato de estos nombres es *pcbox año mes día hora.xml*.

Los módulos de control que se generan se van guardando en *control.xml* para que las sucesivas actualizaciones de estos módulos no creen problemas en caso de ejecutarse a la vez en C y PHP cada vez que se ejecuta la parte de control de C se vuelca el contenido de *control.xml* a *controlc.xml* con el que opera el software de control.

En cuanto a las bases de datos se utilizan en la parte de la configuración de la red. La base de datos *config.sql* guarda la información que tiene *config.xml* de los nodos detectados. La base de datos *instal.sql* tiene la información de la configuración de la red actual de la red. Los contenidos de estas dos bases de datos se comparan para la instalación o desinstalación de nodos. La

información de *instal.sql* es la que se enviará a la aplicación de control por medio de *instal.xml*.

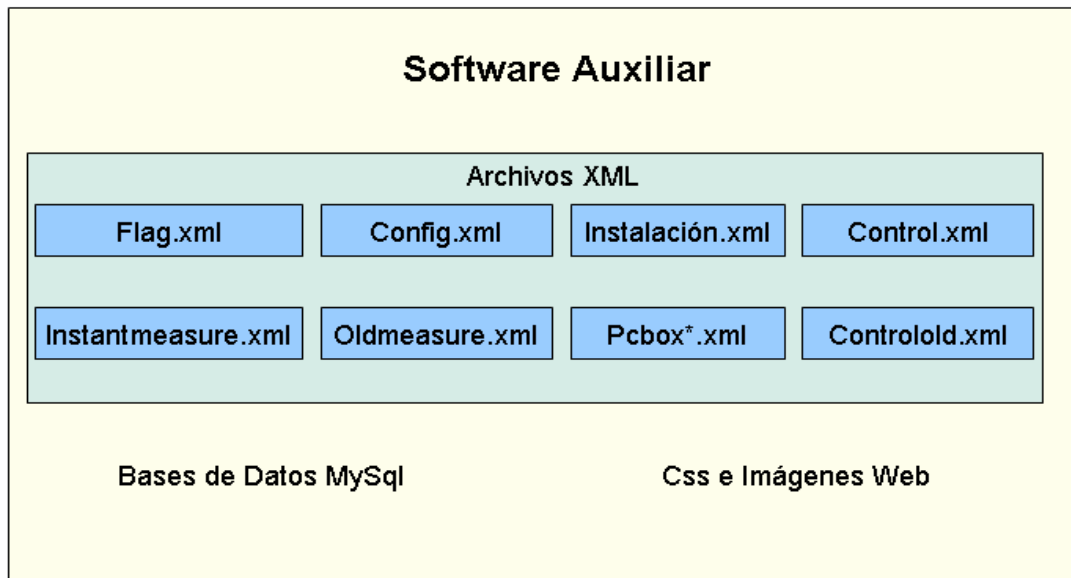


Figura 2.9 Estructura General Software Auxiliar

2.6. Conclusiones.

En este capítulo se han analizado los diferentes elementos de la arquitectura del sistema.

Se han analizado los elementos del subsistema Hardware y el protocolo de comunicaciones diseñado para la comunicación entre el hardware y el ordenador central, donde se ha optado por una arquitectura centralizada con un ordenador empujado comunicándose con el resto de elementos del sistema. Para esta arquitectura se ha implementado un sistema propietario sobre la capa física de la norma RS485, donde el bus de comunicaciones, los nodos y el protocolo de comunicaciones están diseñados con respecto a los parámetros de esta norma.

En cuanto al subsistema software se ha esbozado cual es el funcionamiento general del sistema en el apartado 2.4. Se ha indicado cómo se ha integrado la funcionalidad de un sistema de control desarrollado en C y C++ con una aplicación desarrollada en PHP de manera que el usuario pueda realizar una serie de acciones de control sobre el sistema.

Las acciones de control que permite realizar el sistema implementado son las siguientes:

- Configuración e instalación de una red de nodos y sensores.
- Configuración de módulos de control.
- Activación de módulos de control.
- Visualización del estado de medidas de los sensores.

La organización de los elementos implicados en la realización de estas acciones se ha explicado en el apartado 2.5, y el desarrollo de cada una de estas acciones específicas se detalla en los siguientes capítulos.

Capítulo 3

Software de Control

3.1. Introducción.

En este capítulo se analizará el subsistema software que constituye el motor fundamental del sistema. Este elemento se encarga de la comunicación entre el ordenador central y los elementos de la red, y es el responsable del reconocimiento de las funciones de control y el funcionamiento del sistema completo.

Se parte de un sistema de control eficiente pero diseñado a medida de una vivienda en particular, Magic Box para el concurso Solar Decathlon 2005. Dentro de la incorporación de mejoras y funcionalidades, se fija como meta conseguir un sistema de control genérico, implantable en y portable a cualquier localización, dinámico, abierto, modulable y de fácil expansión.

Manteniendo siempre la intención de la consecución con éxito de estos objetivos, la misión del software de control consiste en manejar cuatro acciones distintas que forman la funcionalidad del sistema.

- Detección de nodos.
- Construcción de una red virtual.
- Toma de medidas en cada elemento de la red.
- Interpretación y ejecución de los módulos de control.

Para realizar todas estas funciones el software de control tiene que estar plenamente integrado con el software de aplicación. La ejecución del sistema de control está gestionada por una serie de elementos escritos en lenguaje C y C++, y utiliza para su correcta comunicación con la parte de aplicación una serie de documentos XML que proporcionan un transporte de información entre ambas aplicaciones de una manera estructurada.

La interacción entre todos estos elementos permite la detección de los nodos instalados en la red en el momento en que se solicite desde la aplicación. Así

mismo, permite interpretar los componentes que forman la red física del sistema y crear una red virtual de objetos implementada en C++ que simule dicha red física real.

En caso de que exista una red configurada, el software de control realiza constantemente un seguimiento del estado de la red con una toma de medidas de cada elemento en breves espacios de tiempo. En cada uno de estos procesos se almacena la medida instantánea tomada de todos los sensores de la red, en otro documento se almacena el valor anterior al actual y por último se efectúa un registro histórico de todas las medidas realizadas en archivos separados por horas de ejecución.

En cada ejecución del programa principal del sistema de control se evalúan los módulos de control que hubiera configurados en un archivo auxiliar que contiene todos los datos del control que se desee efectuar en la vivienda. En caso de que alguno de estos módulos de control se encuentre con el estado activo se procede a efectuar una serie de procesos para garantizar la seguridad de ejecución del sistema de manera que se evite una parada del sistema.

En caso de que se certifique que la ejecución del módulo es segura, se continúa con la interpretación de los componentes del módulo de control, la evaluación de sus funciones y la ejecución de los resultados obtenidos en función de los valores de los elementos integrantes del módulo.

Los cuatro procesos descritos constituyen la funcionalidad del software de control. Con este sistema se logra satisfactoriamente el cumplimiento de los objetivos planteados, principalmente el logro de evitar la necesidad de detener y manipular el código fuente del sistema para la realización de cambios en la parametrización del control de la vivienda.

A continuación se presentarán los lenguajes de programación que se han utilizado para este subsistema software y las principales características por las

que han sido escogidos para implementar el sistema. El apartado 3.2 analizará los objetivos propuestos para el subsistema software de control. En el apartado 3.3 se definirán con mayor detalle la estructura y el funcionamiento del sistema de control, analizando las distintas acciones que forman el sistema. El análisis del sistema de control implementado finalizará con un apartado donde se ofrecen algunas conclusiones obtenidas en el desarrollo del mismo.

3.1.1. Introducción a los lenguajes de programación

Para realizar el Software de Control se necesitaba un lenguaje de programación que ofreciera buenas prestaciones. La mayor parte del código se desarrolló sobre el lenguaje ANSI C. Los orígenes del ANSI C se remontan a 1972, en los laboratorios Bell de la compañía estadounidense AT&T de la mano de Dennis Ritchie. ANSI C es un lenguaje multiuso pensado inicialmente para el desarrollo de sistemas. El uso del lenguaje C presenta una serie de ventajas [16][17]:

- Destaca por su simplicidad, concisión y potencia.
- Gran coherencia, debida en gran parte a haber sido desarrollado por una sola persona y no por un comité.
- Es un lenguaje de alto nivel y bajo nivel simultáneamente porque tiene funciones de alto nivel, pero pocas y utiliza las funciones primitivas del sistema. Casi todos los mecanismos de alto nivel son proporcionados por funciones llamadas explícitamente.
- Dispone de muchos operadores, un operador por cada una de las posibles operaciones en código máquina.
- Los compiladores de C producen códigos rápidos y eficientes.
- Los programas son fácilmente portables ya que C es independiente de cualquier arquitectura. Es decir, un programa desarrollado en un PC podrá ser ejecutado en un Macintosh o en una máquina UNIX, con mínimas modificaciones y una simple recompilación.

- Es fácilmente instalable en todo tipo de máquinas, desde grandes ordenadores a ordenadores personales.
- Usando el C, un programador puede evitar el uso de código ensamblador excepto en situaciones muy exigentes.

En el transcurso del desarrollo del proyecto se pensó en los beneficios que aportaría al sistema el uso para ciertas utilidades de un lenguaje de programación orientado a objetos. C++ ofrecía esa serie de utilidades y además una perfecta integración con el lenguaje C.

El lenguaje C++ fue creado y posteriormente constantemente ampliado por Bjarne Stroustrup. El primer compilador de C++ se desarrolló en la década de los 80, en los AT&T Bell Labs (donde se desarrolló UNIX), bajo la dirección de Stephen C. Dewhurst y Kathi T. Stark, para utilizarse en proyectos de programación de sistemas de conmutación [20].

Originalmente se utilizaba un precompilador de C++, que convertía ficheros .cpp, .c, o .cc a ficheros de C, posteriormente compilados y enlazados con herramientas para C. Este precompilador se encuentra actualmente integrado en el proceso de compilación. El hecho de que C++ se traduzca antes a C da una idea de la enorme similitud entre ambos. En principio, casi cualquier programa escrito en C, con algunas excepciones, puede ser compilado por un compilador de C++. El mismo programa, en un fichero con extensión *.c puede ser convertido en un programa en C++ cambiando la extensión a *.cpp.

Algunas de las mejoras de C++ frente a ANSI C son [19] [20] :

- Agilizar la creación y el manejo de tipos de datos abstractos (creados por el programador).
- Introducir, aparte de las estructuras de C, estructuras ampliadas (**clases**) que no sólo aglutinan datos, sino que también incorporan las funciones que los manejan.
- Lograr que la gestión de memoria resulte más sencilla y transparente.

- Reducir errores de programación mediante un control de tipos más estricto, que asume como errores de compilación desajustes de tipos que en C sólo daban lugar a *warnings*.
- Permitir una mayor reutilización del código existente. Aparte de las conocidas librerías de C, base de la reutilización de código, C++ introduce sintaxis específica para este mismo fin (mecanismos de herencia).

En concreto quizá el mayor logro reside en la aportación de sintaxis específica para la generación de e interacción entre estructuras abstractas de datos, que siguen una programación no orientada a función (Procedural Programming) sino a objetos (Object Oriented Programming).

Toda clase de C++ tiene un constructor y un destructor para el manejo de los objetos. Ambos tienen el mismo nombre que la clase. En C++ los nombres de las funciones se pueden sobrecargar (el destructor es el único que no puede ser sobrecargado), ésta característica permite, como veremos más adelante, tener distintos constructores con distintos parámetros de entrada.

Para dotar de dinamismo al Sistema de Control diseñado, es necesario proporcionarle desde documentos externos los parámetros que regirán la funcionalidad del control. De esta manera al estar los parámetros de control en documentos externos no es necesario paralizar el sistema para modificarlos.

Para proporcionar al sistema la información de control de una manera estructurada, almacenar los datos resultantes de las medidas y servir de interfaz entre el subsistema de control y el subsistema de aplicación, se eligió el formato del Lenguaje de Marcado Extensible (XML-Extensible Markup Language).

En 1996, [21] el World Wide Web Consortium (W3C) comenzó a desarrollar un nuevo lenguaje de marcado estándar que fuera más sencillo de utilizar que SGML, pero con una estructura más rígida que la de HTML. Se enumeran a

continuación algunos de los objetivos de XML tal y como aparecen en la versión 1.0 de su especificación [22]:

- XML debe admitir una gran variedad de aplicaciones.
- Debe ser sencillo escribir programas para procesar documentos XML.
- Los documentos XML deben ser legibles para las personas y razonablemente claros.
- El diseño de XML debe estar preparado rápidamente.
- El diseño de XML debe ser formal y conciso.

La elección de XML proporciona una serie de ventajas frente a otros lenguajes de marcado como SGML y HTML [22] :

- XML es internacional. Basado en Unicote. Mientras que SGML y HTML están basados en ASCII que no funciona muy bien con algunos idiomas.
- XML puede estar estructurado. Se puede utilizar DTD para estructurar XML de manera que tanto el contenido como la sintaxis se puedan validar de forma sencilla y así crear documentos estándar válidos.
- XML puede ser un contenedor de datos. XML permite representar eficientemente casi cualquier dato de forma que pueda ser leído por personas, analizadores sintácticos computerizados y aplicaciones.
- XML proporciona flexibilidad. Con una DTD se puede definir la estructura exacta del documento para que éste pueda validarse, pero su uso no es estrictamente necesario para estructurar un documento.
- XML es sencillo de usar. Sólo es un poco más complicado que HTML.
- XML posee formatos estándar. Se pueden generar formatos estándar para los documentos XML de forma sencilla.

XML se convierte por tanto en una herramienta ideal para la transacción concisa y estructurada de información entre los dos subsistemas que componen la aplicación de control desarrollada.

3.2. Objetivos.

La principal característica deseable para un sistema de control, es que pueda adecuarse a las necesidades del usuario. Cuando se quiere establecer un control, se espera poder establecer las normas que regirán el correcto funcionamiento del sistema. Esta característica implica la necesidad de dotar al sistema de control de un dinamismo que permita modificar su estructura con facilidad.

En este proyecto se ha desarrollado un sistema de control genérico, abierto, modulable y de fácil expansión. Este punto de vista difiere de la intencionalidad con la que se diseñó la aplicación de control original. El propósito original era implementar un sistema de control que dirigiera y rigiera el funcionamiento de Magic Box. El sistema desarrollado cumplía su labor a la perfección incluyendo además la precisión que requerían las instrucciones del concurso *Solar Decathlon*. Sin embargo, considerando la funcionalidad deseada para este proyecto la implementación original tenía dos inconvenientes:

- El control original sólo servía para la distribución de Magic Box. En caso de intentar extrapolar ese sistema de control a otro lugar con distinta distribución de los elementos de la red de sensorización sería necesario realizar un código nuevo en función de esta nueva distribución.
- El segundo inconveniente para realizar un sistema de control genérico era su estatismo. Cada valor del sistema de control estaba fijado en el código. Esto implica que cada modificación que se quisiera realizar en el control de la vivienda requería la paralización del sistema completo, la modificación del código necesario y la compilación y puesta en marcha del nuevo código.

Teniendo en cuenta que la realización de este proyecto forma parte de una labor de investigación que comenzó con la implementación de Magic Box y que continuará posteriormente a la conclusión de este proyecto, y el volumen de la codificación elaborada. Sería de gran incomodidad que un nuevo investigador

tuviera que analizar todo el código elaborado anteriormente para realizar nuevos cambios en el sistema.

Por todo ello el principal objetivo del software de control de este proyecto es conseguir un sistema de control **genérico** lo más **abierto**, **dinámico** y **modulable** posible. Para una mejor comprensión de los objetivos propuestos se detalla a continuación cómo se ha enfocado cada uno de ellos:

- Genérico. Se desea que el software desarrollado se pueda aplicar en todo tipo de entornos con distintas características de la red de sensorización.
- Abierto. Que no dependa del tipo de hardware utilizado, ni del sistema operativo. Con el único condicionamiento de escalar la red en función de los nuevos elementos que se introduzcan.
- Dinámico. Que el usuario no tenga restricciones a la hora de modificar el sistema de control. Que se puedan configurar los elementos de la red de sensorización o los módulos de control del sistema sin que sea necesario paralizar la ejecución del sistema para realizar estas modificaciones. El condicionamiento en este caso es la necesidad de incluir en el código funciones nuevas que se deseen utilizar.
- Modulable. Se ha intentado estructurar de la mejor forma posible cada elemento del Software de Control de manera que se facilite a nuevos investigadores el manejo y la expansión de esta aplicación.

3.3. Estructura.

A continuación se pasará a analizar la forma en que se ha diseñado este subsistema Software. Se observarán los archivos que componen este software y las principales funciones que lo forman.

3.3.1. Gestión del Sistema.

Como ya se ha expresado anteriormente esta parte del software constituye el motor principal del sistema, esta función principal se encuentra en el archivo `main.c`

Main.c

Al comenzar el programa, se crea el objeto de la clase CRed que constituirá la red virtual del sistema, abre el puerto de comunicaciones que se utilizará para el paso de información entre el programa principal y el hardware, y crea el archivo flag.xml que será el manejador de eventos que enlazará las acciones del programa de control en C y la aplicación en PHP. Para una mejor comprensión del capítulo se presentan los contenidos de este archivo auxiliar.

```
<?xml version="1.0" encoding="ISO-8859-15" ?>
```

```
<NodeList version="1.0"
```

```
flagbus="0" flagconfig="0" flaginstal="0" flagcontrol="0" />
```

A continuación el sistema realiza secuencialmente las siguientes acciones:

- Detección de nodos.
- Instalación de la red.
- Toma de medidas.
- Control.

La ejecución de estas tareas se produce al cumplirse ciertas condiciones.

ACCIÓN	CONDICIÓN	EJECUCIÓN
Detección de Nodos	flagbus=1	Bus
Creación de Red	(flaginstal=1) o (instalación.xml y Red vacía)	Get_file
Toma de Medidas	instalación.xml	Asking
Control	control.xml	Controlling

Tabla 3.1 Gestión del Sistema

En la Figura 3.1 se muestra un esquema del funcionamiento de la Gestión del Sistema.

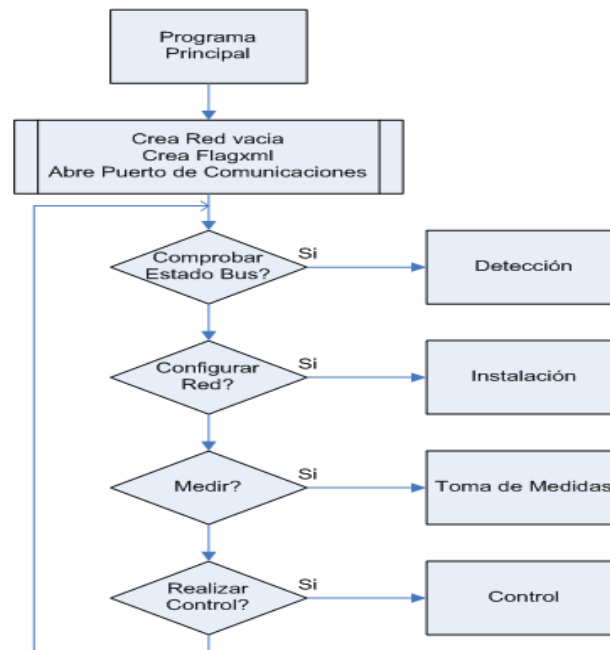


Figura 3.1 Diagrama de Flujo Programa Principal

La tarea de detección responde a una petición procedente de la aplicación en PHP, esta petición se detecta con el archivo de intercambio flag.xml. Esto se produce cada vez que se desea modificar la configuración de la red.

Para la ejecución de la tarea de instalación se pueden dar dos casos. El primero consiste en una alteración de la configuración de la red por el usuario. Donde se indica la petición de una actualización de la red virtual a través de nuevo del archivo de intercambio. Al detectarse la actualización comienza la ejecución de la instalación. La otra opción es un caso en el que no se modifique nada desde la aplicación, sino que se detenga el sistema de control y se reinicie. En ese caso se comprobaría que el objeto de la red virtual esta vacío y que existe la estructura de la red, en *instalación.xml*. Si se cumplieran ambas condiciones se procedería también a la creación de la red virtual.

La tarea de toma de medidas se realiza siempre que exista la estructura de red, *instalación.xml*. De la misma manera la tarea de control se realiza siempre que existan módulos de control creados, almacenados en *control.xml*.

Este bucle es el núcleo del sistema y se realiza constantemente lanzando las operaciones necesarias. A continuación se profundiza en cada una de estas tareas.

3.3.2. Detección de Nodos.

La hebra encargada de la detección de los nodos (*void *bus(void *arg)*) disponibles para la red se encuentra en el archivo *bus.c*.

Al ejecutarse esta tarea, se crea el archivo auxiliar *config.xml* que será el que contenga la información de los nodos disponibles, y posteriormente realiza un barrido del número de nodos máximo fijado para la red. En principio para este proyecto se ha fijado el máximo número de nodos en doce para reducir el tiempo de espera en las ejecuciones pensando en un dimensionamiento moderado de nuestra red. El valor del número máximo de nodos en la red se almacena en *sdsp/lib/includes.h*.

Siguiendo con la ejecución se lanza una pregunta a cada una de estas direcciones de red para comprobar si se encuentran activas. Estas preguntas (*AskVivo*) y las respuestas (*LectureVivo*) se codifican en el archivo *asking.c* y serán analizadas más adelante. Las direcciones de los nodos que han respondido se almacenan en un array y son procesadas después en una función de este mismo archivo llamada *ConfigNodo*.

ConfigNodo se encarga de dar formato al archivo *config.xml* generado al principio con las direcciones de los nodos detectados de manera que luego puedan ser procesados por la parte de aplicación. A continuación se muestra un ejemplo de la estructura de este archivo con la detección de los nodos 1 y 7.

```
<?xml version="1.0" encoding="ISO-8859-15" ?>
```

```
<NodeList version="1.0">
```

```
<node address="1" />
```

```
<node address="7" />
```

```
</NodeList>
```

Una vez se ha finalizado la generación de este archivo se modifican los valores de flagbus a cero y flagconfig a uno para avisar a la aplicación de que puede proceder a la instalación o desinstalación de los nodos. (Ver Figura 3.2).

3.3.3. Creación de la Red.

Una vez que se dispone del archivo *instalación.xml* con la configuración de la red que se ha configurado se procede a la creación de la red virtual, por medio de la hebra (*void *get_file(void*filename)*), codificada en el archivo *instal.c*

Esta tarea se encarga de procesar la creación de la red virtual. Cada vez que se ejecuta la hebra de instalación se borran los elementos de la red antigua mediante la función *ResetRed*. A continuación recorre los elementos del archivo *instalación.xml* que contiene la información de la red enviada por la aplicación. Cada vez que se encuentra con un elemento de tipo nodo mediante la función (*parse_node*) se procesa la información que contenga cada nodo.

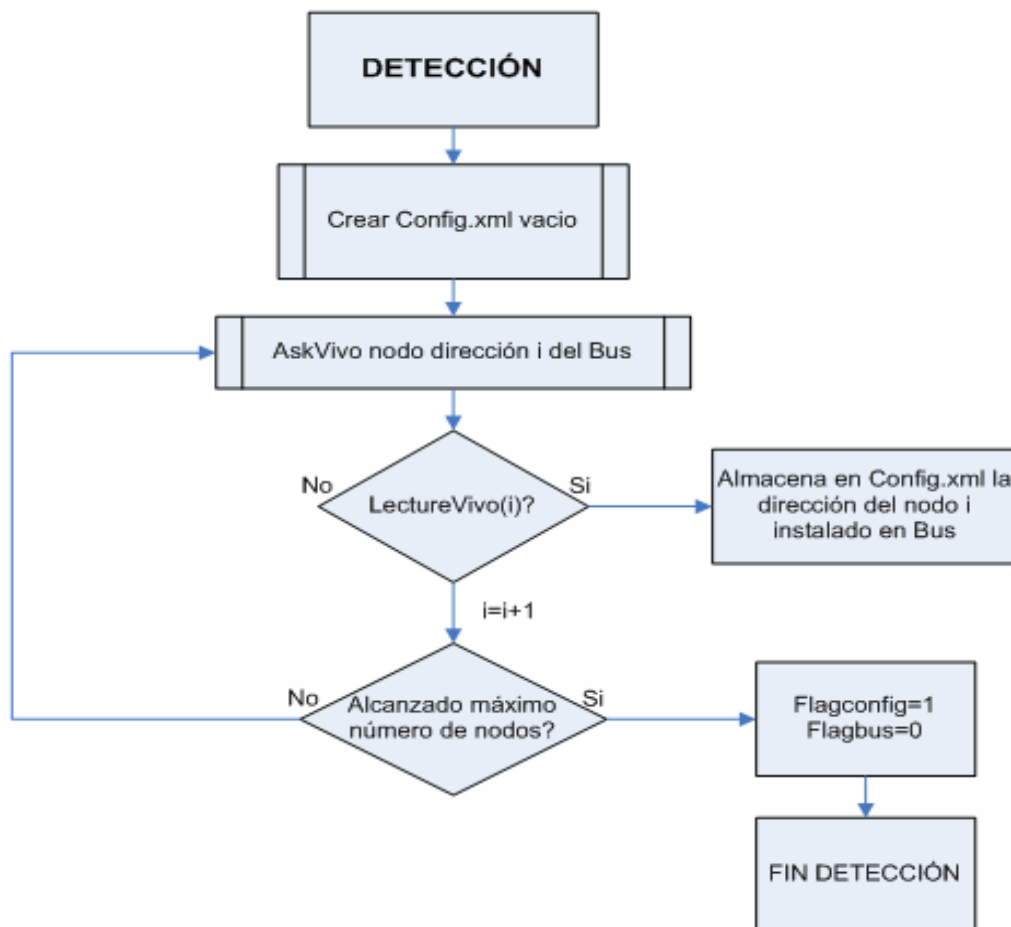


Figura 3.2 Diagrama de Flujo tarea Detección

Parse_node examina los atributos del nodo del archivo XML y se crea un objeto de tipo CNode con esos atributos. Seguidamente recorre cada uno de los sensores pertenecientes a ese nodo y nuevamente coge sus atributos del XML y crea un objeto tipo CSensor con esos valores. Una vez es creado el objeto tipo CSensor se fija como elemento de su nodo propietario y prosigue con la exploración del XML hasta finalizar con los sensores de ese nodo.

Cuando se ha acabado de procesar la información del nodo, éste se fija a la red virtual y se procede a realizar la primera medida de los sensores para eliminar el valor original con el que se crean. Para ello se llama a la función AskNode cuyo funcionamiento se detallará posteriormente (Asking.c).

Tras ejecutar esta función el nodo se haya por primera vez instalado en la red virtual y con los valores de sus sensores obtenidos después del primer proceso de medida, con lo cual se procede a crear los primeros archivos de datos, manejados por la función *AddNodo*.

AddNodo comienza tomando los valores de cada uno de los atributos del nodo virtual. A continuación se localiza el archivo de datos histórico (*pcboxañomesdiahora.xml*) y en caso de no existir se crea. Los datos del nodo se introducen en una estructura de datos XML y se llama a una nueva función *SensorsXML* para que procese los datos de los sensores.

Cuando se procesan los datos de un sensor se hace una doble llamada con diferentes parámetros a la función *AddMeasure* que analizaremos con el archivo *measures.c* y que introducirá los datos estructurados correctamente en los dos archivos de datos, el histórico (*pcboxañomesdiahora.xml*) y el que almacena la medida instantánea (*instantmeasure.xml*).

La tarea de instalación acabará cuando la información de todos los nodos y sus sensores con sus correspondientes medidas esté representada en los mencionados archivos auxiliares. A continuación se muestra un ejemplo para cada archivo de medidas.

Pcboxañomesdiahora.xml

```
<?xml version="1.0" encoding="ISO-8859-15" ?>

<NodeList version="1.0">

<node address="1" status="1" zone="1" nsensors="3">

<sensor sensorid="1" tipo="1">

<measure hour="16:33:17" value="29.5" />
```

```
<measure hour="16:35:06" value="29" />

</sensor>

<sensor sensorid="2" tipo="2">

    <measure hour="16:33:17" value="17.2" />

    <measure hour="16:35:06" value="16" />

</sensor>

<sensor sensorid="3" tipo="4">

    <measure hour="16:33:17" value="1" />

    <measure hour="16:35:06" value="1" />

</sensor>

</node>

<node address="7" status="1" zone="1" nsensors="3">

    <sensor sensorid="1" tipo="1">

        <measure hour="16:33:17" value="28.3" />

        <measure hour="16:35:06" value="30" />

    </sensor>

    <sensor sensorid="2" tipo="2">

        <measure hour="16:33:17" value="13" />

        <measure hour="16:35:06" value="13" />

    </sensor>

    <sensor sensorid="3" tipo="3">

        <measure hour="16:33:17" value="1" />

        <measure hour="16:35:06" value="1" />

    </sensor>

</node>
```

```
</sensor>

<sensor sensorid="3" tipo="4">

    <measure hour="16:33:17" value="1" />

    <measure hour="16:35:06" value="1" />

</sensor>

</node>
```

Instantmeasure.xml

```
<?xml version="1.0" encoding="ISO-8859-15" ?>

<NodeList>

<node address="1" status="1" zona="1" nsensores="2">

<sensor sensorid="1" tipo="1" medida="28.5" hour="19:44:49" />

<sensor sensorid="2" tipo="2" medida="20" hour="19:44:49" />

</node>

<node address="3" status="1" zona="2" nsensores="2">

<sensor sensorid="1" tipo="1" medida="28.2" hour="19:44:59" />

<sensor sensorid="2" tipo="2" medida="38.2" hour="19:44:59" />

</node>

</NodeList>
```

En la figura 3.3 se muestra un esquema del funcionamiento de la tarea de Instalación del Sistema.

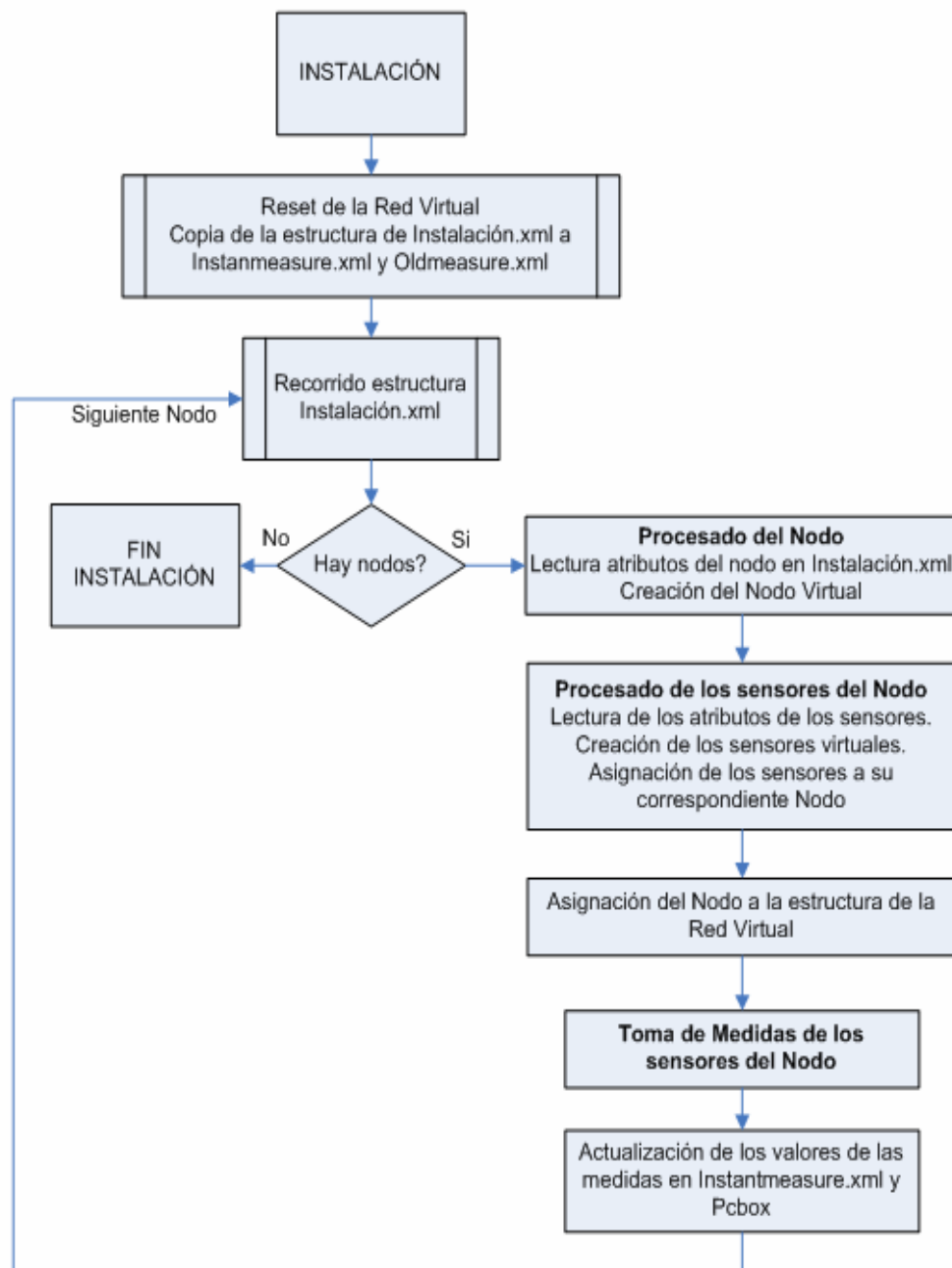


Figura 3.3 Diagrama de Flujo tarea Instalación

3.3.4. Toma de medidas.

Siguiendo con el repaso de cada una de las tareas lanzadas por el programa principal, *Measures.c* contiene la hebra (*void *asking (void *arg)*) encargada de realizar todas las medidas posteriores a la primera medida tomada en el momento de la instalación.

Esta tarea se realiza en cada ciclo de ejecución del programa principal siempre que la red esté configurada. Al comenzar la tarea de medida se guarda la medida instantánea en *oldmeasure.xml* y después se recorren los nodos de la red virtual almacenando las medidas de los sensores en el archivo de medida instantánea y en el histórico de medidas. Para ello se utiliza el mismo mecanismo seguido en la tarea de instalación, mediante las funciones de *AskNode* para proceder a la toma de medidas y *AddMeasure* para almacenar los datos de los sensores en sus correspondientes registros.

Se puede apreciar el funcionamiento de la tarea de Toma de Medidas en la Figura 3.4

3.3.5. Control.

La última tarea del sistema de control (*void *controlling (void *filename)*) es la encargada de reconocer las instrucciones de los módulos de control, realizar los cálculos oportunos y ejecutar las acciones correspondientes. Su ejecución se realiza en el archivo *control.c*. (Ver Figura 3.5).

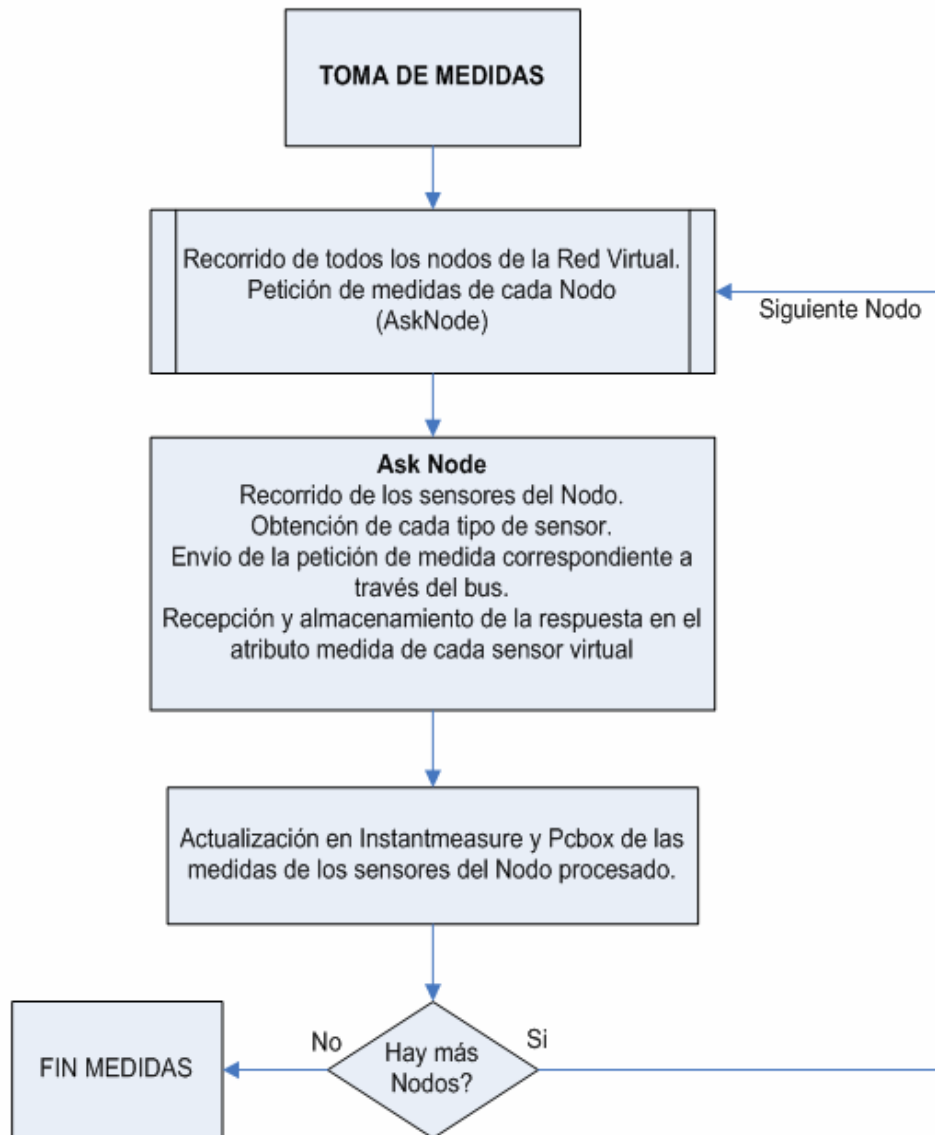


Figura 3.4 Diagrama de Flujo tarea Toma de Medidas

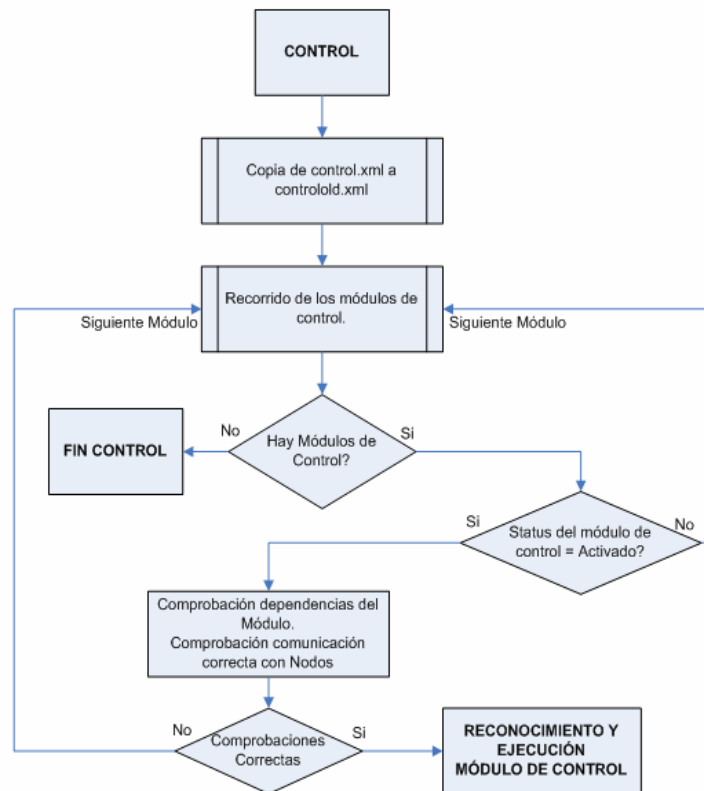


Figura 3.5 Diagrama de Flujo tarea Control

En la parte de aplicación los cambios que se realizan en la configuración de los módulos de control se almacenan en el archivo *control.xml*. El continuo movimiento de información hace particularmente tedioso el trabajo con dicho archivo de intercambio. Es por ello que al comenzar la tarea se crea una copia del contenido del archivo *control.xml* a modo de semáforo. De esta manera se evitan problemas de exclusión mutua en ejecución. Estos problemas se podrían producir además por los motivos siguientes:

- En primer lugar es necesario evitar que se proceda a la ejecución de un módulo de control si este está incompleto, algo que puede ocurrir cuando la configuración del módulo no ha finalizado. Para esto cada módulo posee un atributo llamado *status* que se inicializa a cero cuando procedemos a la instalación. Una vez finalizada la configuración el usuario puede poner el

status a uno para que ése módulo entre en la tarea de ejecución. Por lo tanto la hebra de control después de actualizar el archivo de control con el que va a trabajar va detectando los módulos que tiene instalados y pasa a procesar aquellos que tienen el atributo *status* a uno. De este procesado se encarga la función *getfunctionxml*.

- El segundo problema que podría causar errores de ejecución sería intentar trabajar con un nodo que estuviera estropeado o desconectado. Como medida de seguridad cada módulo posee un atributo llamado *dependencias*, en su interior se almacenan las direcciones de los nodos que son necesarios para la ejecución del módulo de control. Para evitar errores se comprueba que los nodos necesarios están instalados y no tienen problemas de comunicación.

Getfunctionxml es una de las funciones más importantes de las implementadas, ya que es la que aporta la capacidad del reconocimiento de cualquier función de control y su ejecución. Para comprender mejor su funcionamiento se explicará primero un modelo de lo que podría ser una función completa de control.

Un módulo de control consta de una condición principal y un bloque de acciones que llevar a cabo según el valor de esa condición. Una condición tipo podría ser como la siguiente.

“Si la media de los sensores de temperatura de los nodos 1 y 3 es mayor que 31 y el valor de la temperatura del nodo 5 es mayor que la temperatura del mismo sensor en la toma anterior...”

El modelo formalizado de la condición corresponde con la ecuación:

$$\text{If } ((\text{media } (T1, T3) > 31) \text{ and } (T5(t) > T5(t-1)))$$

Operación comparador operación link operación comparador operación

Estos serían los elementos de una condición donde habría cuatro tipos principales de operaciones:

- Valor de un sensor
- Función media de n número de sensores
- Valor numérico
- Valor de un sensor en la medida anterior

Se puede observar que una condición simple siempre va a tener una misma estructura “operación – comparador – operación”. Se pueden anidar varias condiciones simples con linkers o nexos de tipo *and* o tipo *or*. También se pueden generar niveles de anidamientos de condiciones con condicionales que contienen otros condicionales.

Según el cumplimiento de esta condición, las acciones a realizar podrían ser de este tipo.

“...Si la condición es cierta activar el relay del nodo 1 y si además el relay del nodo 5 está desconectado activarlo.

Si la condición es falsa desactivar el relay del nodo 3.”

Actions if true

- ON R1
- If R5= OFF then ON R5
- Action if false
- OFF R3

Se pueden configurar las respuestas para una resolución positiva de la condición y una resolución negativa, anidando los ON y OFF necesarios y pudiendo incluirse también condiciones dentro de la misma respuesta.

A continuación se muestra la estructura del módulo de control interpretada por la función `getfunctionxml`.

```
<module name="proyecto" status="0" description="If ((media (T1,T3)>31)
and (T5>T'5)) true->ON R1, If R5= OFF then ON R5,false->OFF R3">
```

```
<dependences>
```

```
<node value="1" />
```

```
<node value="3" />
```

```
<node value="5" />
```

```
</dependences>
```

```
<operation type="average">
```

```
<sensor sensorid="1" tipo="1" nodoid="1" />
```

```
<sensor sensorid="1" tipo="1" nodoid="3" />
```

```
</operation>
```

```
<comparator type="5" />
```

```
<operation type="valor">
```

```
<limit value="31" />
```

```
</operation>
```

```
<linker type="1" />
```

```
<operation type="sensor">
```

```
<sensor sensorid="1" tipo="1" nodoid="5" />
```

```
</operation>

<comparator type="5" />

<operation type="sensorT-1">

  <sensor sensorid="1" tipo="1" nodoid="5" />

</operation>

<answer type="1" id="0">

<sensor answer="1" sensorid="2" tipo="4" nodoid="1" />

</answer>

<answer type="1" id="1">

<if answer="3" />

<operation type="sensor">

  <sensor sensorid="1" tipo="4" nodoid="5" />

</operation>

<comparator type="1" />

<operation type="valor">

  <limit value="0" />

</operation>

<answer type="1" id="1">

  <sensor answer="1" sensorid="1" tipo="4" nodoid="3" />
```

```
</answer>

</answer>

<answer type="2" id="0">

  <sensor answer="2" sensorid="1" tipo="4" nodoid="3" />

</answer>

</module>
```

Se puede observar cómo cada elemento se estructura en una etiqueta para su posterior interpretación. Estas etiquetas nos diferencian los diferentes componentes del módulo:

- Operaciones,
- Comparadores,
- Nexos,
- Respuestas
- Nuevas condiciones.

Una vez que estos términos ya son familiares para el lector será más sencillo comprender el funcionamiento de la interpretación de estas órdenes y su ejecución.

La función comienza a leer la estructura en árbol del documento *control.xml*. El primer elemento lo constituyen las dependencias. Las direcciones de los nodos necesarios para la ejecución del módulo se pasan como parámetros a una función de la clase *CRed* que analiza si todos los nodos necesarios están disponibles en la red virtual, ya que los valores de las medidas se obtienen de estos elementos.

De manera adicional no sólo se comprueba si dichos nodos están en la red virtual sino que además se comprueba a través de un *AskVivo* si la comunicación con cada uno de los nodos es correcta, por si algún nodo hubiera dejado de funcionar. En caso de que la respuesta de un nodo a alguna de estas preguntas fuera negativa no se ejecutaría el módulo de control.

En caso de que todas las comprobaciones sean correctas se procede al análisis de la orden de control. Para este análisis se utilizan dos variables (*comparable* y *linkable*) para saber si la estructura de la orden se encuentra dentro de una comparación o de un linkado. El proceso sería el siguiente:

La función va recorriendo los elementos del nodo raíz del documento. De estos elementos se dirá que pertenecen al nivel cero de indexación.

Después de las dependencias el primer elemento sería una operación. Se analiza cuál es el tipo de la operación. En caso de ser un sensor (t) o un sensor (t-1) (valor del sensor de la toma de medidas anterior a la actual) se procesan los datos del sensor en una función *getvalue* que devuelve el valor del sensor pedido.

Si se trata de una función, se procesa su información en el código correspondiente del archivo *función.c* donde se encuentran los desarrollos de todas las funciones. En caso de que la operación sea propiamente un valor, éste se extrae del mismo documento. Por lo tanto independientemente de la operación al final de su procesado siempre se obtiene un valor numérico.

A continuación se observa la variable auxiliar que indica si el análisis de la condición se encuentra detrás de un comparador. En caso de que esto fuera cierto se realizaría la comparación correspondiente entre el valor recién obtenido y el valor anterior a la comparación, y a continuación se observaría también si se encuentra detrás de un proceso de linkado procediendo a operar los resultados de las condiciones simples correspondientes, y todo este

proceso finalizaría desactivando los flags de linkado y comparación. En caso de que fuera una operación antes de un comparador el programa continúa analizando la estructura de control.

El siguiente elemento es un comparador, por lo que se activa el flag de comparación y se almacena el valor antiguo de la operación en una variable de valor a comparar.

De esta manera después de cada bloque operación – comparador – operación se obtiene una variable que a modo de un booleano nos indica si la comparación es cierta o es falsa en el nivel de indexación que estemos. La condición de control podría haber finalizado ya en caso de ser simple, o podría tener uno o varios linkers ampliando la condición general. De modo similar a la actuación en la comparación en caso de que haya un link se almacena el valor booleano de la condición cierta o falsa en una variable auxiliar y se activa el flag de linkado. Como ya se ha indicado previamente la operación del linkado se evalúa tras finalizar los bloques de condiciones simples. Así que al final del análisis de todos estos elementos se obtiene siempre una variable con un valor de condición general cierta o falsa.

El análisis del módulo de control continuaría con las respuestas. Se realizan sólo las respuestas que se cumplen, esto es, respuestas positivas de condición positiva y respuestas negativas de condición negativa. Puede haber dos tipos de acción a realizar, si la respuesta es un relay se efectúa el ON o el OFF sobre ese relay. Para ello se emplean las funciones *PutRelay* y *LectureVivo* para mandar la orden al hardware para activar o desactivar el actuador, y *FijaRelay* para actualizar el valor del relay en el archivo de medida instantánea.

Además de la acción sobre un relay, la propia respuesta puede requerir la comprobación de nuevas condiciones y dentro de estas condiciones se pueden incluir otras más. Éste factor es el que nos determina el llamado nivel de

indexación. La condición general corresponde al nivel de indexación cero, en caso de que la respuesta a la condición general sea una nueva condición sus elementos se guardan con nivel de indexación uno, y así sucesivamente. Esto nos posibilita incluir acciones sobre relays para cada uno de los niveles. En caso de que tengamos un nuevo condicional se llama de nuevo a la función *getfunction* introduciendo como uno de sus parámetros el nivel nuevo de indexación.

La Figura 3.6 muestra un esquema del algoritmo para el reconocimiento y ejecución de los módulos de control.

Con control se termina el repaso de las cuatro tareas principales del subsistema de control. El resto de componentes auxiliares se explican en el apartado siguiente.

3.3.6. Elementos Auxiliares.

A continuación se resaltan las funcionalidades del resto de sistemas que complementan el software de control.

Asking.c

Asking.c resulta un archivo fundamental en todo el sistema ya que en su contenido se encuentran las funciones encargadas del flujo de información con los nodos y los sensores. Podemos destacar distintas funciones dentro de este archivo.

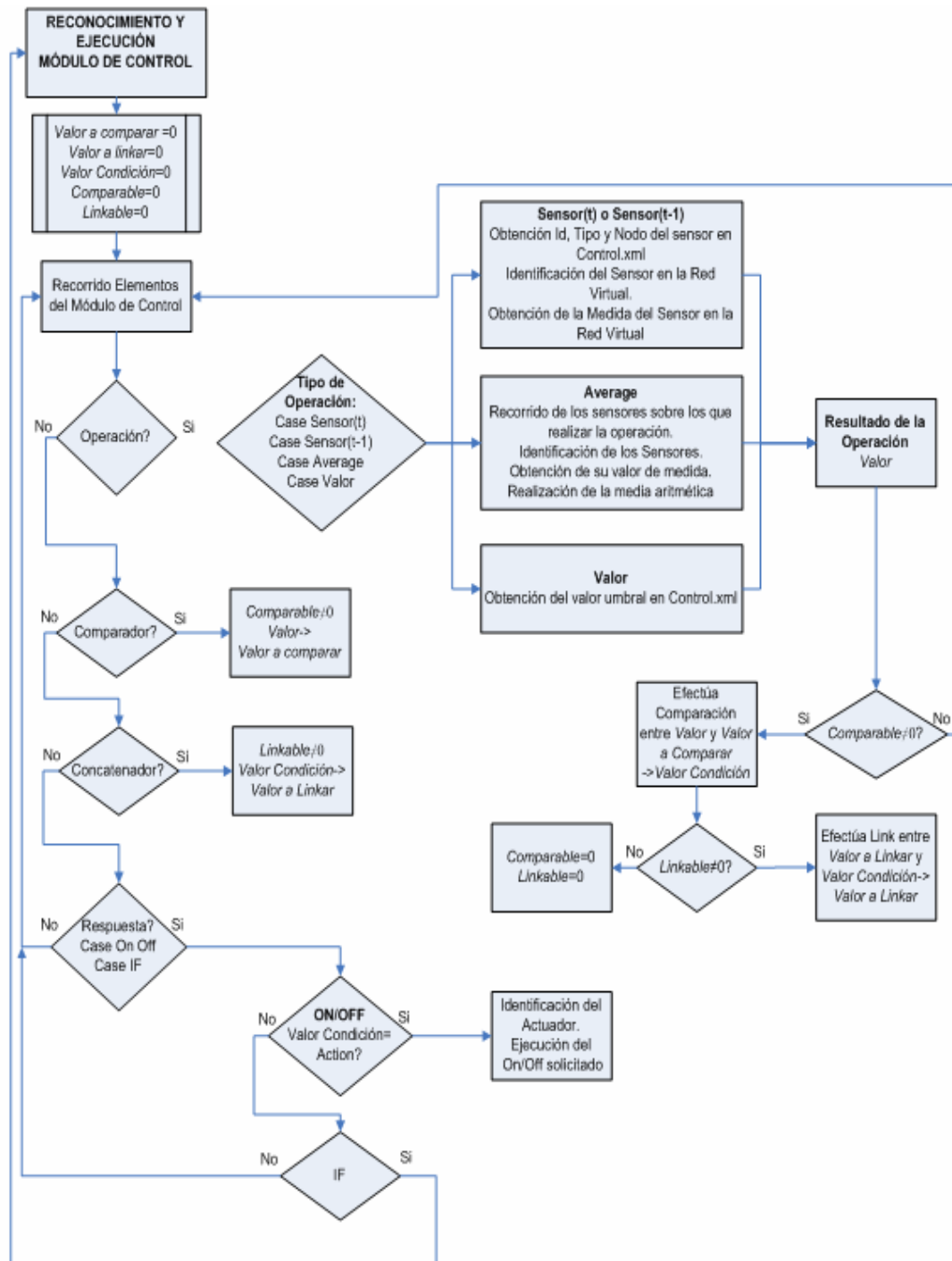


Figura 3.6 Esquema Reconocimiento y Ejecución Control

FUNCIÓN	PARÁMETROS	DESCRIPCIÓN
void AskNode	(unsigned char *num_frame, int fd_serial, CNode &node_asked)	Recorre los sensores del nodo para proceder a realizar la medida
void AskTemperature	(char address, unsigned char *num_frame, int fd_serial)	Envía al sensor de dirección address una petición de medida de temperatura
void AskHumidity	(char address, unsigned char *num_frame, int fd_serial)	Envía al sensor de dirección address una petición de medida de humedad
void AskLight	(char address, unsigned char *num_frame, int fd_serial)	Envía al sensor de dirección address una petición de medida de luz
void PutRelay	(char address, unsigned char *num_frame, char mode, int fd_serial)	Posiciona el estado del actuador en función del parámetro mode
void AskVivo	(char address, unsigned char *num_frame, int fd_serial)	Envía al sensor de dirección address una pregunta para comprobar si el nodo responde
char Lecture	(int value, CSensor &sensor)	Lee el valor enviado por el sensor y lo incluye en el atributo medida del sensor virtual
char Lecturevivo	(int value)	Lee el valor ACK enviado por el sensor

Tabla 3.2 Tabla Resumen Funciones Comunicación Bus

La función *void AskNode (unsigned char *num_frame, int fd_serial, CNode &node_asked)* recorre los sensores del nodo que se le pasa como parámetro y en función del tipo de sensor lanza la correspondiente función para realizar el proceso de medida. En los casos en que el sensor sea de temperatura, humedad o luz el tratamiento es idéntico salvo valores numéricos.

Las funciones *void AskTemperature (char address, unsigned char *num_frame, int fd_serial)*, *void AskHumidity (char address, unsigned char *num_frame, int fd_serial)* y *void AskLight (char address, unsigned char *num_frame, int fd_serial)* tienen una base muy similar, ya que en ellas se genera el mensaje de petición de medida siguiendo el protocolo de comunicaciones explicado en el Capítulo 2.

Los resultados de estas medidas se recogen en la función *Lecture*, que recibe y descifra el mensaje devuelto por el nodo y opera en consecuencia

según el tipo de la medida recibida. El valor de la medida se almacena en el sensor de la red virtual por medio de la función de la clase *CSensor FijaMeasure*. Si la trama se ha recibido con éxito se envía un uno que se almacena como status del nodo para indicar que todo es correcto, en caso de que la conexión con el nodo falle se envía un 0.

En el caso en que el sensor sea de tipo relay el procedimiento es distinto ya que los relays no tienen capacidad de comunicar cual es su estado. Por ese motivo es necesario registrar cada alteración de su estado partiendo de que al comienzo de la ejecución los relays siempre están activos. En este caso para conocer el estado del relay se realiza la siguiente operación: en primer lugar se realiza un *AskVivo* al nodo al que pertenece el relay, seguido de un *LectureVivo* fijando el status del nodo de igual forma que se realizaba en *Lectura*, para comprobar que está activo y la comunicación es correcta. Después la función *GetRelay* del archivo *xmlutil.c* se encarga de recuperar el valor del relay del archivo de datos antiguo.

La última función que queda por destacar es *void PutRelay(char address, unsigned char *num_frame, char mode, int fd_serial)* que es la encargada de mandar la trama al nodo con la orden de activar o desactivar su relay correspondiente según las decisiones tomadas en *control.c*.

Function.c

Es el archivo que contiene todas las funciones necesarias para realizar las operaciones de control. La función principal es *float getvalue (int sensor, int nodo, int tipo, int mode)* la cuál recibe el sensor del que tiene que obtener la medida y también incluye un parámetro que le indica si tiene que devolver la medida del archivo de datos instantáneo o del anterior a éste. La función encuentra el nodo y sensor objetivo y devuelve el valor de su medida.

La otra función implementada para las labores de control es la función matemática media. La función *float media (xmlDocPtr doc, xmlNodePtr node)* recibe como parámetro el nodo del documento de control que contiene la información de los elementos de los que se quiere obtener la media. Los valores de cada uno de esos elementos se suman y finalmente se dividen por el total de elementos sumados.

Dado el carácter escalable del sistema, para añadir funcionalidades al sistema de control sólo es necesario incluirlas dentro de este fichero.

Xmlutil.c

Este archivo contiene aquellas funciones relacionadas con el uso de archivos XML. Entre ellas *get_file_name* proporciona los nombres de los archivos históricos de medida *pcboxañomesdía hora.xml* y *get_pcbox_time* permite introducir en todos los archivos de datos la hora en que la medida es realizada.

GetFlag devuelve el estado de un flag de flag.xml cuyo nombre se le ha pasado previamente como parámetro a la función.

ActionFlag funciona de la misma manera que *GetFlag*, sólo que la función que realiza es modificar el estado del flag indicado fijando como su valor uno pasado como parámetro.

GetRelay es la función que obtiene del archivo de datos previo al instantáneo el valor que tenía almacenado cierto relay pasado como parámetro.

FijaRelay te almacena un valor pasado como parámetro en el valor de medida del relay objetivo en el archivo de datos instantáneo.

CreateXML te crea vacío un archivo de tipo XML cuyo nombre recibe la función como parámetro de entrada.

Red.cpp

Los objetos de la clase *CRed* tienen como componente un array de objetos de tipo *CNodo* y la clase para manejar estos objetos tiene diversas funciones. *ResetRed* elimina el contenido del array de nodos para vaciar la red. *FijaNodo* inserta un objeto de tipo *CNodo* en la red. *GetNumNodos* devuelve el número de nodos instalados. *GetNodo* devuelve el objeto *CNodo* seleccionado para operar con él. Por último *NodoAlive* devuelve un uno si el nodo de dirección igual a la introducida a la función como parámetro está instalado en la red y un cero en caso contrario.

Nodo.cpp

Los objetos de la clase *CNodo* tienen varios atributos como son la dirección, status, número de sensores, zona y un array de sensores. Esta clase dispone de dos constructores. Uno tiene como parámetros la dirección y el estado, el otro constructor fija además de estos dos el número de sensores y la zona.

Esta clase consta de numerosas funciones para fijar los atributos de status, zona, incluir un sensor en el array y obtener todos los elementos de la clase, *GetAddress*, *GetZona*, *GetStatus*, *GetNumSensors*, *GetArray* para obtener el array de sensores y *GetSensor* para poder manipular un sensor determinado.

Sensor.cpp

Por último la clase *CSensor* crea objetos con tres atributos:

- **ID**, identifica a cada sensor dentro de su correspondiente nodo.
- **Tipo**, caracteriza a cada sensor definiendo si es de Temperatura, Humedad, Luz o Relay
- **Medida**, almacena el valor de la magnitud correspondiente a cada sensor

La clase posee tres constructores, uno por defecto que inicializa todos los atributos a cero, otro que inicializa la ID del sensor y el tipo, y uno completo que inicializa los tres atributos la ID, el tipo y el valor de medida. La clase también consta de funciones para fijar y obtener el valor de cada atributo.

La Figura 3.7 muestra un cuadro resumen de la estructura de la Red Virtual

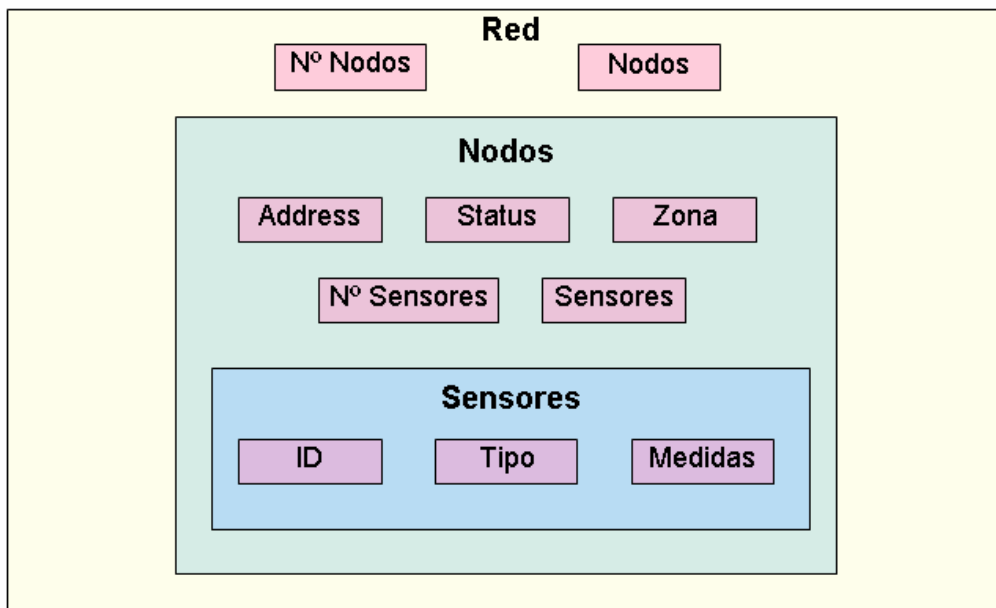


Figura 3.7 Estructura y Atributos de la Red Virtual

3.4. Conclusiones.

Tras la implementación del primer sistema de control elaborado para Magic Box se ha perseguido continuar esa línea de trabajo introduciendo mejoras e incorporando funcionalidades al producto original. El principal objetivo ha sido lograr un sistema de control **genérico, portable, dinámico, abierto, modulable y de fácil expansión.**

El Software de Control desarrollado permite el manejo de cuatro acciones principales:

- Detección de los nodos presentes en el bus de comunicaciones.
- Construcción de una red virtual como representación de la red instalada en el sistema.
- Toma de medidas en cada elemento de la red.
- Interpretación y ejecución de los módulos de control.

Estos procesos constituyen la funcionalidad del software de control. Con este sistema se satisfacen los objetivos planteados inicialmente, y fundamentalmente se consigue evitar la necesidad de detener y manipular el código fuente del sistema para la realización de cambios en la parametrización del control de la vivienda.

Aún así el logro absoluto de la automatización no es posible:

- El sistema está dimensionado para un máximo de 12 nodos, ya que el hardware es lento al responder y llevaría mucho tiempo dejarlo abierto.
- Diseñado para sensores de cuatro tipos, temperatura, humedad, luz y actuadores.
- Implementado con funciones de control sencillas como la obtención de valores de sensores y la realización de medias aritméticas.

Para la modificación de estos parámetros en la expansión del sistema sí es necesario la paralización de la ejecución y modificación de ciertos elementos del código. Pero sin dificultad para nuevos investigadores gracias a la estructura modulable del sistema. Los pasos necesarios para esta expansión se detallarán posteriormente en el *Capítulo 6 Conclusiones y líneas futuras*.

Capítulo 4

Software de Aplicación

4.1. Introducción.

En este capítulo se analizará el interfaz desarrollado para el correcto manejo del sistema por el usuario. Este subsistema software es el que otorga al sistema el dinamismo necesario para definir la configuración de la red y de los módulos de control. El software de aplicación permite al usuario modificar y manipular cada elemento de la configuración, ya sea incorporar nodos, desinstalarlos, cambiar sus sensores, modificar controladores, etc.

Para satisfacer las necesidades de comunicación con el sistema de control se ha realizado una aplicación web basada en una estructura de formularios para las configuraciones de los nodos, los sensores y los módulos de control.

La arquitectura de la aplicación web realizada posee una estructura cliente/servidor donde el cliente es el proceso que inicia el diálogo o solicita recursos, y el servidor responde a dichas solicitudes. En la aplicación desarrollada el servidor consta de tres niveles o capas. Como primera capa se emplea el navegador con elementos visuales, código HTML y una plantilla CSS, para homogeneizar la apariencia de la aplicación. Para generar contenidos dinámicos se utiliza PHP que constituye la segunda capa de la aplicación y permite la gestión del componente lógico de la aplicación.

La última capa está formada por los elementos que proporcionan el acceso a los datos. Se utilizan dos fuentes de información distintas, los archivos auxiliares generados en XML para el intercambio de información, y varias bases de datos realizadas sobre MySQL. En la Figura 4.1 se observa la arquitectura web del sistema.

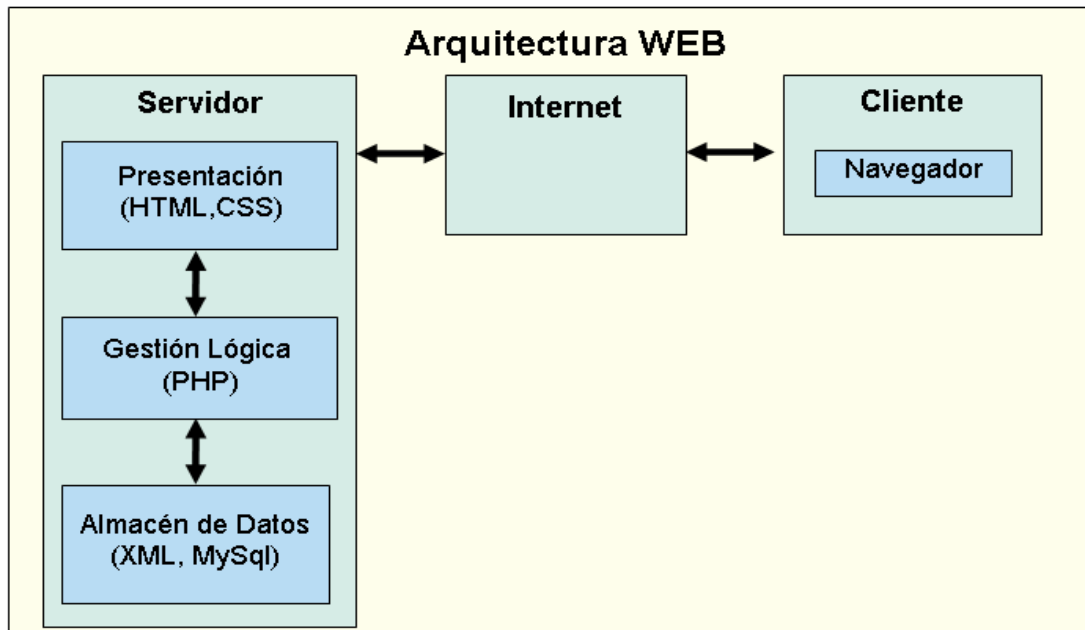


Figura 4.1 Arquitectura Web

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web. Dentro de una página Web se puede insertar código PHP que se ejecutará cada vez que se cargue esa página. El código PHP es interpretado en el servidor Web y genera código HTML y más contenidos para el cliente. PHP permite incorporar cierta lógica de negocio, y dinamismo a páginas web que de otra manera serían estáticas.

4.1.1. ¿Por qué PHP?

PHP fue creado en 1994 por Rasmus Lerdorf. Desde su creación ha experimentado cuatro importantes transformaciones hasta llegar en la actualidad a su quinta versión, PHP5, que es con la que se ha trabajado en este proyecto.

Algunos de los más importantes competidores de PHP son Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP), Allaire Cold Fusion. En

comparación con estos productos PHP presenta las siguientes fortalezas o ventajas [23] [24]:

- **Alto Rendimiento.** PHP es muy eficiente. Usando un servidor sencillo y económico, se pueden servir millones de visitas o hits al día.
- **Integración con diferentes sistemas de bases de datos.** PHP dispone de conexiones propias con muchos sistemas de bases de datos. Además de MySQL se puede conectar con las bases de datos PostgreSQL, mSQL, Oracle, dbm, filePro, Hyperwave, Informix, InterBase, y Sybase entre otras. Usa el estándar *Open Database Connectivity Standard (ODBC)* de manera que puede conectar con cualquier base de datos que suministre un controlador ODBC.
- **Bibliotecas integradas.** Como PHP se diseñó para su uso en la Web, dispone de muchas funciones integradas en librerías para realizar tareas relacionadas con el entorno Web.
- **Bajo coste.** PHP es gratuito y se puede descargar la última versión de <http://www.php.net>.
- **Fácil de aprender y de usar.** La sintaxis de PHP está basada en otros lenguajes de programación como C o Perl. Si ya se conocen estos lenguajes, o alguno tipo ANSI C, como C++ o Java la productividad usando PHP es casi inmediata.
- **Portabilidad.** Este lenguaje de programación está disponible para diferentes sistemas operativos. Se puede escribir código PHP en las versiones gratuitas de Unix como Linux o FreeBSD, en versiones comerciales como Solaris o IRIX, o en diferentes versiones de Microsoft Windows. La ejecución del código PHP obtendrá los mismos resultados independientemente del sistema operativo con el que se trabaje.
- **Disponibilidad del código fuente.** A diferencia de los productos comerciales o de código fuente cerrado, si se quiere modificar o añadir algo al lenguaje se puede hacer. No se necesita esperar nuevos parches del fabricante ni existe la preocupación de que el fabricante deje de proporcionar el producto.

- **Compatibilidad con el enfoque orientado a objetos.** La versión 5 de PHP que se ha usado en este proyecto cuenta con funciones orientadas a objetos, donde se encontrarán todas las funciones que se esperan de manejo de objetos, como herencia, atributos y métodos privados o protegidos, clases, constructores, destructores, etc.

En conclusión PHP es un producto de código abierto, eficiente, barato y de uso sencillo. El uso de PHP no depende de ningún navegador, (este proyecto ha sido probado en Internet Explorer, Mozilla FireFox y Konqueror), ni del Sistema Operativo (probado en Linux y Microsoft Windows). Además se integra fácilmente con numerosas bases de datos y posee una clase llamada DOM que proporciona una gran utilidad para el manejo de archivos XML. Por éstas y otras características se ha optado por PHP5 para el desarrollo de nuestro sistema.

Otro de los elementos a analizar es la elección del sistema de manejo de bases de datos, MySQL.

4.1.2. ¿Por qué MySQL?

MySQL es un sistema para la administración de bases de datos relacionales rápido y sólido. Una base de datos permite almacenar, buscar, ordenar, y recuperar datos de forma eficiente. El servidor de MySQL controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios, proporcionar acceso a dichos datos, y asegurarse de que sólo obtienen acceso a ellos los usuarios con autorización. Por lo tanto, MySQL es un servidor multiusuario, y de subprocesamiento múltiple [23]. Utiliza *SQL (Structured Query Language)*, que es el lenguaje estándar para la consulta de bases de datos utilizado ampliamente en el mundo.

MySQL lleva disponible desde 1996 pero su nacimiento se remonta a 1979. Se distribuye bajo un sistema de licencias dual. Puede ser utilizado bajo una licencia de código abierto (GPL), que es gratuita mientras se cumpla con las

condiciones de la misma. Sin embargo, si se desea distribuir una aplicación que no sea GPL y que incluya MySQL, se puede adquirir una licencia comercial.

Los principales competidores de MySQL son PostgreSQL, Microsoft SQL Server y Oracle. Algunas de las ventajas que ofrece MySQL con respecto a estos otros sistemas son [25]:

- **Rendimiento.** La arquitectura de MySQL permite configurar el servidor de bases de datos específicamente para aplicaciones particulares consiguiendo así excelentes resultados. Aunque la aplicación que se desee ejecutar requiera un sistema de alta velocidad de proceso o un servidor web que proporcione billones de peticiones al día, MySQL puede satisfacer los requisitos de los sistemas más exigentes gracias a sus utilidades de alta velocidad de carga, memorias cachés características, índices y otros mecanismos de mejora del rendimiento.
- **Bajo Coste.** Ya que está disponible de manera gratuita, bajo una licencia de código abierto. Si no se desea distribuir una aplicación o se trabaja en software gratuito, como es el caso de este proyecto, no es necesario adquirir una licencia comercial.
- **Facilidad de Uso.** MySQL ofrece una rápida puesta en marcha ya que requiere un tiempo medio de descarga e instalación inferior a 15 minutos. Esta norma se cumple para todo tipo de plataforma (Windows, Linux, Unix, Macintosh...). Además MySQL posee una completa gama de herramientas de gestión que le permiten controlar y solucionar problemas de varios servidores MySQL desde una misma máquina. MySQL permite también realizar tareas ETL (Extracción, Transformación y Carga) y de diseño de bases de datos, para proporcionar una completa administración de bases de datos, gestión del trabajo y monitorización de resultados.
- **Portabilidad.** Una de las características principales de MySQL es su flexibilidad e independencia de la plataforma con la que se trabaje. Se puede utilizar MySQL en una gran cantidad de sistemas Unix diferentes, así como bajo cualquier sistema Windows.

- **Acceso al Código Fuente.** Como en el caso de PHP se puede acceder al código fuente de MySQL y modificarlo en caso de necesidad. La naturaleza de código abierto de MySQL permite una completa personalización para aquellos que deseen añadir requerimientos particulares al servidor de la base de datos.

4.2. Objetivos.

En este proyecto se pretende lograr un sistema de control dinámico, con constante comunicación con el usuario, y permitiendo a éste la opción de configurar el sistema a su gusto, en cualquier momento, con facilidad y sin tener que detener la ejecución del sistema.

Mientras que la misión del software de control es reconocer las instrucciones que se le envían para procesarlas, la parte de aplicación es la encargada de elaborar las instrucciones precisas para la ejecución del sistema de control. Para ello se implementa una aplicación Web mediante la cual el usuario pueda introducir todos aquellos datos que sean necesarios para definir los elementos del sistema.

De esta forma el software de aplicación otorga al usuario el control sobre la configuración del sistema y las acciones a realizar en él. La aplicación proporciona una estructura encadenada de formularios para facilitar la definición tanto de los elementos de la red de trabajo, como de los módulos con las instrucciones de control.

Por lo tanto el principal objetivo de este elemento software es elaborar una aplicación web que sirva de interfaz hombre-máquina en nuestro sistema. De este modo se encargará de transmitir al sistema de control las instrucciones del usuario y proporcionará la monitorización de los datos obtenidos por el sistema, permitiendo una correcta comunicación entre ambos.

Como ya se ha comentado en la introducción del capítulo, para la realización de la aplicación se optó por una arquitectura de servidor formada por un motor en PHP utilizando como capa de datos los archivos auxiliares XML y la base de datos MySQL.

Los principales objetivos que se han fijado en la funcionalidad del software de aplicación son:

- **Elaboración de la aplicación web.** Construir la infraestructura necesaria para dotar al sistema de comunicación. Dentro de este objetivo el principal enfoque ha sido lograr una gran funcionalidad para la manipulación y gestión del sistema de control. Aún así se ha procurado que el interfaz sea amigable e intuitivo.
- **Configuración de la Red.** Informar al usuario de los nodos que se encuentran instalados en el bus. Permitirle instalar o desinstalar los nodos que desee, guiándole con sencillas instrucciones en la introducción de los parámetros que no son automáticamente proporcionados por el hardware.
- **Configuración de los Módulos de Control.** Proporcionar al usuario la posibilidad de establecer los parámetros de las funciones de control a ejecutar. Pese a ser posiblemente el elemento más complejo de la aplicación, se ha realizado de manera que sea intuitivo.
- **Activación de los Módulos de Control.** Se muestra al usuario un cuadro con toda la información asociada a los módulos instalados. Éste puede actuar eligiendo los distintos módulos que se desea que efectúen la labor de control del sistema. Los módulos que se seleccionen se activarán y serán ejecutados sin necesidad de paralizar la ejecución del sistema.
- **Monitorización del Estado del Sistema.** Proporciona la visualización del estado de los elementos integrantes de la red. Se facilitan diferentes visualizaciones enfocadas según distintas agrupaciones de estos elementos a elección del usuario, totalidad de la red, elementos de un nodo, sensores del mismo tipo o sensores pertenecientes a la misma zona de la vivienda.

Además de estos objetivos funcionales se han tenido también en cuenta aquellos objetivos estructurales presentes en todo el proyecto como son la elaboración de un sistema genérico, dinámico, de código abierto y modutable.

4.3. Estructura.

Al igual que se realizó con el Software de Control se procederá a continuación a analizar la estructura que se ha fijado para este componente Software. Se estudiará cual es la arquitectura que permite la evolución de cada uno de los procesos de configuración y actuación de la aplicación desarrollada.

Para un mejor entendimiento se muestra en la Figura 4.2, un esquema global del subsistema Software de Aplicación.

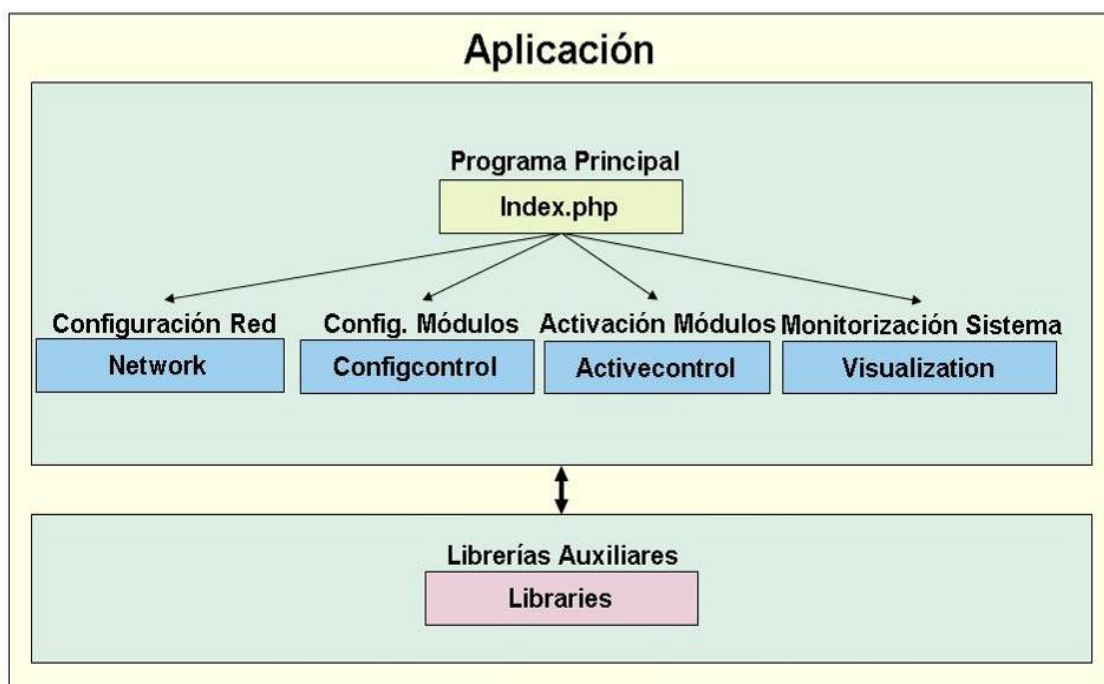


Figura 4.2 Software de Aplicación

4.3.1. Gestión del Sistema.

La aplicación cuenta con un menú principal (Ver Figura 4.3) que muestra al usuario diferentes opciones para actuar con el sistema. Se presentan al usuario cuatro opciones de actuación:

- **Configuración de la Red.**
- **Configuración de los Módulos de Control.**
- **Activación de los Módulos de Control.** Esta opción sólo aparece si existe algún módulo instalado.
- **Monitorización del estado del sistema.** Esta opción se encuentra habilitada únicamente cuando se han realizado y almacenado medidas instantáneas.

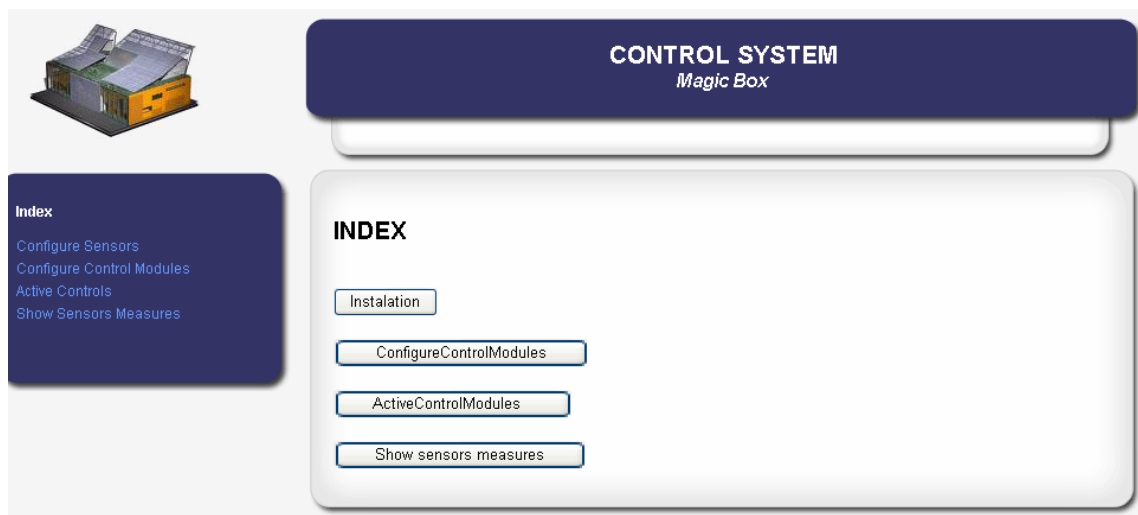


Figura 4.3 Menú Principal de Aplicación

Una vez el usuario determine iniciar una de estas operaciones la aplicación lo irá guiando hasta completar cada proceso. Durante todo el proceso de aplicación, el sistema de control continúa ejecutándose.

4.3.2. Configuración de la Red.

En la realización de un sistema de control en una vivienda, es muy común que en algún momento se deseen hacer cambios en la configuración de la red de control, bien añadiendo nuevos nodos, modificando los sensores de un nodo o eliminando nodos que ya no son de utilidad.

Una vez que el usuario decide modificar la configuración de la red se avisa al sistema de control que proceda a realizar la localización de posibles nodos instalados físicamente en la red. Una vez finalizado este proceso las direcciones de los nodos detectados se almacenan en una base de datos MySQL. Después se comparan los nodos detectados con los nodos que estuvieran instalados por una configuración previa de la red.

La aplicación notifica al usuario qué nodos detectados son nuevos y se pueden instalar y qué nodos están ya configurados permitiendo eliminarlos.

Instalación de Nodos

Como ya se ha descrito anteriormente los atributos de un nodo son dirección, status, zona y número de sensores. La dirección y el status son datos que se obtienen de la propia respuesta del hardware, pero el hardware no es capaz de comunicar la zona del nodo y el número de sensores que posee. Estos datos se solicitan sean introducidos por el usuario por medio de menús desplegables de selección.

Como primer modelo de esta aplicación se han configurado como zonas seleccionables las zonas hábiles de la vivienda Magic Box: cocina, salón, comedor, oficina, dormitorio, baño, falso suelo, tejado y los exteriores Norte, Sur, Este y Oeste, y se ha fijado un número máximo de siete sensores por nodo aunque no se ha trabajado con ninguno que posea más de cuatro dada la limitación del Hardware.

Finalizada la introducción de los datos de los nodos se procede al procesado de los sensores. Un sensor consta de una *ID* que la da la propia configuración, *tipo* que es necesario que se introduzca y *medida* que en principio se inicializa con valor por defecto uno.

Los tipos de sensores configurados en esta primera versión del programa son sensores de temperatura, de humedad, de luz y actuadores o tipo relay o triacs. La aplicación sería igualmente funcional con cualquier otro tipo de sensores que se desearan instalar, las pocas modificaciones necesarias para incluir sensores de otro tipo se explican en el *Capítulo 6*.

Cuando concluye el procesado de la información de los sensores se introducen los datos del nodo y sus respectivos sensores en la base de datos, se crea una nueva configuración de la red física y se indica a la parte de control que puede proceder a la creación/modificación de la red virtual. Para estas dos acciones se utilizan dos funciones (*creainstalxml*) y (*setflagxml*) de la librería auxiliar creada para el manejo de documentos XML. Finalizada la instalación del nodo, el usuario puede volver a la configuración de nodos o volver al menú principal si ya se ha creado la red de interés.

Eliminación de Nodos.

En caso de que se desee eliminar algún nodo de la red, bien por que no se necesite más o para añadirle nuevos sensores, se permite la eliminación de todos sus datos de manera que no interfiera el funcionamiento del sistema completo.

Antes de proceder a borrar la información del nodo se solicita al usuario la verificación de su deseo de eliminar dicho nodo. Si se continúa con el proceso de eliminación del nodo se borra toda su información de la base de datos y al igual que en la finalización del proceso de instalación, se crea la nueva

configuración de red y se avisa al subsistema de control que puede proceder a la modificación de la red virtual.

4.3.3. Configuración de los Módulos de Control.

Sin duda ésta constituye la parte más compleja de la aplicación, pero gracias a la modularidad del código y a que cada proceso se efectúa paso a paso se espera que no resulte complicado el manejo de la herramienta para usuarios no iniciados.

La configuración de los módulos de control permite al usuario instalar un nuevo módulo o borrar un módulo existente.

Ejemplo de un módulo de control

Para facilitar la comprensión del desarrollo de esta configuración será muy útil recordar cual es la estructura de una función de control. Claramente se pueden diferenciar dos partes en un módulo de control, la condición principal y las acciones a llevar a cabo en función del cumplimiento de esa condición, tal y como se observa en la Figura 4.4.

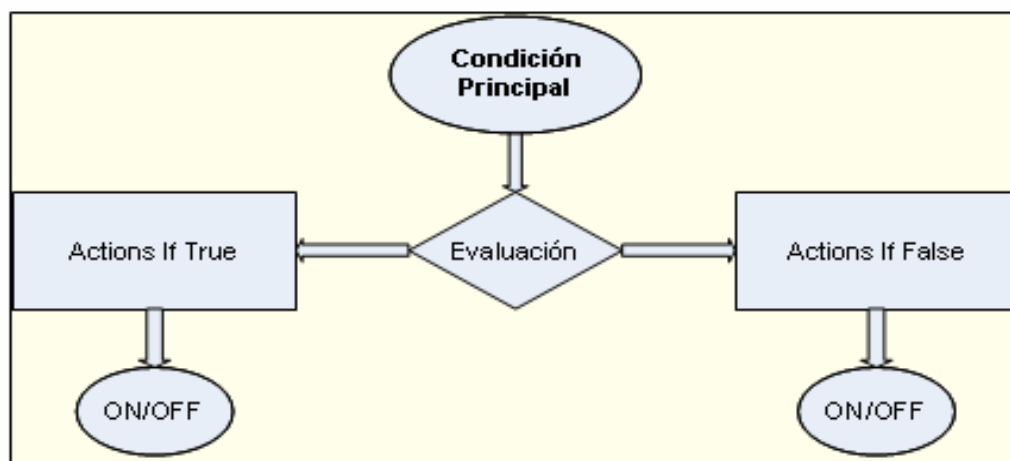


Figura 4.4 Esquema Módulo de Control

La condición general está formada por una o varias condiciones simples que se muestran en la Figura 4.5.

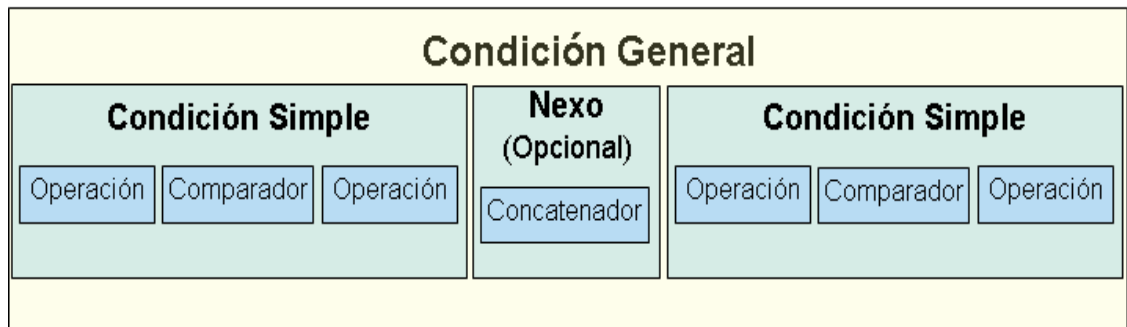


Figura 4.5 Esquema Condición General

Un ejemplo de enunciado para dicha condición sería:

“Si la media de los sensores de temperatura de los nodos 1 y 3 es mayor que 31 y el valor de la temperatura del nodo 5 es mayor que la temperatura del mismo sensor en la toma anterior...”

Esquemáticamente esta condición estaría formada por los siguientes elementos.

If ((media (T1,T3) > 31) and (T5(t) > T5(t-1)))

operación comparador operación link operación comparador operación

Se puede observar nuevamente la agrupación de condiciones simples que tienen siempre la estructura de **operación-comparador-operación** unidas mediante nexos o concatenadores.

El módulo de control puede tener diferentes respuestas en función del resultado positivo o negativo de la condición principal. Las respuestas serán del tipo ON y OFF, pudiendo incluirse también condiciones dentro de la misma respuesta.

La Figura 4.6 muestra un ejemplo de cómo se pueden estructurar las diferentes respuestas.

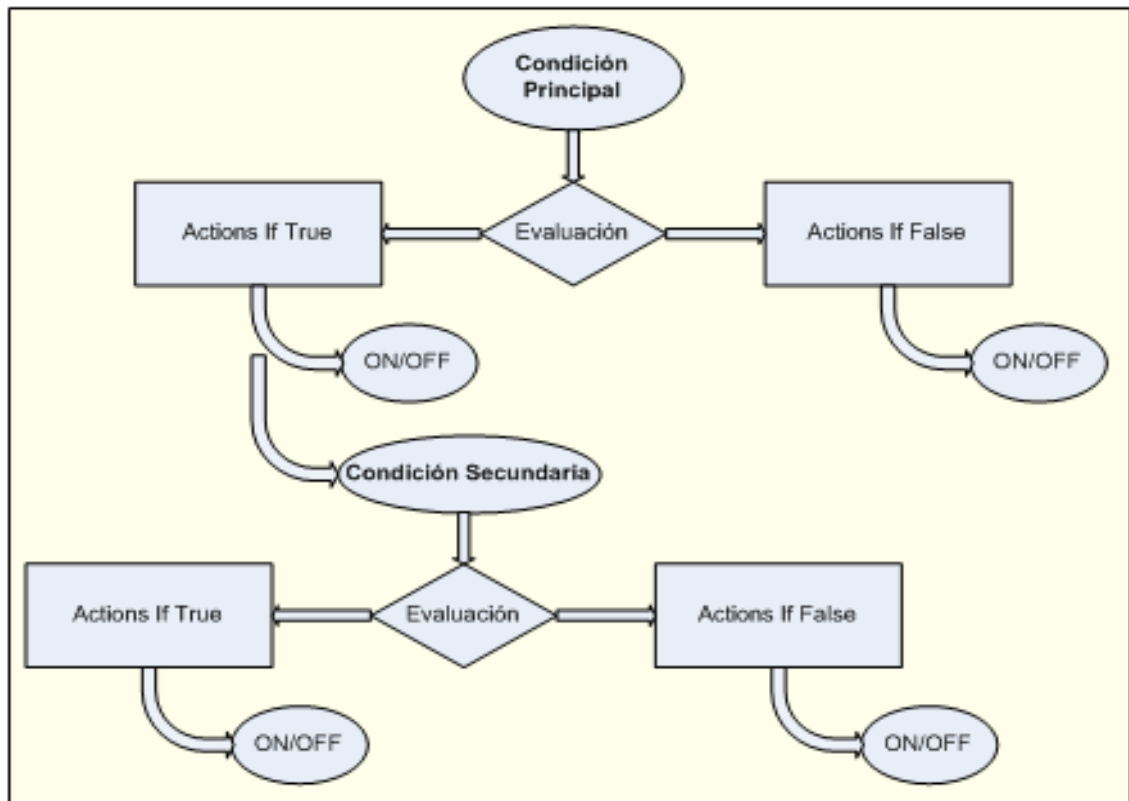


Figura 4.6 Esquema Bloque de Respuestas

Las acciones a realizar para el ejemplo del enunciado anterior pudieran seguir la siguiente estructura:

“...Si la condición es cierta activar el relay del nodo 1 y si además el relay del nodo 5 está desconectado activarlo. Si la condición es falsa desactivar el relay del nodo 3.”

- Actions if true
ON R1
If R5= OFF then ON R5

- Action if false

OFF R3

Debido a estas condiciones, que pueden ir incluidas dentro de una repuesta, tenemos que distinguir entre las acciones que hay que realizar tomando como referencia la condición principal y tomando como referencia la condición insertada en la respuesta. Para esto se ha generado un indicador del orden de indexación, donde las acciones que correspondan a la condición general tienen el identificador 0 y las acciones que correspondan a otras condiciones tendrán su propio identificador.

El identificador del nivel de indexación junto con el nombre del módulo se envía en todos los puntos del proceso de configuración. De esta manera la aplicación siempre tiene un control exacto de donde tiene que situar cada elemento que procesa.

Tras estos ejemplos, se procede a explicar la configuración general de un módulo de control.

Configuración de un módulo de control

Para la instalación de un nuevo módulo el usuario puede introducir el nombre del módulo y una descripción de su funcionalidad, disponiendo de dicha información en la parte de activación de módulos.

Como se ha visto se puede comprobar que la condición principal siempre estará formada por una condición o varias condiciones simples unidas. El primer elemento de una condición simple siempre será una operación y se procesa en *Getoperation.php*. En el proceso de esta operación hay dos posibilidades:

- Si se trata de la primera operación del módulo se guarda en *control.xml* la información del módulo para que todos los demás elementos se vayan

quedando asociados a él. Por lo tanto en el documento de control se almacenaría el nombre del módulo y la descripción, y se inicializan el estado y las dependencias a cero.

- En el caso de que no sea la primera operación del módulo (puede ser posterior a un concatenador o proceder de una condición incorporada en una respuesta) se obvia el almacenamiento de la información del módulo y se procede directamente a la selección del tipo de operación.

En principio se han instalado cuatro tipos de operación (Ver Figura 4.7):

- Valor de sensor.
- Función media aritmética.
- Valor numérico.
- Valor de sensor en medida anterior.

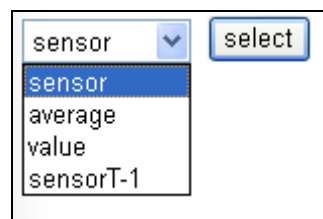


Figura 4.7 Selección de Operación

El valor de estos cuatro tipos se procesa en el siguiente paso de la configuración. Además del tipo de operación también se indica cual es el siguiente elemento que se espera (comparador o concatenador).

En la Figura 4.5 se puede comprobar que después de una operación puede ir un comparador (principio de una condición simple) o un concatenador (final de una condición simple).

La aplicación avanza posteriormente a `Procesoperation.php` y realiza diferentes procesos en función del tipo de operación que se le haya enviado:

- En el caso de que la operación sea un valor de un sensor, o sea un valor de un sensor de una medida anterior, el tratamiento es el mismo. Se muestran los sensores disponibles en la red por medio de la función (*showsensor*) de la librería auxiliar de sensores.
- En el caso de la función media, la aplicación procede a preguntar al usuario el número de sensores de los que se desea realizar la media. Y una vez introducido este dato se muestran tantos menús de selección de sensores (*showsensor*) como el usuario ha solicitado.

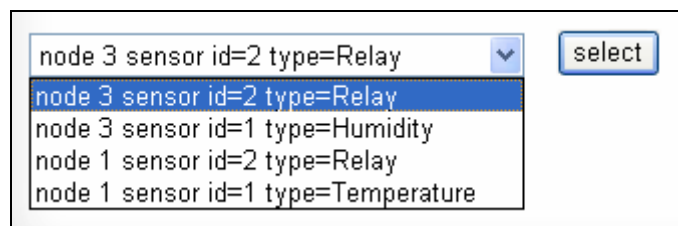


Figura 4.8 Selección de Sensor

- Si la operación que se solicita es un valor numérico la aplicación solicita al usuario que introduzca el valor umbral.

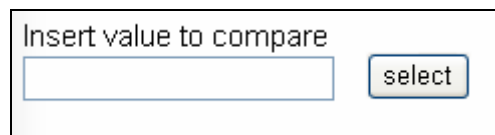


Figura 4.9 Selección de Valor Umbral

El sensor o sensores seleccionados o el valor numérico introducido en la operación se procesan en el siguiente paso de la configuración, donde se selecciona también el siguiente elemento de la instrucción, que como se ha comentado anteriormente podrá ser un comparador o un concatenador. La aplicación maneja la información necesaria para dirigir la configuración al elemento adecuado haciendo este proceso de manera transparente para el usuario. De manera que si detecta que el siguiente elemento es un comparador se dirige a *Getcomparator.php* y si detecta que el siguiente elemento es un concatenador se ejecutará *Getlink.php*.

En el proceso normal de configuración después de la primera operación se procedería a seleccionar el comparador. *Getcomparator.php* lo primero que hace es almacenar la operación seleccionada en el paso anterior en el documento de control, dentro de su módulo y su nivel de indexación correspondiente (en principio recordamos el nivel sería cero y la operación se asociaría al la condición principal del módulo).

En el caso de que los elementos de la operación sean sensores se introducen las direcciones de los nodos implicados en la ejecución del módulo, en el atributo de dependencias del módulo, así proporcionará al software de control un elemento de seguridad de la ejecución del sistema.

Toda esta operación siempre es transparente al usuario. Éste lo único que percibe es una información del estado de la configuración que se está realizando (función *readxml* de la librería auxiliar de módulos) y la opción de seleccionar el comparador que desee (función *showcomparator* de la librería auxiliar de control).



Figura 4.10 Selección de Comparador

Una vez seleccionado el comparador deseado se almacena en el archivo de control en *Postcomparator.php*, nuevamente dentro de su módulo y su nivel de indexación. Se solicita de nuevo la selección de una operación para finalizar la condición simple y se procede al análisis de esa selección de la operación. Para ello la aplicación vuelve a *Procesoperation.php* pero esta vez con la

indicación de que el próximo elemento es un concatenador. Por lo tanto seleccionada la última operación de la condición la ejecución pasa a *Getlink.php*.

Getlink.php comienza almacenando la información de la segunda operación comprobando de nuevo su correcta ubicación en la estructura de control.xml y actualizando las dependencias. En este punto del proceso ya se tiene configurado el primer bloque **operación-comparador-operación**, este bloque podría ser el único si el usuario en la elección del concatenador escogiera la opción “ninguno”. La selección del concatenador se evalúa en *Postlink.php*.

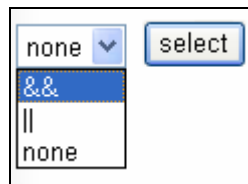


Figura 4.11 Selección de Concatenador

En *Postlink.php* se interpreta cual es el elemento concatenador que se ha elegido para decidir cómo continuar con la configuración del módulo. Si se ha elegido un elemento de linkado (*and* u *or* lógicos) entonces el siguiente elemento sería otro bloque de condición simple como los que ya se han revisado. En este caso se almacenaría el concatenador seleccionado en el documento de control y se volvería a la ejecución de *Getoperation.php* repitiendo el proceso descrito hasta ahora de obtención de una condición simple.

Pero en el caso de que no se haya seleccionado ningún concatenador se procede a la selección del tipo de acción, acción si la condición es cierta o acción si la condición es falsa (función *action* de la librería de control).

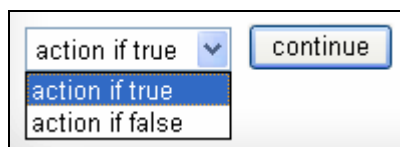


Figura 4.12 Selección de Acción

Como cada acción que se desee configurar comenzará su proceso en este archivo, Postlink.php tiene implementada una cierta lógica para determinar el nivel de indexación al que corresponde cada acción que debe empezar a procesar. En el caso de que la acción provenga de la condición principal, nivel cero de indexación, procede a la configuración de la respuesta en ese mismo nivel, cero. Pero si procede del final de una respuesta con cierto nivel de indexación, entonces interpreta una orden introducida por el usuario del nivel de indexación que corresponde a la nueva acción.

Esta operación se entiende mucho mejor con un ejemplo ilustrativo. En la configuración del módulo de control dentro de las acciones que se ejecutarán en caso de que se cumpla la condición principal (nivel 0) es posible que se requiera la evaluación de otra condición más (nivel 1), eso implicaría que las acciones relacionadas con esta nueva condición se tienen que ejecutar en función del cumplimiento de la condición nueva (nivel 1) y no de la antigua (nivel 0). Y en función de la complejidad del control aún se podrían incluir más niveles.

Más adelante se verá que cuando el usuario ha acabado de configurar una respuesta puede solicitar incorporar más respuestas al módulo. Supongamos que el usuario hubiera configurado una respuesta en un nivel 2 de indexación, la aplicación le permitiría elegir asociar la siguiente respuesta al mismo nivel de indexación en que estuviera (nivel 2), en un nivel inferior (nivel 1), o asociar la respuesta al nivel principal (nivel 0).

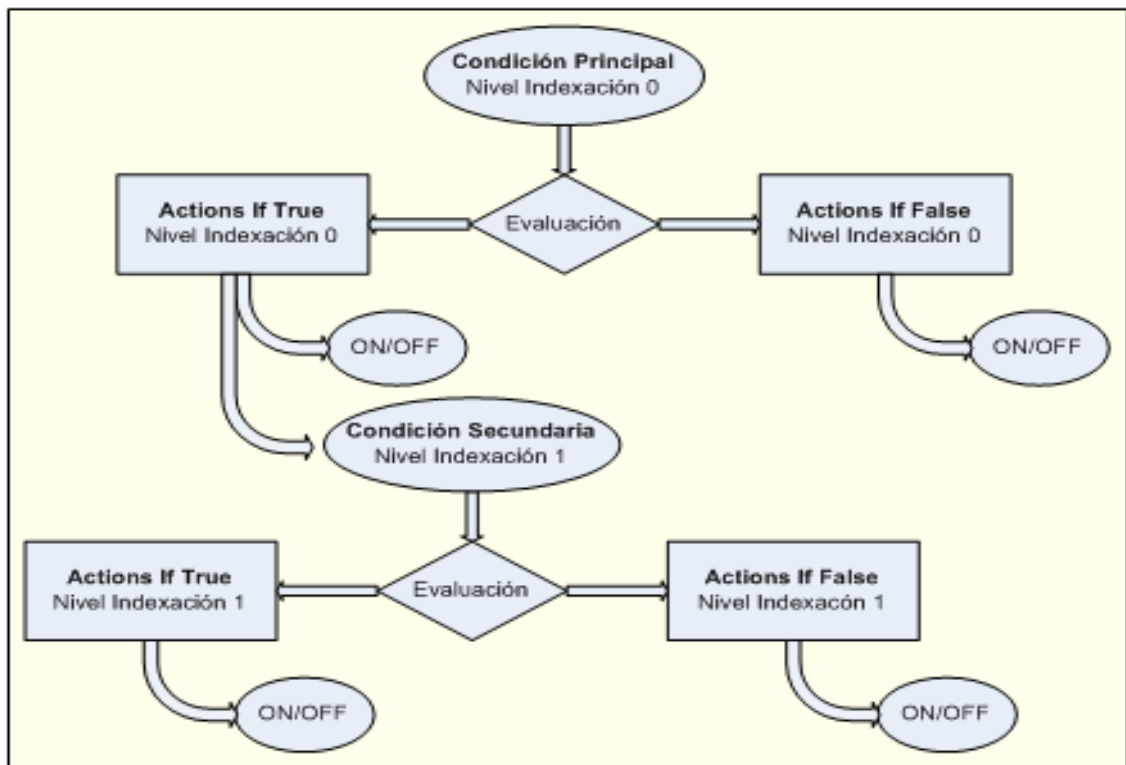


Figura 4.13 Niveles de Indexación de Respuestas

Para controlar estos niveles de indexación *Postlink.php* usa dos funciones de la librería auxiliar general, *lessid* que disminuye en uno el nivel de indexación y *downid* que fija el nivel a cero, el valor asociado a la condición principal. Identificado el nivel de la respuesta se procede a determinar su acción en *selectaction.php*.

Selectaction.php permite seleccionar el tipo de acción a realizar. Hay tres tipos de acciones posibles activar o desactivar un actuador (ON/OFF) y la opción de introducir otra condición a evaluar (if). La acción seleccionada se evalúa en *Getanswer.php*.

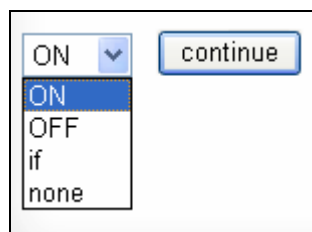


Figura 4.14 Selección de Respuestas

En caso de que sea una nueva condición, se incrementa el nivel de indexación en uno, se almacena el tipo de respuesta en el documento de control y se vuelve de nuevo a *Getoperation.php* con el actualizado nivel de indexación a generar la nueva condición de control.

Si la acción es activar o desactivar un actuador se ejecuta *Getrelay.php*, que muestra los actuadores disponibles en la red para ser utilizados por medio de la función *showrelay* de la librería auxiliar de sensores. Los datos de la acción y el relay elegidos se almacenan en *Postrelay.php* que constituye el último archivo del proceso de configuración. Así mismo se actualizan las dependencias del módulo con la dirección del nodo al que pertenece el relay.

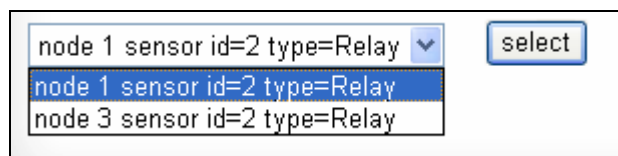


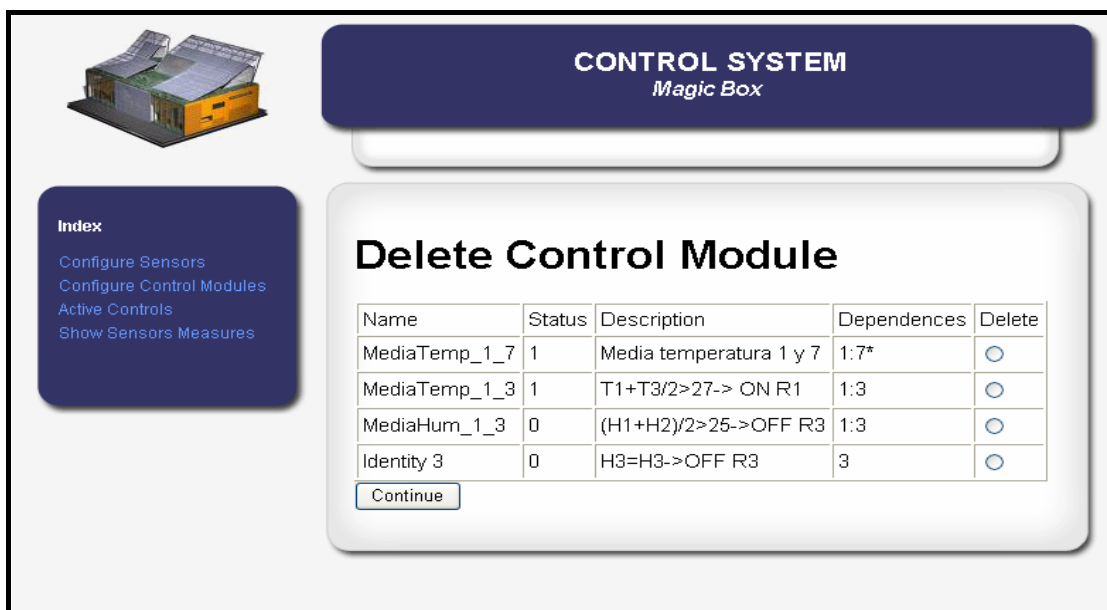
Figura 4.15 Selección del Relay

Una vez acabada la instalación de la respuesta, el usuario puede dar por terminada la configuración del módulo volviendo al menú principal o añadir más respuestas, y en este caso si el nivel de indexación en que se encuentra es distinto de cero. Puede elegir que la siguiente respuesta pertenezca al mismo nivel de indexación, a un nivel inferior o al nivel principal, esa información se almacenará en una variable que *Postlink.php* interpretará para instalar la nueva respuesta en el lugar que corresponda.

Eliminación de un módulo de control.

En el caso de que se desee desinstalar un módulo de control y borrar la información asociada a él. Se muestra en *Deletemodule.php* una tabla (función *selectmodules* de la librería auxiliar de módulos) con todos los módulos instalados y los datos principales de cada uno de ellos como el nombre, la descripción, el status y las dependencias. (Ver Figura 4.16)

En caso de que exista un problema con algún nodo de las dependencias, esto es, que el módulo necesite un nodo que no esté instalado o no funcione, esa información se le indica al usuario acompañando la dirección del nodo que genera error con un asterisco.



Name	Status	Description	Dependences	Delete
MediaTemp_1_7	1	Media temperatura 1 y 7	1:7*	☐
MediaTemp_1_3	1	T1+T3/2>27-> ON R1	1:3	☐
MediaHum_1_3	0	(H1+H2)/2>25->OFF R3	1:3	☐
Identity 3	0	H3=H3->OFF R3	3	☐

Continue

Figura 4.16 Eliminación Módulo de Control

La aplicación permite seleccionar un módulo y proceder a eliminar sus datos o volver al menú principal. Si se selecciona un módulo para ser eliminado se advierte al usuario que la información de ese módulo será borrada y no se podrá recuperar, dándole nuevamente la opción de confirmarlo, o cancelar y volver al menú principal. En caso de seguir adelante con la eliminación del

módulo se borran los datos del archivo de control y se regresa al menú principal.

4.3.4. Activación de los Módulos de Control.

Debido a que el proceso de configuración de un módulo es bastante largo, todos los módulos se configuran inicialmente con status a cero de manera que no se pueda intentar ejecutar un módulo que no se ha finalizado de configurar.

Para activar los módulos de control que se desean ejecutar se muestra una tabla (función showmodules de la librería auxiliar de módulos) con todos los módulos instalados y los datos principales de cada uno de ellos del mismo modo que se realizaba en el proceso de eliminación de módulos, con el nombre, la descripción, el status y las dependencias.



Figura 4.17 Activación Módulos de Control

La aplicación informa al usuario del estado de todos los nodos involucrados en cada módulo de control, de esta forma si se necesita un nodo que no se encuentra instalado se indica la dirección del nodo que genera error por medio

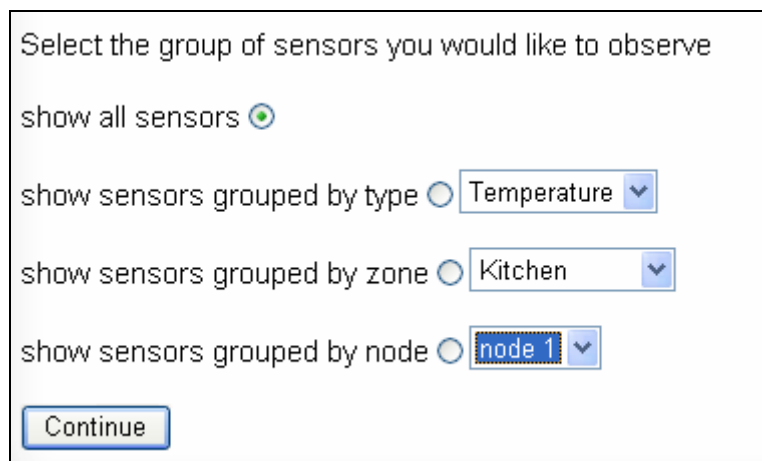
de un asterisco.(Ver Figura 4.17) El usuario puede seleccionar el estado de todos los módulos de control instalados. Los que se encuentran activos se marcan inicialmente como tal, del mismo modo que los que tienen el status a cero aparecen desactivados. Los valores de status que se seleccionen se almacenan en el archivo *control.xml* y se vuelve al menú principal.

4.3.5. Visualización del Estado de la Red.

La última opción de la aplicación permite comprobar el estado de los elementos de la red agrupando los sensores que se deseen visualizar en función de distintas características. Hay cuatro opciones de visualización: (Ver Figura 4.18)

- Todos los sensores de la red
- Sensores de un mismo tipo
- Sensores de una misma zona
- Sensores de un mismo nodo.

Para ello usan las funciones *visualall*, *visualtype*, *visualzone*, o *visualnode* respectivamente, funciones de la librería auxiliar de visualización.



Select the group of sensors you would like to observe

show all sensors ☒

show sensors grouped by type ☐ Temperature ▼

show sensors grouped by zone ☐ Kitchen ▼

show sensors grouped by node ☐ node 1 ▼

Continue

Figura 4.18 Selección de Visualización

Una vez elegida la visualización se muestra una tabla con la información de los sensores seleccionados. La tabla muestra la ID que identifica al sensor dentro de su nodo, el tipo de sensor, la medida o estado del sensor, el nodo al que pertenece y la zona donde se encuentra ubicado.

showing sensors of node: 1

Sensor ID	Type	Measure	Node	Zone
3	Relay	ON	1	Kitchen
2	Humidity	25%	1	Kitchen
1	Temperature	25°C	1	Kitchen

Select other visualization

Index

Figura 4.19 Visualización sensores nodo 1.

La aplicación permite recargar la misma visualización para poder actualizar los datos que se están mostrando, seleccionar otra visualización distinta o volver al menú principal.

4.3.6. Librerías Auxiliares.

Para dotar al código de mayor sencillez y modularidad, y así mismo aprovechar la reutilización de funciones necesarias en distintas partes de la aplicación, se han agrupado en la carpeta *libraries* aquellos documentos que proporcionan el soporte funcional de todo el Software de Aplicación.

Como punto final de la explicación de la estructura del Software de Aplicación, se comentarán las distintas funcionalidades que aportan cada una de estas bibliotecas auxiliares:

- **Library.** La más general de las librerías aporta la codificación de la cabecera de cada archivo y el manejo del nivel de indexación.
- **Nodelibrary.** Contiene las utilidades que proporcionan el manejo de los nodos, seleccionar su instalación, eliminación, selección de un nodo, envío de todo el array de nodos disponibles, y muestra, obtención y selección de la zona de un nodo.
- **Sensor Library.** Se encarga del manejo de la información de los sensores. Incluye la selección de sensores, selección de sensores actuadores, y muestra, obtención y selección del tipo del sensor.
- **Xml Library.** En este documento se crea el archivo que guarda la estructura de la red, *Instalación.xml*. También se efectúa el manejo de los indicadores de *flag.xml* para la comunicación con la parte de control.
- **Control Library.** Aporta las funciones para seleccionar cada uno de los elementos de una función de control, comparadores, concatenadores, operaciones, acciones y respuestas.
- **Module Library.** Permite el manejo de la información asociada con los módulos de control. Manejo de las dependencias, lectura de los elementos de un módulo, así como la muestra y selección de los módulos disponibles.
- **Visual Library.** Esta biblioteca auxiliar maneja los elementos de la visualización del estado del sistema, gestionando la monitorización de los elementos que el usuario ha solicitado.

4.4. Conclusiones.

Se ha implementado una aplicación Web que proporcione la interfaz necesaria para el manejo de la aplicación por el usuario, y que permita gestionar la configuración de la red de control de la vivienda, la configuración y puesta en marcha de diferentes módulos de control del sistema y la monitorización del estado del sistema.

Para realizar todas estas funciones la aplicación Web utiliza PHP como motor para generar contenidos dinámicos, y como capa de obtención de datos utiliza bases de datos implementadas en MySql y los archivos auxiliares XML que proporcionan la comunicación entre el sistema de control y de aplicación.

El Software de Aplicación constituye un paso adelante en la evolución del sistema de control original. Gracias a este componente software el dominio sobre la actuación del sistema en cada momento pertenece al usuario. La aplicación proporciona una estructura encadenada de formularios y una gestión de la información del sistema que permite la fácil configuración de los elementos del sistema de control, tanto los elementos de la red de trabajo, como los módulos con las instrucciones de control.

Dentro de los objetivos generales del proyecto de elaboración de un sistema de control genérico, abierto y portable, el Software de Aplicación tiene como principal función incorporar dinamismo al sistema, facilitar su expansión y proporcionar al usuario el control de la configuración y ejecución del sistema.

Para lograr estos objetivos el Software de Aplicación aporta una serie de funcionalidades:

- **Configuración de la red de nodos y sensores.** El sistema comunica los nodos que se encuentran instalados en el bus de comunicaciones y el usuario puede instalar los nodos y sus sensores definiendo sus características o eliminar un nodo y sus componentes de la red.
- **Configuración de los módulos de control.** La aplicación proporciona secuencialmente al usuario los elementos necesarios para facilitar la construcción de distintos módulos de control de una forma estructurada que permita su interpretación y ejecución automática. Así mismo el usuario también puede eliminar los módulos de control que desee.
- **Activación de los módulos de control.** El usuario mediante la visualización de la información más importante de cada módulo puede seleccionar los módulos que desea que sean ejecutados.

- **Monitorización del estado del sistema.** La aplicación permite al usuario visualizar el estado de la sensorización del sistema. Para favorecer el acceso a elementos concretos del sistema se proporcionan distintas visualizaciones según distintas tipologías de sensores.

Estos cuatro procesos constituyen la funcionalidad del Software de Aplicación, que permite la realización de los objetivos propuestos, que principalmente se puede resumir en dos:

- Proporcionar al usuario los elementos necesarios para que pueda definir el control del sistema.
- Estructurar la información de la configuración del control proporcionada por el usuario de manera que el software de control pueda interpretarla y ejecutarla.

Capítulo 5

Aplicación Práctica

5.1. Introducción.

Se ha realizado una prueba del sistema desarrollado en un escenario real, la vivienda **Magic Box**. En este capítulo se proporcionarán detalles acerca de la instalación realizada, el funcionamiento de los distintos elementos del sistema y las conclusiones que se pueden extraer de los resultados bioclimáticos obtenidos en la vivienda.

5.2. Instalación.

Se ha utilizado la vivienda Magic Box como laboratorio de pruebas del sistema, aprovechando su arquitectura bioclimática y los numerosos elementos de control que posee [13]. Para ejercer el control bioclimático de la vivienda se ha actuado sobre los servomotores que accionan las rejillas norte y sur gestionando las entradas de aire del exterior y las turbinas que se encargan de la generación de corrientes de aire del suelo térmico.

Se ha instalado una arquitectura serie basada en el estándar RS-485 como ya se ha descrito en el Capítulo 2. Para ello se ha utilizado un ordenador empotrado como ordenador central que se encarga de la gestión del funcionamiento de todo el sistema. La red de adquisición y actuación se ha formado con 5 nodos. Estos nodos nos permitirán registrar medidas de tres zonas distintas de la vivienda:

- El interior de la vivienda que conforma el hábitat donde se desenvuelven los habitantes de la misma.
- La zona de invernadero sur y la zona de rejillas norte, que se analizará para aislar o favorecer su influencia en el interior de la casa.
- El suelo térmico situado debajo del suelo de la vivienda. Las características de este suelo se utilizarán para tratar de mantener el interior de la vivienda en unas condiciones ambientales suaves.

La aplicación práctica se ha desarrollado sobre el ala este de la vivienda, donde están situadas la habitación y la oficina. No se ha podido realizar una aplicación en toda la vivienda al no encontrarse acabada la construcción de la misma. La distribución de los nodos se puede visualizar en la Figura 5.1

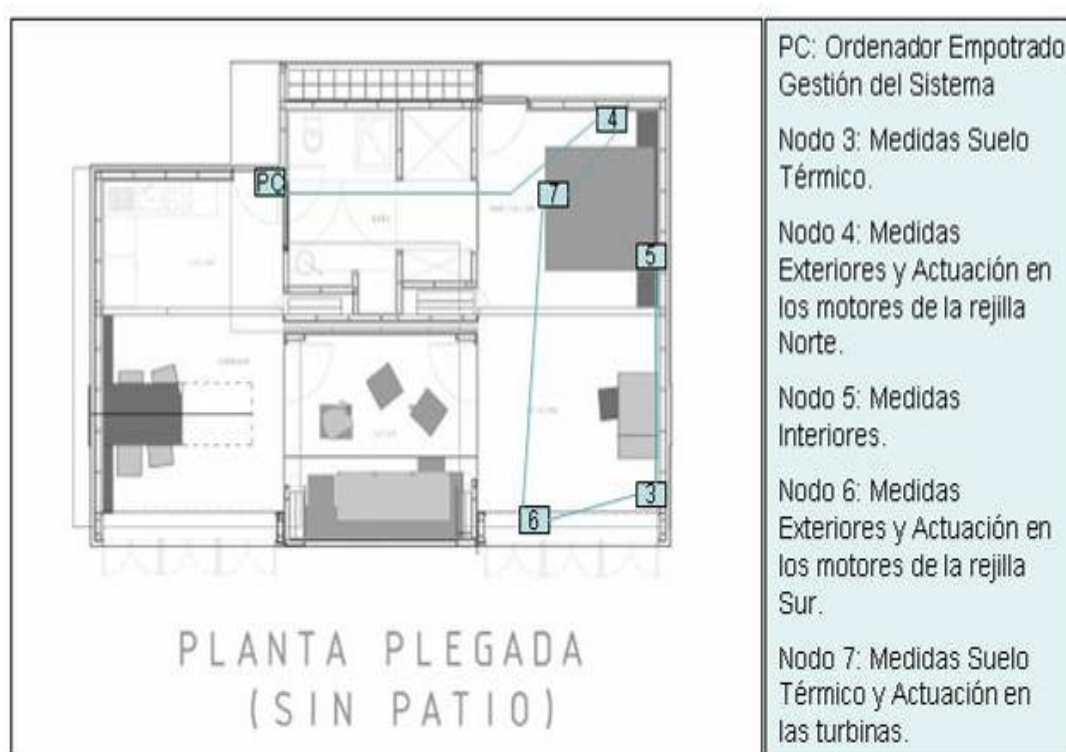


Figura 5.1 Distribución arquitectura sistema

5.3. Sistema de Control.

El acondicionamiento térmico ambiental de la vivienda es enteramente pasivo. La fuente de energía en invierno es la radiación solar, a la que se unen las cargas internas (ocupación, alumbrado y equipos) mientras que en verano la única energía empleada es el frescor aportado por la ventilación nocturna.

En ambos casos se precisa de un sistema de acumulación que reparta la energía captada a lo largo de la totalidad del día. En el diseño de Magic Box se optó por la acumulación de energía en forma de calor latente; es decir, provocando el cambio de estado de una sustancia aportando energía y recuperándola invirtiendo el proceso.

Los sistemas de acumulación, geles de cambio de fase, empleados se encuentran en cápsulas herméticas de aproximadamente 3 Kg. cada una (dimensiones 28 x 48 x 3 cm.). Estas cápsulas están colocadas en el hueco de un suelo técnico, sobre un aislante térmico que se apoya sobre el forjado estructural. Seis pequeños ventiladores, situados entre estas cápsulas, harán circular el aire para extraer la energía.

El tipo de aire que moverán los ventiladores dependerá de la época del año. Si es verano será el aire fresco nocturno y si es invierno el aire interior, que previamente ha pasado por el invernadero arrastrando el calor aportado por el sol, y después por la vivienda recogiendo las cargas internas; en esa época, también el sol directo sobre el pavimento aportará calor directo a la primera capa de cápsulas.

Para controlar el flujo de aire exterior al interior de la casa se accionan los servomotores que controlan las rejillas de la zona Norte y Sur de la vivienda. De esta forma y en función de las reglas de control y los datos de temperatura obtenidos por los sensores, las trampillas se sitúan permitiendo el paso de aire de las zonas más calientes a las frías en invierno y de las más frescas a las más calientes en verano.

Para la aplicación de prueba se han establecido una serie de reglas de actuación a seguir para tratar de mantener la vivienda en unos márgenes climáticos óptimos. Esta serie de normas se pueden observar en la tabla 5.1

Temperatura Interior.	Temperatura Exterior	Temperatura de Geles	Acción	Estrategia
Frío	Frío	Frío	Se mantienen cerradas las rejillas. Se mantienen apagadas las turbinas.	Evitar que la temperatura interior baje más aún. Esperar fuentes de calor
Frío	Frío	Templado o Caliente	Se mantienen cerradas las rejillas. Se mantienen encendidas las turbinas.	Evitar la influencia del Frío exterior y utilizar el calor de los geles para calentar el interior.
Frío	Templado o Caliente	--	Se mantiene cerrada la rejilla sur. Se abre la rejilla norte y se encienden las turbinas.	Favorecer la entrada de calor del exterior y distribuirlo por la vivienda.
Templado	Frío	Frío o Templado	Se mantienen cerradas las rejillas. Se mantienen apagadas las turbinas.	Aislar la vivienda del exterior para evitar que la temperatura interior y de los geles baje a un estado Frío.
Templado	Frío o Templado	Caliente	Se mantienen abiertas las rejillas. Se mantienen encendidas las turbinas.	Favorecer la corriente de aire exterior para refrigerar la temperatura de los geles.
Templado	Templado	Templado	Se mantienen cerradas las rejillas. Se mantienen apagadas las turbinas.	Se mantiene la casa aislada conservando sus condiciones óptimas
Templado	Templado o Caliente	Frío	Se mantienen abiertas las rejillas. Se mantienen encendidas las turbinas.	Favorecer la corriente de aire exterior para aumentar la temperatura de los geles.
Templado	Caliente	Templado o Caliente	Se mantienen cerradas las rejillas. Se mantienen apagadas las turbinas.	Aislar la vivienda del exterior para evitar que la temperatura interior aumente más.
Caliente	Frío o Templado	--	Se mantiene cerrada la rejilla sur. Se abre la rejilla norte y se encienden las turbinas.	Favorecer la entrada de Frío del exterior y distribuirlo por la vivienda.
Caliente	Caliente	Frío o Templado	Se mantienen cerradas las rejillas. Se mantienen encendidas las turbinas.	Evitar la influencia del calor exterior y utilizar el Frío de los geles para enfriar el interior.
Caliente	Caliente	Caliente	Se mantienen cerradas las rejillas. Se mantienen apagadas las turbinas.	Evitar que la temperatura interior aumente más aún. Esperar fuentes de Frío

Tabla 5.1 Aplicación de Control del Sistema

5.4. Análisis de los Resultados de las Pruebas.

En este apartado se muestran los resultados de dos pruebas que se han realizado para verificar el funcionamiento del sistema. Como se ha explicado en el apartado 5.2 se ha monitorizado la temperatura y la humedad de los geles de cambio de estado, así como del interior y el exterior de la vivienda. También se ha monitorizado el estado de los distintos actuadores del sistema.

Para establecer los distintos estados de temperatura de la vivienda y así activar las correspondientes funciones de control de la aplicación (Ver Tabla 5.1), se han fijado como parámetros del sistema unos valores umbrales que delimitan el paso de frío a templado (23 °C) y de templado a caliente (25 °C).

5.4.1. Resultados Prueba nº1

Se han analizado más de 22000 medidas de cada uno de los 13 sensores instalados para la prueba. Estas medidas proporcionan la monitorización del estado de la vivienda en un período comprendido entre las 10 de la mañana del día 28 de Abril del 2007 a las 5 de la tarde del día 30 de Abril del 2007. No ha sido posible realizar medidas en un período más largo, debido a la necesidad de parar el sistema para acabar de poner a punto la instalación de la vivienda.

Los resultados obtenidos de estas medidas no han sido lo satisfactorios que se deseaba ya que las condiciones climatológicas del período del estudio y los días posteriores a éste, han sido completamente desfavorables para poder validar varias de las soluciones diseñadas para el control de la vivienda, aunque sí han permitido la verificación del funcionamiento del sistema de control.

Se puede apreciar en la Figura 5.2 como las temperaturas de las distintas zonas de la vivienda nunca han alcanzado una magnitud suficiente para cambiar la función de control del sistema. La vivienda ha permanecido por lo tanto siempre con las rejillas cerradas y los ventiladores apagados tratando de aislar en lo posible la vivienda del exterior y conservar al máximo la temperatura interior, se puede deducir por tanto que el sistema climático ofrecido por los expertos bioclimáticos necesita un aporte de sistemas activos como aire acondicionado o calefacción para situaciones extremas. (Ver conclusiones de Figura 5.2)

En la Figura 5.2 se muestran la media de la Temperatura Exterior, que es la que mayor variación sufre y la Temperatura Interior y la de los geles de cambio de estado.

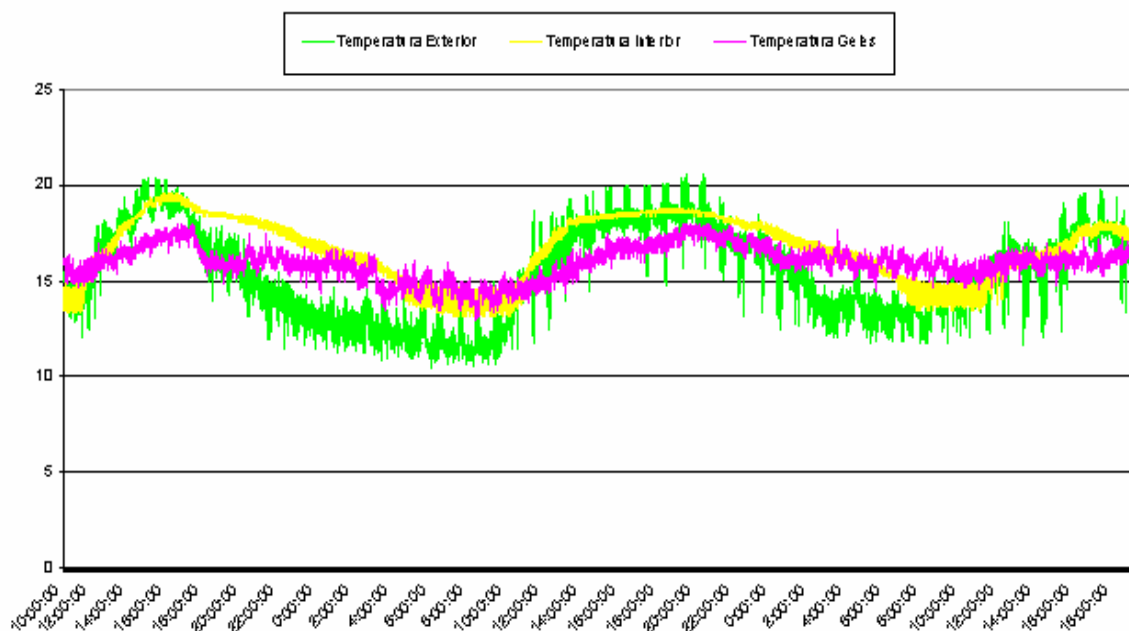


Figura 5.2 Resultados Temperatura Vivienda

Al no variar la función de control del sistema no se ha podido obtener una gráfica representativa de las distintas posiciones de los actuadores de nuestro sistema. Aún así se pueden deducir varias conclusiones de estas medidas:

- La temperatura de los geles es la más estable. Esto se debe a su inercia térmica.
- El frío (en nuestro ejemplo) se desplaza desde el exterior al interior, comparando las señales de temperatura exterior y temperatura interior se puede percibir cierto desfase entre ambas, así como una amplitud más reducida de la temperatura interior debido al aporte de los geles.
- Para una completa climatización de la vivienda especialmente en casos extremos no basta con el acondicionamiento térmico ambiental pasivo de la vivienda, se hace necesaria una aportación extra de algún sistema activo (aire acondicionado, calefacción, etc.)

La monitorización de la humedad sí que puede revelar un dato de cierto interés. En las fechas del estudio se produjo de manera accidental una pequeña inundación del ala oeste de la vivienda. A pesar de que la monitorización se realizó en un ala distinta a la afectada, se puede percibir la influencia de ese aumento de humedad en las graficas obtenidas.

La figura 5.3 muestra las medidas de humedad obtenidas durante las 55 horas del estudio. Se puede apreciar que debido a las lluvias los niveles de humedad son considerablemente altos. Analizando los resultados obtenidos entre las 12 de la mañana y las 2 de la tarde del día 30, se puede apreciar un considerable aumento de humedad, contrario a la tendencia normal que deberían seguir las medidas en las condiciones ambientales correspondientes a esas horas donde los efectos de la incidencia del sol son cada vez más notables.

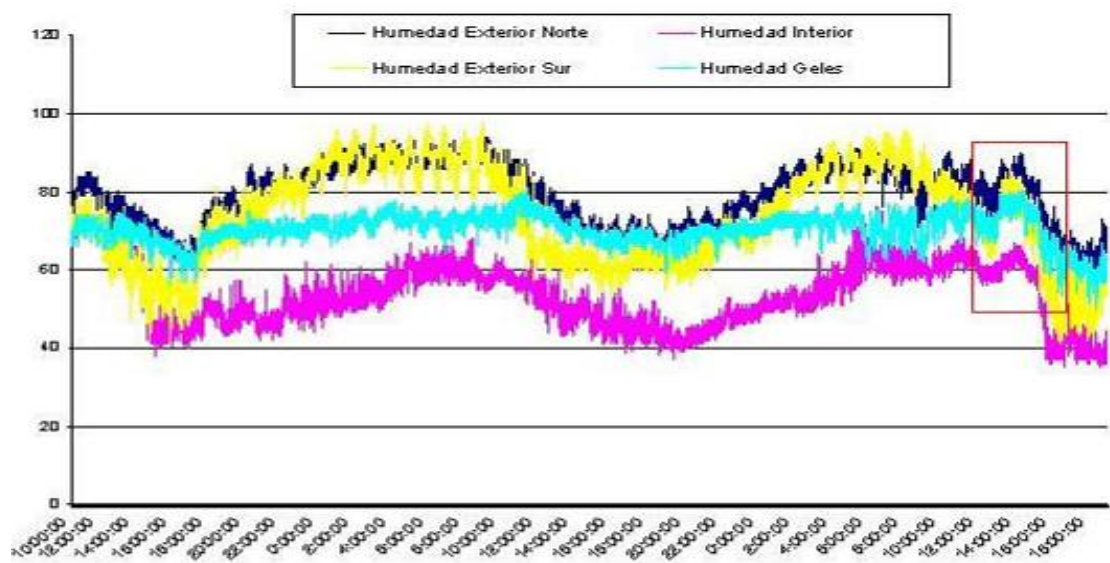


Figura 5.3 Resultados Humedad Vivienda

Para poder apreciar mejor este crecimiento inusual de la humedad de la vivienda debido a la incidencia comentada anteriormente, se puede realizar una comparación más en detalle en las Figuras 5.4 y 5.5 que corresponden respectivamente, a las medidas obtenidas en las mismas horas y en similares condiciones ambientales, para el día 30 que presenta la anomalía y el día 29 con una progresión normal de la humedad.

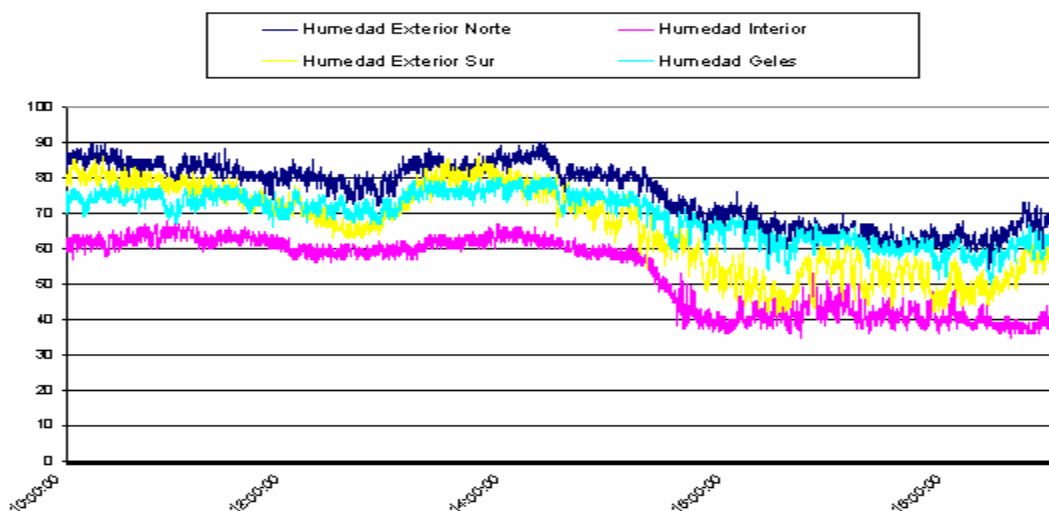


Figura 5.4 Resultados Humedad Vivienda Día 30

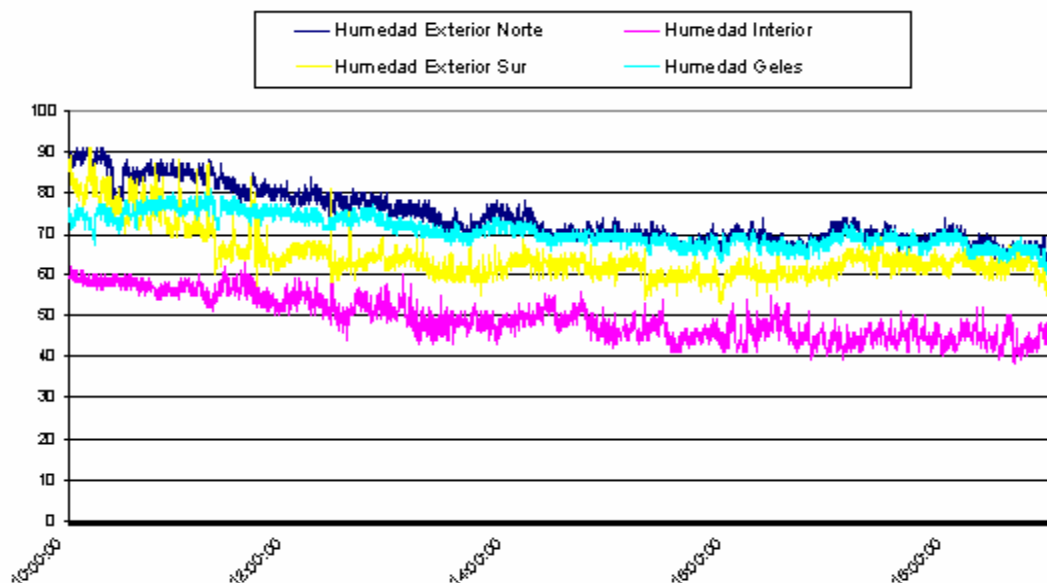


Figura 5.5 Resultados Humedad Vivienda Día 29

5.4.1. Resultados Prueba nº2

Debido a que la climatología durante el período en que se realizó la prueba nº 1 no era muy satisfactoria para demostrar el funcionamiento del sistema en condiciones próximas a las temperaturas óptimas de habitabilidad, consideradas en el diseño del sistema (entre 23°C y 25 °C), se ha realizado una segunda prueba con condiciones climatológicas más favorables para poder verificar la actuación del sistema sobre los dispositivos de control de la vivienda.

En esta segunda prueba se ha analizado una medición de más de 6000 muestras recogidas entre las 20 horas del día 7 de Mayo del 2007 y las 19 horas del día 8 de Mayo del 2007.

Los resultados obtenidos permiten monitorizar el estado de los distintos actuadores del sistema de control en función de las reglas bioclimáticas descritas en la Tabla 5.1

La Figura 5.6 muestra la media de la zona de invernadero sur y la zona de rejillas norte (Temperatura Exterior), la Temperatura Interior y la de los geles de cambio de estado durante todo el período de prueba.

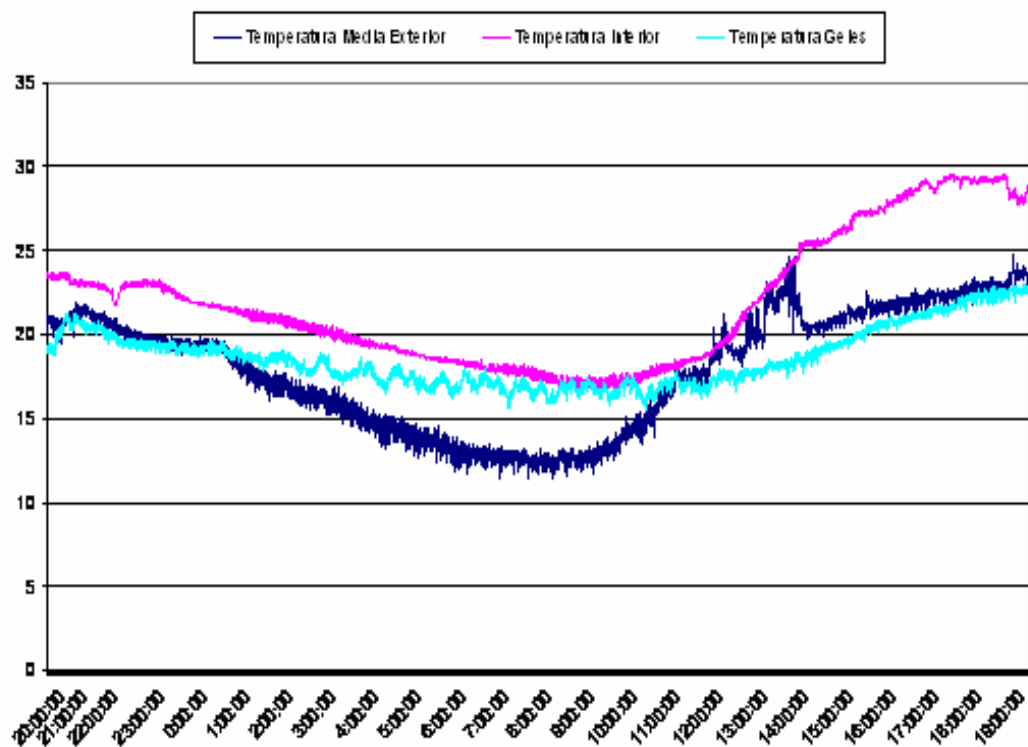


Figura 5.6 Resultados Temperatura Vivienda Prueba 2

De esta gráfica se pueden obtener varios datos relevantes. En primer lugar conviene señalar que la prueba se realiza tras varios días soleados de temperaturas medias-altas, este factor explica que la temperatura del interior de la vivienda esté en unos niveles óptimos del orden de 23°. La media de la temperatura exterior con la caída del día comienza a descender ejerciendo su influencia normal en la temperatura interior de la vivienda y como siempre más atenuada en los geles que continúan mostrándose más regulares gracias a su inercia térmica.

Llama la atención el importante aumento que sufren todas las temperaturas a partir de las 12 de la mañana, pero particularmente destaca el nivel de la temperatura interior claramente superior a la tendencia natural que debería seguir. Este crecimiento anómalo se debe a que durante ese periodo se realizaron en la vivienda una serie de pruebas del sistema fotovoltaico de Magic Box.

Para la realización de estas pruebas es necesario estudiar el comportamiento del sistema en función al consumo de una serie de electrodomésticos. Dado que la instalación de los mismos no ha sido finalizada para su correcto funcionamiento, se han utilizado una serie de resistencias a modo de carga simulando el consumo que llevarían acabo los electrodomésticos. Esta serie de resistencias son las que han aportado el exceso de calor que hace que el aumento de temperatura sea superior al normal. La potencia consumida en las resistencias (3000 Watios) es disipada en forma de calor en la vivienda provocando las temperaturas que se observan en la Figura 5.6

En las siguientes gráficas se puede observar el estado de cada uno de los actuadores durante el período de la toma de muestras. En estas figuras se puede comprobar como los actuadores en varias ocasiones han cambiado de posición accionando distintos sistemas de control de la vivienda.

La Figura 5.7 y 5.8 muestran el estado de los motores de las rejillas Norte y Sur respectivamente, el estado 0 corresponde a abierto y el estado 1 corresponde a cerrado.

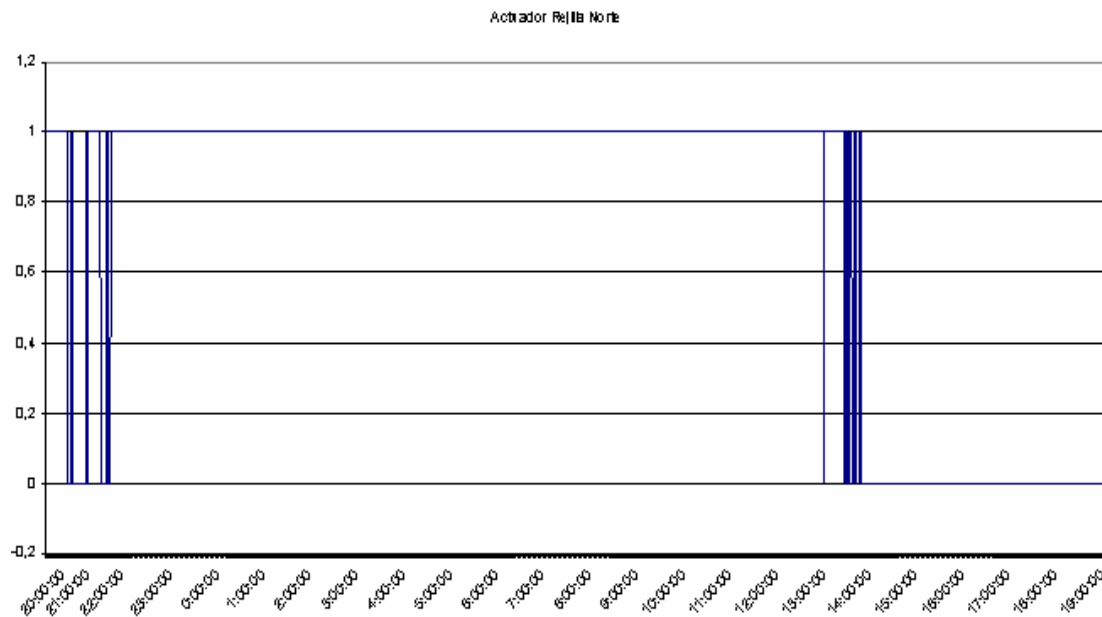


Figura 5.7 Resultados Actuador Rejilla Norte

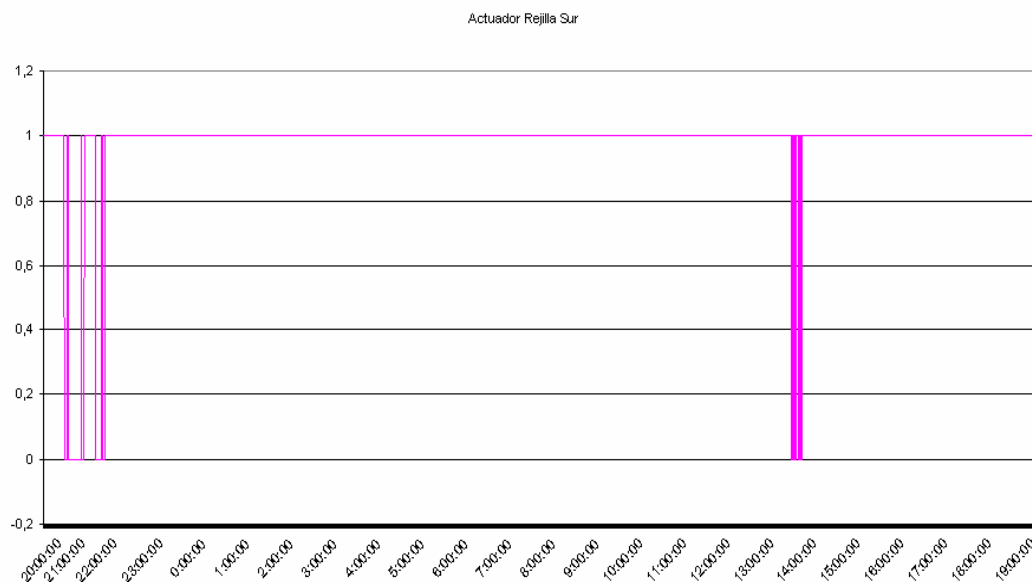


Figura 5.8 Resultados Actuador Rejilla Sur

La Figura 5.9 muestra el estado de las turbinas del suelo térmico donde se encuentran los geles. Al igual que en el caso de las rejillas el estado 0 es encendido y el estado 1 es apagado.

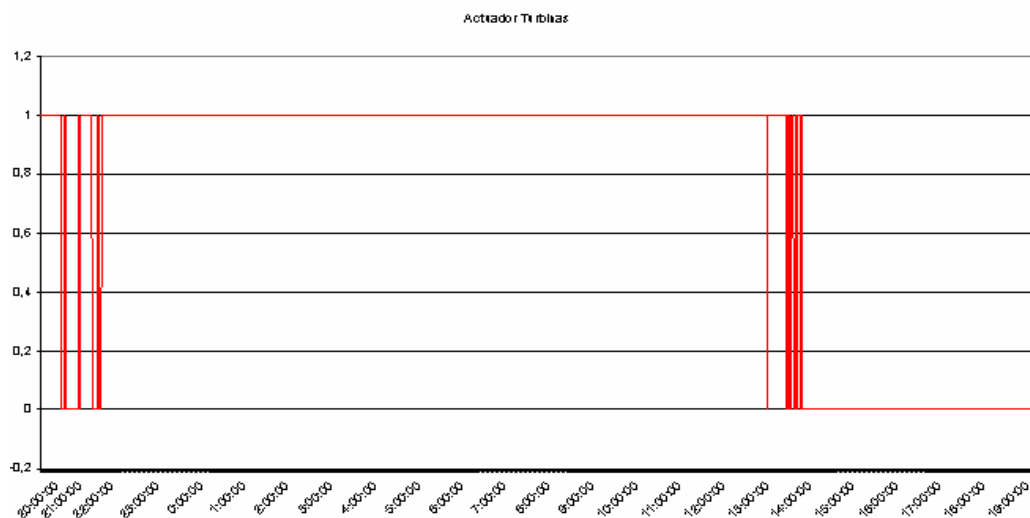


Figura 5.9 Resultados Actuador Turbinas

Comparando estas gráficas junto con la Figura 5.6 se puede verificar cómo cuando la casa se encontraba con temperaturas frías los 3 actuadores estaban apagados (valor 1) aislando a la casa del exterior y conservando en lo posible su temperatura. En cambio en los momentos en que las temperaturas estaban entorno a los valores óptimos se pueden observar variaciones en los actuadores provocadas por la ejecución de distintas funciones de control de nuestro sistema. Se puede observar también a partir de las 14:00 horas la temperatura en el interior de la casa es alta (mayor que 25°) mientras que la temperatura exterior y de los geles está entre templada y fría. Se comprueba en las Figuras 5.7, 5.8 y 5.9 que está abierta la rejilla del Norte y encendidas las turbinas pero con la rejilla sur cerrada para que la corriente más fría no salga por el suelo sino que entre en la casa y refrigere el interior, exactamente como indica en el noveno estado de la tabla 5.1.

Las Figuras 5.10, 5.11 y 5.12 muestran en mayor detalle las medidas tomadas en uno de los periodos de mayor numero de alteraciones en los estados de los actuadores, concretamente se presentan las muestras entre las 13:00 y las 14:00 horas de la rejilla Norte, Sur y las Turbinas respectivamente.

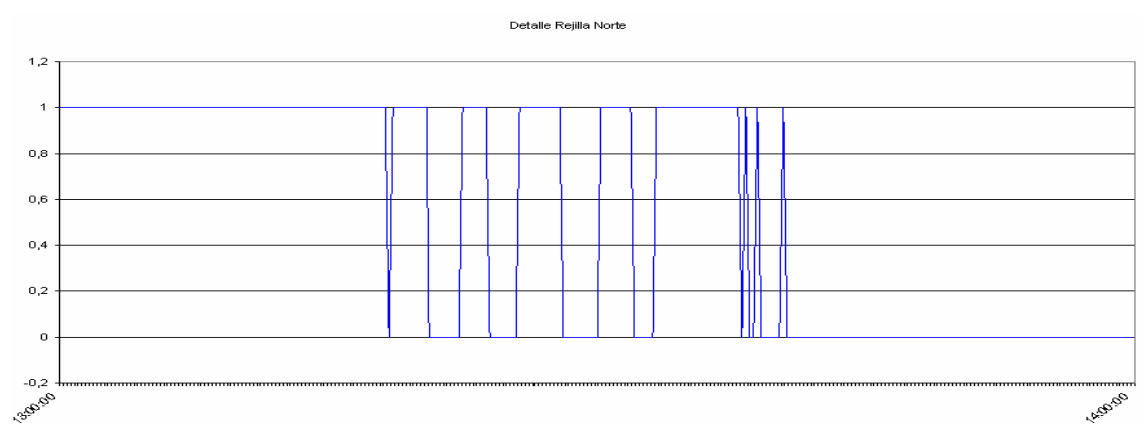


Figura 5.10 Resultados Detalle Rejilla Norte

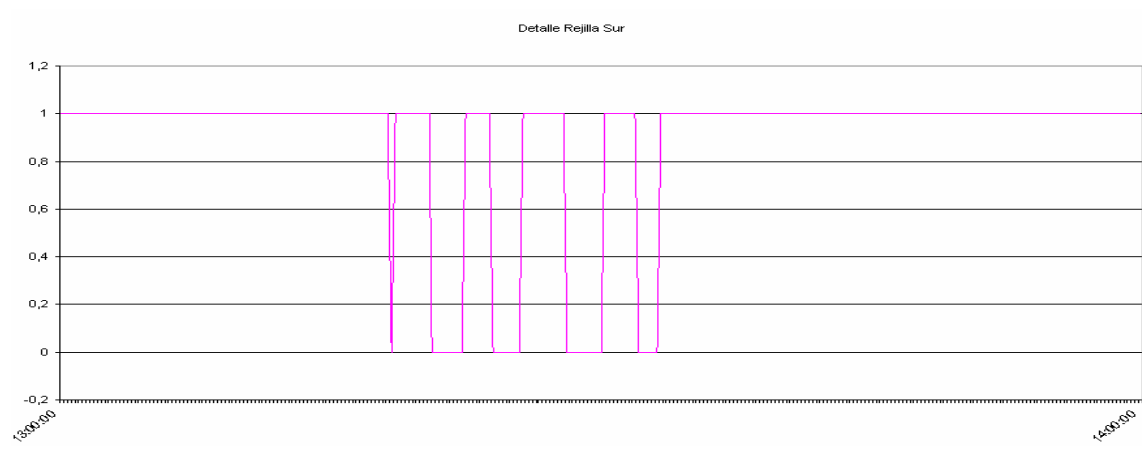


Figura 5.11 Resultados Detalle Rejilla Sur

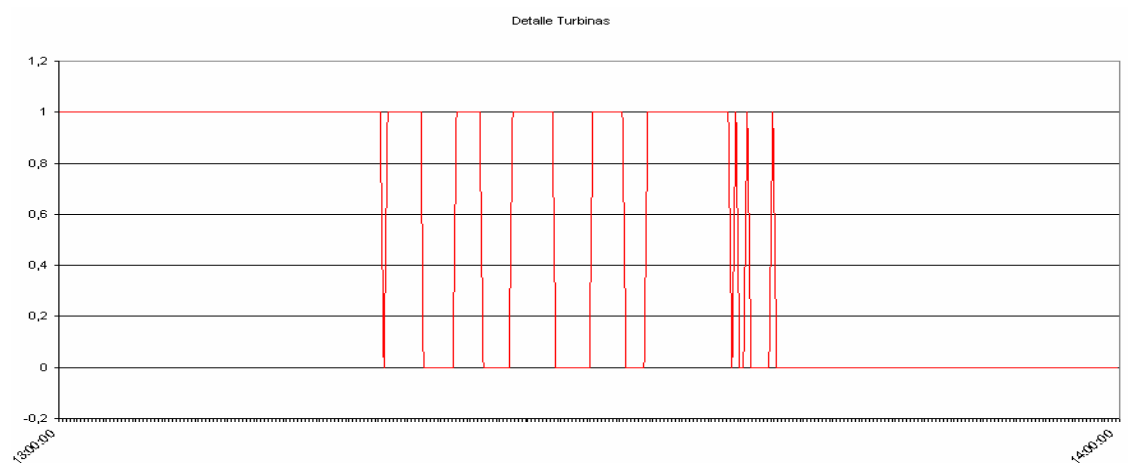


Figura 5.12 Resultados Detalle Turbinas

Capítulo 6

Conclusiones y Líneas Futuras

6.1. Introducción.

Una vez se han analizado con detalle los distintos elementos integrantes del sistema, y se ha explicado el funcionamiento y las conclusiones de la aplicación diseñada, se realiza, en este capítulo final, una revisión general del sistema implementado. Así mismo se aporta al lector una visión de las líneas futuras de expansión del sistema y otros desarrollos que se pueden realizar para aumentar su funcionalidad.

6.2. Conclusiones.

Se ha diseñado una aplicación, que permite gestionar la configuración y el funcionamiento del sistema de control de una vivienda bioclimática autosuficiente, **Magic Box**. Se ha modificado el sistema de control original de esta vivienda para dotarlo de nuevas mejoras y funcionalidades que hacen que el nuevo sistema de control sea **genérico, portable** a cualquier localización, **dinámico, abierto, modulable** y de **fácil expansión**.

Como se expuso en el **Capítulo 1** de este tomo, los principales objetivos propuestos al comienzo de este proyecto eran:

Analizar las características de los sensores y parámetros fundamentales para su configuración y funcionamiento.

- Configurar la comunicación entre los sensores, actuadores y nodos de nuestra red con el ordenador central desde el que ejerceremos el control del hogar.
- Configurar los elementos de comunicación entre el ordenador central que controla la red del hogar y el usuario que proporcionará las instrucciones de control que deben ser interpretadas y ejecutadas.

- Elaborar un software modulable y abierto que permita modificar los distintos elementos de la red de control con facilidad, de una manera intuitiva y dentro de lo posible sin necesidad de paralizar el sistema y de modificar y recompilar código.
- Incluir en la aplicación distintos sistemas de control de los diversos dispositivos. Permitir su selección, configuración y la posibilidad de incrementar el control a través de la introducción de nuevos módulos.

Estos objetivos se pueden resumir en dos aspectos fundamentales que constituyen la funcionalidad del sistema:

- Automatizar en lo posible la instalación, detección y configuración de la red de nodos y sensores empleados en Magic Box, teniendo en cuenta las limitaciones del Hardware existente.
- Facilitar la incorporación al código de distintas funciones de control a elección del usuario, de manera que puedan añadirse y seleccionarse distintos sistemas de control sin dificultad y sin paralizar el funcionamiento del sistema.

Repasando estos objetivos se puede observar que se ha logrado satisfacer con éxito todos ellos. La aplicación implementada otorga al usuario el absoluto control de la gestión del funcionamiento del sistema, permitiéndole realizar modificaciones en la configuración de la red de sensorización o en la parametrización del control del sistema, sin necesidad de detener la ejecución del mismo y reiniciar el sistema tras cada modificación. El software desarrollado se puede aplicar en todo tipo de entornos con distintas características de la red de sensorización, independientemente del tipo de hardware utilizado, ni del sistema operativo.

La tabla 6.1 muestra un resumen de las distintas tareas que conforman la funcionalidad del sistema y que realizan los subsistemas software de control y de aplicación.

SOFTWARE	TAREA	DESCRIPCIÓN
Control	Detección de Nodos	Envío a la lógica de Aplicación de las direcciones de los Nodos disponibles en la Red
Aplicación	Configuración de la Red	Configuración de los nodos y sensores de la estructura de Red por parte del usuario.
Control	Creación de la Red	Creación de una réplica virtual de la estructura de la Red física utilizada
Control	Seguimiento del Estado del Sistema	Realización y almacenamiento de sucesivas medidas para caracterizar el estado del Sistema
Aplicación	Configuración de los módulos de Control	Creación, Configuración y Borrado de módulos de control.
Aplicación	Activación de los módulos de Control	Visualización de las características de los módulos instalados. Selección de los módulos activos
Control	Ejecución del Sistema de Control	Interpretación y ejecución de las instrucciones de los módulos de control activos.
Aplicación	Visualización del Estado del Sistema	Diferentes monitorizaciones del estado de los elementos de la red de sensorización del sistema

Tabla 6.1 Resumen Funcionalidad del Sistema

Teniendo en cuenta que la realización de este proyecto forma parte de una labor de investigación que comenzó con la implementación de Magic Box y que continuará posteriormente a la conclusión de este proyecto. Sería de gran incomodidad que un nuevo investigador tuviera que analizar todo el código elaborado anteriormente para realizar nuevos cambios en el sistema. Para evitar esto se ha diseñado una estructura de software completamente modutable y dentro del siguiente apartado Líneas Futuras de Desarrollo se

describen los principales pasos a seguir para expandir el sistema implementado.

6.3. Líneas Futuras de Desarrollo.

A medida que este proyecto avanzaba en su desarrollo iban surgiendo nuevas ideas y posibilidades que incluir en la aplicación de control implementada. Muchas de estas ideas han sido implementadas completando cada vez más el sistema de control básico planteado inicialmente como objetivo, añadiendo más funcionalidades y proporcionando una aplicación más sencilla y cómoda de usar. Sin embargo el sistema desarrollado es sólo un paso más dentro de un amplio rango de evoluciones que pueden convertir lo que ahora es una sencilla aplicación en una potente herramienta de control.

El potencial de la aplicación con la que trabajamos es inmenso. Este proyecto se corresponde con las *“Mejoras en el motor de reglas bioclimático”* y *“Aplicaciones Web”* dentro de las líneas futuras de desarrollo del proyecto original [14]. Se ha proporcionado un sistema de control que permite trabajar con cualquier sistema operativo y en cualquier localización, que permite modificaciones de la red y la aplicación de control en caliente, “hot plug”, sin necesidad de paralizar el sistema, y a su vez se ha desarrollado una aplicación que permite al usuario manejar al sistema de control en todo momento. Pero sin alejarnos en exceso del estado actual del sistema se puede pensar en una aplicación que además del control ambiental, gestione sistemas de alarmas, el estado de elementos de almacenamiento de energía, incorpore distintos sensores de seguridad, que pueda ser manejada desde un teléfono móvil o una PDA o que proporcione una visualización en detalle al usuario del estado de los elementos de la vivienda por medio de mapas u otros elementos.

Uno de los principales objetivos de este proyecto era realizar un sistema lo más modulable posible para facilitar nuevas expansiones. Por lo tanto este aspecto ha sido cuidadosamente tratado en el desarrollo de todo el proyecto.

A continuación se detallan algunas de las principales líneas de expansión que podrían seguir completando el sistema implementado.

6.3.1. Expansión del sistema

La aplicación desarrollada tiene fijadas algunas limitaciones bien por las características del hardware, por eficiencia de tiempo, por diseño o escalabilidad del sistema.

Algunos elementos del sistema son muy sencillos de expandir, son aquellos que básicamente corresponden con la escalabilidad del sistema como el número de direcciones de nodos que se preguntan al bus para realizar la detección, las zonas a las que se puede asignar un nodo y la cantidad de sensores que la aplicación permite asignar a un nodo o a una función.

Dentro de la expansión del sistema hay dos elementos que requieren más elaboración para ser incluidos en la aplicación, nuevas funciones y nuevos tipos de sensores.

En el caso de que se deseen añadir nuevos tipos de sensor son necesarias diversas acciones:

- Incluir en el protocolo de comunicaciones el nuevo tipo de datos.
- Incluir el nuevo tipo de sensor en la función de reconocimiento de tipos del subsistema de control.
- Incluir el procesado de la petición del estado del nuevo sensor y la obtención de los datos correspondientes en el subsistema de control.
- Modificar en el subsistema de aplicación aquellas funciones que contengan los tipos de sensores.

EXPANSIÓN	UBICACIÓN	MODIFICACIÓN	LIMITACIÓN
Número de Nodos del Sistema	sdsp/lib/includes.h	modificar variable NUM_NODOS	Máximo 12 nodos
Número de Sensores en un Nodo	network/instalnode.php	ampliar opciones formulario 'numsensors'	Máximo de 7 sensores por nodo
Número de Sensores para una función de control	configcontrol/procesoperation.php	ampliar opciones formulario 'operación'	Máximo de 20 sensores por función
Tipos de Sensores	libraries/sensorlibrary.php	incluir nuevo tipo en 'showtype', 'getkind' y 'selecttype'	Sensores de Temperatura, Humedad, Luz y Relays
	libraries/modulelibrary.php	incluir nuevo tipo en 'kind'	
	sdsp/asking.c	Incluir en 'AskNode' el nuevo tipo de sensor. Añadir las modificaciones en el protocolo y las funciones necesarias para la comunicación con el bus.	
Zonas de los Nodos	libraries/nodelibrary.php	Modificar 'showzone', 'getzone' y 'selectzone'	Zonas de Magic Box
Nuevas funciones	libraries/controllibrary	incluir nombre nueva función en 'operation'	Funciones básicas de obtención de valor numérico, magnitud de sensores y media de sensores.
	configcontrol/procesoperation.php	añadir el procesado de la nueva función	
	sdsp/control.c	modificar 'getfunctionxml' para que reconozca la nueva función	
	sdsp/function.c	añadir el procesado de la nueva función	

Tabla 6.2 Resumen Expansión del Sistema

Si se desean añadir nuevas funciones de control al sistema también son necesarias varias tareas:

- Incluir en el subsistema de aplicación la nueva función en el formulario de selección de funciones.
- Incluir en el subsistema de aplicación el código necesario para obtener los datos que requiera la nueva función.
- Incluir en el reconocimiento de funciones del subsistema de control la nueva función a implementar.
- Añadir el procesamiento de la información de la nueva función para proceder a su evaluación y ejecución.

En la tabla 6.2 se muestra un resumen de los pasos necesarios para realizar todas estas expansiones.

6.3.2. Utilización de Smart Sensors

Uno de los principales objetivos planteados en este proyecto era automatizar la configuración de la red de control del sistema. Este objetivo sólo se ha podido cumplir parcialmente ya que por el hardware utilizado, la comunicación dentro de la red de control se realiza únicamente hasta el nivel de los nodos. Los sensores utilizados no poseen capacidad de comunicación, por lo tanto ha de ser el propio usuario el que introduzca en la aplicación la información sobre los sensores que se instalan en el sistema.

Para lograr una completa automatización del sistema sería necesario cambiar la red de manera que se pueda utilizar el estándar de sensores, IEEE 1451.4, [26] que aporta soluciones a todos estos problemas asociados a la configuración de sensores. Este estándar permite desarrollar sensores plug-and-play que poseen una interfaz de modo mixto que combina la señal analógica que refleja los fenómenos físicos (como la temperatura, presión y

fuerza) de los sensores tradicionales con una conexión digital serial para acceder a la hoja de datos electrónica del transductor (TEDS) incluida en el sensor. Esta memoria contiene la información binaria que identifica y describe al sensor y actuador. El TEDS contiene información como el fabricante, número de modelo del sensor, número de serie, rango de medición, sensibilidad, e información de calibración. Con TEDS, el sensor puede identificarse y describir su propio funcionamiento al sistema de adquisición de datos al cual está conectado.

La Figura 6.1 muestra un esquema de la interfaz mixta que constituye el corazón de los Smart Sensors IEEE 1451.4

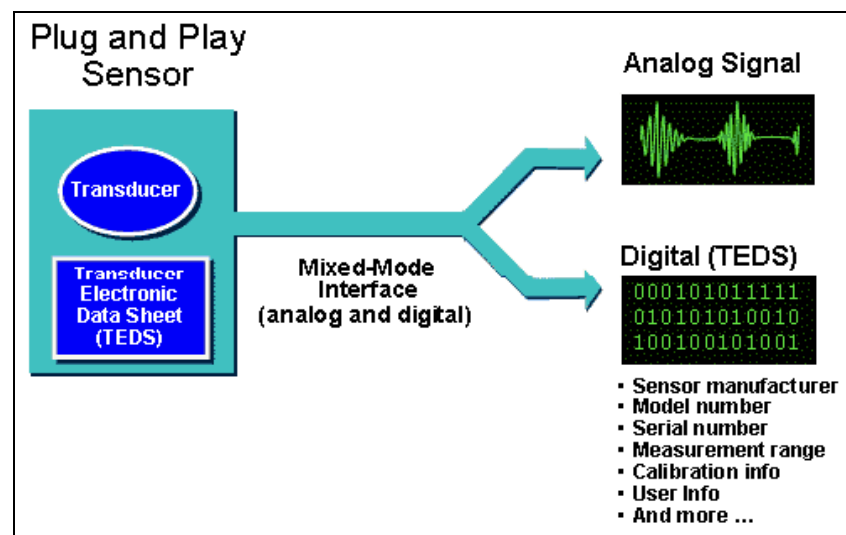


Figura 6.1 Smart Sensors IEEE 1451.4

Históricamente, cuando se ajustaba y configuraba un sistema de medición, se debía proporcionar de manera manual los parámetros importantes de los sensores de tal manera que el software convirtiera e interpretara apropiadamente los datos del sensor. Ahora, el sistema incorporado con los sensores inteligentes TEDS puede automatizar el paso de configuración, mientras incrementa de manera general la integración y fiabilidad del sistema.

Mientras otras tecnologías de sensores inteligentes también proporcionan la operación plug and play, IEEE 1451.4 es único ya que mantiene la salida análoga del sensor. Por lo tanto, los sensores inteligentes TEDS son compatibles con sistemas anteriores que incluyen interfases análogas tradicionales. Además, la simplicidad de las implementaciones IEEE 1451.4 tienen ventajas significativas y pragmáticas para usar los sensores anteriores. Los dos componentes principales del estándar son las hojas de datos estandarizadas (TEDS) y la interfaz de modo mixto.

La versión del estándar IEEE 1451.5 [27] pretende ejercer la misma función que los Smart Sensors de su versión anterior pero de forma inalámbrica, para ello pretende incorporar como capas físicas los principales protocolos inalámbricos de comunicación como Bluetooth/802.15.1, ZigBee/802.15.4 y WiFi/802.11. De esta manera se debería obtener la misma información del sensor inalámbrico implementando cualquiera de estos tres protocolos.

6.2.3. Utilización de una arquitectura Ethernet

Ethernet es una tecnología de redes de computadoras de área local (LANs) basada en tramas de datos bajo el estándar IEEE 802.3. Ethernet es la capa física más popular de la tecnología LAN usada actualmente, es tan popular porque permite un buen equilibrio entre velocidad, costo y facilidad de instalación. Estos puntos fuertes, combinados con la amplia aceptación en el mercado y la habilidad de soportar virtualmente todos los protocolos de red populares, hacen a Ethernet la tecnología ideal para la red de la mayoría de usuarios de la informática actual. Ethernet define las características de cableado y señalización de nivel físico y los formatos de trama del nivel de enlace de datos del modelo OSI.

Los elementos en una red Ethernet son los nodos de red y el medio de interconexión. Dichos nodos de red se pueden clasificar en dos grandes

grupos: Equipo Terminal de Datos (DTE) y Equipo de Comunicación de Datos (DCE). Los DTE son los dispositivos que generan o son el destino de los datos, tales como las computadoras personales, las estaciones de trabajo, los servidores de archivos, los servidores de impresión, todos son parte del grupo de estaciones finales. Mientras que los DCE son los dispositivos de red intermediarios que reciben y retransmiten las tramas dentro de la red, y pueden ser ruteadores, conmutadores (switch), concentradores (hub), repetidores, o interfaces de comunicación, como un módem o una tarjeta de interfase por ejemplo.

Se podría evaluar un posible cambio de la arquitectura actual basada en RS-485 a una arquitectura basada en Ethernet. Se pasaría de una estructura de bus con los nodos conectados en serie, (Ver Figura 6.2) a otra formada por uno o varios conmutadores a los que irían conectados los nodos (Figura 6.3), con lo cual se evitaría el problema que puede proporcionar la estructura en serie, en la cual, si falla un elemento del bus falla la comunicación con todos aquellos que se encuentran detrás de él. La capa de nivel de transporte TCP solucionaría los problemas que pudieran ocasionarse de colisiones de tramas en una arquitectura de red para Ethernet.

La migración a Ethernet proporcionaría ciertas ventajas:

- Uso de la tecnología LAN con más aceptación del mercado. Es un medio físico estandarizado en todo el mundo lo que facilita la obtención de los elementos de red que se necesiten.
- Ethernet presenta mayor capacidad de transmisión de información, gracias a su mayor velocidad, y que no está limitado en distancia ni en número de equipos que se pueden conectar.
- Identificación de los equipos mediante direcciones IP.
- Evita la inhabilitación de la red en caso de que uno de sus elementos falle.
- El uso de la capa TCP evita los problemas de colisión de tramas.

La mayor desventaja de un cambio a Ethernet radicaría en el incremento de complejidad que se introduciría en el sistema:

- Complejidad en el cableado, ya que sería necesario un cable de cada nodo a un conmutador y conectar también los conmutadores entre sí.
- Complejidad de comunicación ya que cada mensaje creado en un elemento de red llegaría a todos los componentes de la red siendo sólo aceptado por uno de ellos.



Figura 6.2 Arquitectura Bus-Serie RS-485

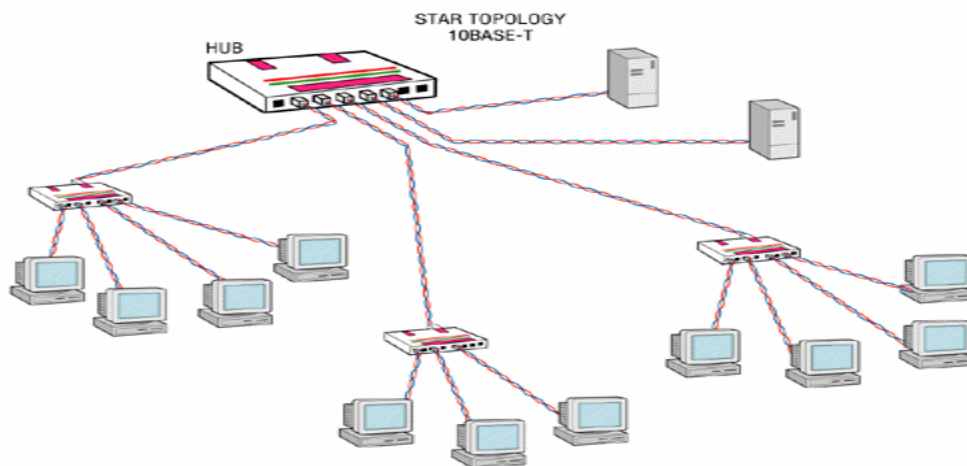


Figura 6.3 Arquitectura Estrella Ethernet

6.2.4. Utilización de una arquitectura Inalámbrica. Wi-Fi

Wi-Fi es un conjunto de estándares para redes inalámbricas basados en las especificaciones **IEEE 802.11**. Creado para ser utilizado en redes locales inalámbricas, es frecuente que en la actualidad también se utilice para acceder a Internet.

La norma IEEE.802.11 fue diseñada para sustituir a las capas físicas y MAC de la norma 802.3 (Ethernet). Esto quiere decir que en lo único que se diferencia una red Wi-Fi de una red Ethernet, es en la forma como los ordenadores y terminales en general acceden a la red. Por tanto una red local inalámbrica 802.11 es completamente compatible con todos los servicios de las redes locales de cable 802.3 (Ethernet).

La principal ventaja que aporta emplear una arquitectura de red inalámbrica es la gran comodidad de no necesitar interconectar la red por cable y la movilidad que esto permite. Pero existen muchos aspectos que conviene tener en cuenta antes de decidir un cambio a una arquitectura inalámbrica.

- Seguridad. Ante la capacidad de terceras personas para conectarse a redes ajenas es necesario utilizar medidas como la encriptación de datos para garantizar la seguridad de estas redes.
- Pérdida de velocidad en relación a la misma conexión utilizando cables, debido a las interferencias y pérdidas de señal que el ambiente puede acarrear.
- Autonomía de los elementos de la red. Dado el alto consumo energético son necesarias pilas y baterías para alimentar los equipos que se consumen con rapidez.

6.2.5. Utilización de una arquitectura Inalámbrica. Bluetooth.

Bluetooth es el nombre común de la especificación industrial **IEEE 802.15.1**, que define un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura, globalmente y sin licencia de corto rango [28].

Bluetooth es una tecnología orientada a sistemas de comunicación a corta/media distancia y optimizados para un bajo coste y menor consumo que otras tecnologías inalámbricas como WiFi, posicionándose como la tecnología del futuro para pequeñas redes o sistemas de captación de información. Los dispositivos Bluetooth requieren poca potencia y tienen como finalidad transmitir pequeñas cantidades de datos a casi 1Mbps en cortas distancias.

Una de las principales ventajas del uso de la tecnología inalámbrica Bluetooth es su estandarización. Bluetooth utiliza la banda de radio ISM de 2.4 GHz. El uso de una banda común de frecuencias significa que se pueden emplear dispositivos que utilicen la especificación Bluetooth en cualquier parte del mundo, y serán capaces de enlazar con otros dispositivos similares, independientemente del país en que se encuentren.

La conexión entre dispositivos Bluetooth sigue un esquema maestro-esclavo, pero uno de los aspectos más críticos para su utilización es que como máximo soporta ocho dispositivos conectados simultáneamente en lo que es la estructura básica de comunicación (piconet).

Estableciendo una comparativa frente a una arquitectura WiFi presenta como ventajas un menor coste y un menor consumo pero por el contrario presenta mayores limitaciones en cuanto al número de elementos de la red, velocidad y longitud de la transmisión.

6.2.6. Utilización de una arquitectura Inalámbrica. ZigBee.

ZigBee es un protocolo de comunicaciones inalámbrico, similar a Bluetooth, y basado en el estándar para redes inalámbricas de área personal (WPANs) **IEEE_802.15.4**. Surge del fruto de una alianza de más de 200 empresas con el objetivo de conseguir el desarrollo e implantación de una tecnología inalámbrica de bajo coste [29].

Esta tecnología encuentra su mayor difusión en el campo de la domótica debido a su bajo consumo, su sistema de comunicaciones vía radio (con topología MESH) y su fácil integración (se pueden fabricar nodos con muy poca electrónica).

ZigBee se presenta como una solución más favorable que Bluetooth para aquellas aplicaciones que no requieran traspasos de gran cantidad de información o de largas distancias y en las que se desee disminuir en lo posible el consumo de los dispositivos. Además ZigBee permite configurar redes de hasta 64000 nodos ampliamente superior a los 8 de la tecnología Bluetooth.

Uno de los aspectos más críticos a la hora de evaluar la conveniencia de la implementación de una arquitectura ZigBee, aparte de las limitaciones de longitud y velocidad de transmisión, es la falta de estandarización que presentan los distintos fabricantes de dispositivos de esta tecnología. Con lo cual el diseño de cada sistema puede depender de las características de los distintos fabricantes.

Bibliografía

Bibliografía.

- [1] JOSÉ JIMÉNEZ DELGADO. DIRECTOR INNOVACIÓN TELEFÓNICA I+D.
Estado actual y perspectivas de investigación y desarrollo del "Hogar Digital".
Foro UPM-Empresa. Jornada sobre el Hogar Digital.
- [2] ASOCIACIÓN ESPAÑOLA DE DOMÓTICA (CEDOM).
URL <http://www.cedom.org>.
- [3] BERNARDO ALMONACID CRUZ.
Situación actual y evolución de la Domótica. Aspectos Generales.
1er Congreso Nacional de Arquitectura y Domótica.
- [4] Definición desarrollada por Asimelec y Acceda para la presentación "*La ICT y el Hogar Digital*" del V Congreso Internacional de Empresas Instaladoras de Comunicaciones.
- [5] RAFAEL COOMONTE BELMONTE.
Introducción al Hogar Digital.
Foro UPM-Empresa. Jornada sobre el Hogar Digital.
- [6] ESTEFANÍA CAAMAÑO MARTÍN, JAVIER NEILA GONZÁLEZ, FCO. JAVIER JIMÉNEZ LEUBE, MIGUEL ÁNGEL EGIDO AGUILERA, MARÍA UZQUIANO, JOSÉ MIGUEL GÓMEZ OSUNA, CÉSAR BEDOYA FRUTOS, LUÍS MAGDALENA LAYOS, ALFONSO GARCÍA SANTOS.
Viviendas Solares Autosuficientes: Participación de la Universidad Politécnica de Madrid en el concurso "Solar Decathlon".

[7] Concurso Solar Decathlon.

URL www.solardecathlon.org

[8] ÁLVARO GUTIÉRREZ MARTÍN.

Bus domótico: Sistema de Control y monitorización.

Tech. Rep., Equipo UPM para el Solar Decathlon, 2004.

[9] ÁLVARO GUTIÉRREZ MARTÍN Y DANIEL MARTÍN GÓMEZ.

Aplicaciones domóticas.

Tech. Rep., Equipo UPM para el Solar Decathlon, 2004.

[10] ÁLVARO GUTIÉRREZ MARTÍN Y DANIEL MARTÍN GÓMEZ.

Sistemas domóticos: EIB vs. PLC.

Tech. Rep., Equipo UPM para el Solar Decathlon, 2004.

[11] ÁLVARO GUTIÉRREZ MARTÍN Y DANIEL MARTÍN GÓMEZ.

Sistemas domóticos: Lonworks.

Tech. Rep., Equipo UPM para el Solar Decathlon, 2004.

[12] ÁLVARO GUTIÉRREZ MARTÍN Y DANIEL MARTÍN GÓMEZ.

Posibilidades y necesidades de buses domóticos.

Tech. Rep., Equipo UPM para el Solar Decathlon, 2004.

[13] ÁLVARO GUTIÉRREZ MARTÍN.

Control Domótico de una vivienda sostenible basado en RS-485.

Tech. Rep., Equipo UPM para el Solar Decathlon, 2004.

[14] DANIEL MARTÍN GÓMEZ.

Proyecto Fin de Carrera: *Diseño e implementación de una plataforma del hogar digital para una vivienda sostenible.*

[15] Equipo UPM para el Solar Decathlon, 2004.

Design and Specifications of the Monitoring and Control System.

[16] BYRON GOTTFRIED.

Programación en C.

Editorial Mc Graw Hill.

[17] BRIAN W. KERNIGHAN, DENNIS M. RITCHIE.

El Lenguaje de Programación C.

Editorial Prentice Hall.

[18] JAVIER GARCÍA DE JALÓN DE LA FUENTE, JOSÉ IGNACIO RODRÍGUEZ GARRIDO, RUFINO GOÑI LACERAS, ALFONSO BRAZÁLEZ GUERRA, PATXI FUNES MARTÍNEZ, RUBÉN RODRÍGUEZ TAMAYO.

Aprenda lenguaje ANSI C como si estuviera en Primero.

Escuela Superior de Ingenieros Industriales. Universidad de Navarra

[19] STEPHEN C. DEWHURST, KATHI T. STARK.

Programming in C++.

Editorial Prentice Hall.

[20] JAVIER GARCÍA DE JALÓN, JOSÉ IGNACIO RODRÍGUEZ, JOSÉ MARÍA SARRIEGUI, RUFINO GOÑI, ALFONSO BRAZÁLEZ, PATXI FUNES, ALBERTO LARZABAL, RUBÉN RODRÍGUEZ.

Aprenda lenguaje C++ como si estuviera en Primero.

Escuela Superior de Ingenieros Industriales. Universidad de Navarra.

[21] JACK STURM

Desarrollo de Soluciones XML.

Editorial McGraw-Hill.

[22] Página Web Especificación XML

URL <http://www.w3.org/TR/WD-xml-lang#sec1.1>

[23] LUKE WELLING, LAURA THOMSON.

PHP and MySQL Web Development.

Editorial Anaya.

[24] Página Web PHP

Manual PHP5

URL www.php.net

[25] Página Web MySQL

URL www.mysql.com

[26] Página Web National Instruments TEDS

URL <http://www.ni.com/teds>

[27] Página Web IEEE 1451

URL <http://ieee1451.nist.gov>

[28] FRANCISCO JOSÉ BELLIDO OUTEIRIÑO, JOSÉ LUÍS DE LA CRUZ FERNÁNDEZ,

MANUEL TORRES ROLDÁN Y JOSÉ ANTONIO GISTAS PEYRONA

Comunicación inalámbrica con Bluetooth

[29] Artículos Web ZigBee

URL <http://netandtech.wordpress.com/zigbee/wireless-con-zigbee/>

URL <http://www.domodesk.com/content.aspx?co=97&t=146&c=43>

Apéndice A
Manual de Usuario

A.1. Introducción.

Este apéndice trata de servir de guía para la instalación y manejo del sistema desarrollado, bien para usuarios de la aplicación o para nuevos desarrolladores. En el siguiente apartado se analizarán los distintos componentes que son necesarios instalar en un sistema Linux para la correcta compilación y ejecución del sistema de control y para la configuración básica de la arquitectura cliente servidor de la aplicación Web. El apéndice finaliza con un apartado que detalla el manejo de la aplicación que gestiona el sistema.

A.2. Instalación.

Aunque el sistema desarrollado podría ejecutarse también sobre un sistema Windows, se ha optado por implementarlo en un entorno Linux. Para ello es necesario tener una distribución de Linux instalada en el ordenador que ejercerá el control de todo el sistema (la aplicación actual está desarrollada en Ubuntu). El primer paso a realizar es habilitar los repositorios **UNIVERSE** y **MULTIVERSE** en `/etc/apt/sources.list` y actualizar los repositorios del sistema (con `aptitude update`).

Para que el sistema funcione correctamente son necesarios los siguientes pasos de instalación:

- Instalar el paquete GCC
- Instalar el paquete apache2
- Instalar el paquete php5
- Instalar el paquete libapache2-mod-auth-mysql
- Instalar el paquete php5-gd
- Instalar el paquete php5-pgsql
- Instalar el paquete php5-mysql

- Instalar el paquete `php5-xmlrpc`
- Instalar el paquete `make`
- Instalar el paquete `g++`
- Instalar el paquete `libxmlrpc-c3-dev`
- Instalar el paquete `xml2-dev`

Una vez se han instalado estos paquetes el sistema de control debería compilar y ejecutar correctamente. Para la aplicación Web aún es necesario reiniciar los servicios de `apache` y `mysql` (`/etc/init.d/apache2 restart` y `/etc/init.d/mysql restart`), y crearse un usuario en `Mysql` con los permisos habilitados, para que permita el acceso a las bases de datos que maneja la aplicación Web.

A.3. Manejo del Sistema.

En este apartado se analizará con detalle el funcionamiento de la aplicación para facilitar su manejo por nuevos usuarios o desarrolladores.

El sistema está diseñado para que el subsistema de control esté constantemente en ejecución, por lo tanto su funcionamiento es transparente para el usuario y no se explicará aquí de nuevo. Para más información sobre el funcionamiento del sistema de control se puede revisar el capítulo 3.

La aplicación cuenta con un menú principal (Ver Figura A.1) que muestra al usuario diferentes opciones para actuar con el sistema. Se presentan al usuario cuatro opciones de actuación:

- **Configuración de la Red.**
- **Configuración de los Módulos de Control.**
- **Activación de los Módulos de Control.** Esta opción sólo aparece si existe algún módulo instalado.

- **Monitorización del estado del sistema.** Esta opción se encuentra habilitada únicamente cuando se han realizado y almacenado medidas instantáneas.

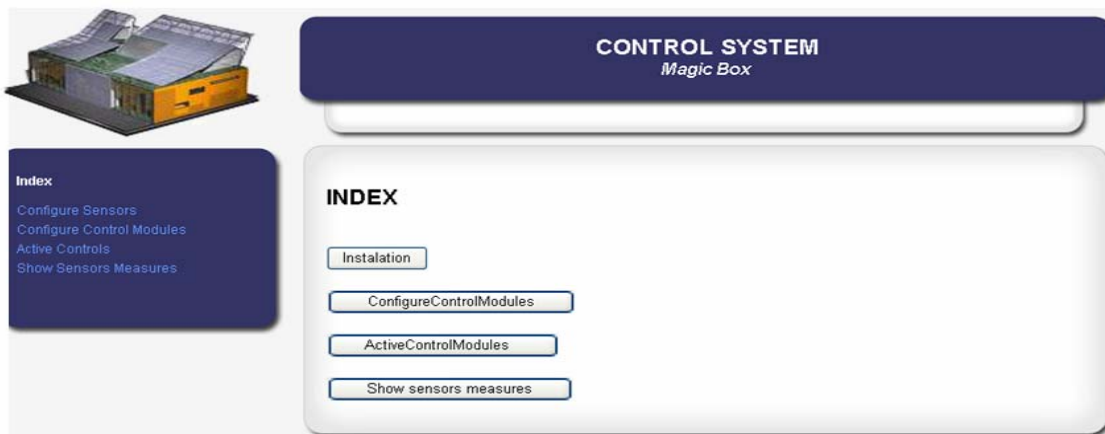


Figura A.1 Menú Principal de Aplicación

A.3.1. Configuración de la Red

En este apartado se detalla el funcionamiento de la configuración de los nodos y los sensores de la red de control del sistema

Cuando el usuario desea configurar la red se realiza una detección automática de los nodos instalados en la red física del sistema. La aplicación permanece en espera hasta que la detección automática concluye.

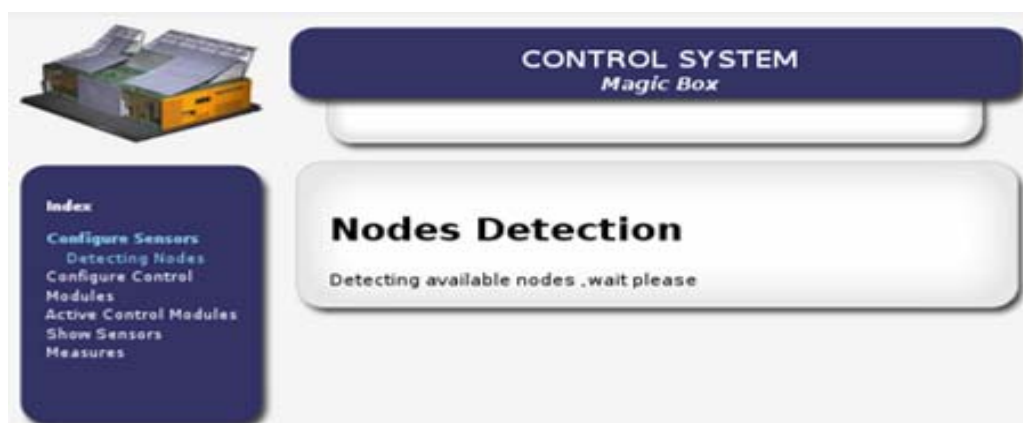


Figura A.2.1 Detección de Nodos.

Cuando acaba la detección la aplicación notifica al usuario qué nodos detectados son nuevos y se pueden instalar y qué nodos están ya configurados permitiendo eliminarlos. (Ver Figura A.2.2)

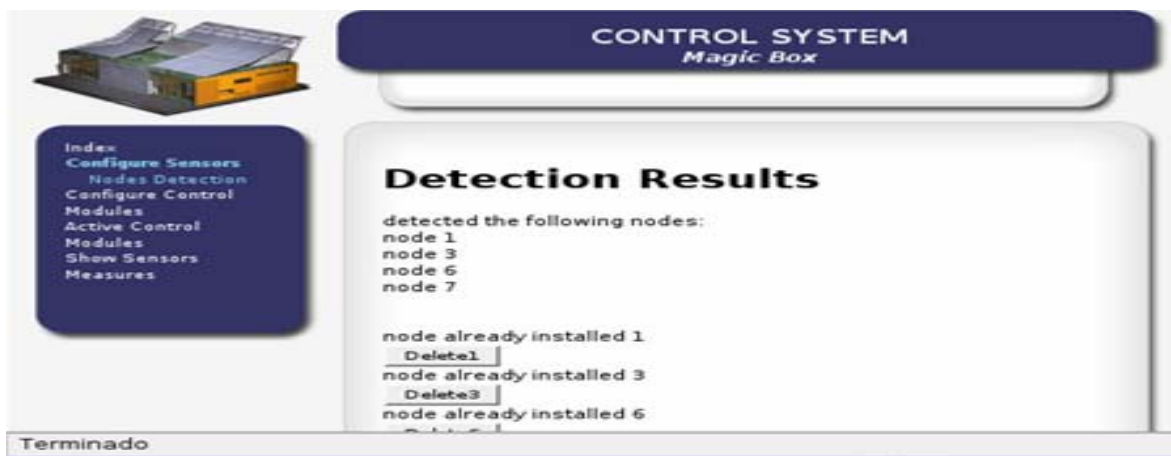


Figura A.2.2 Resultado Detección de Nodos.

A.3.1.1. Instalación de nodos y sensores

Si se decide instalar un nodo, la aplicación solicita que se seleccione la zona del nodo que se ha seleccionado instalar y el número de sensores que se desean configurar en el nodo.

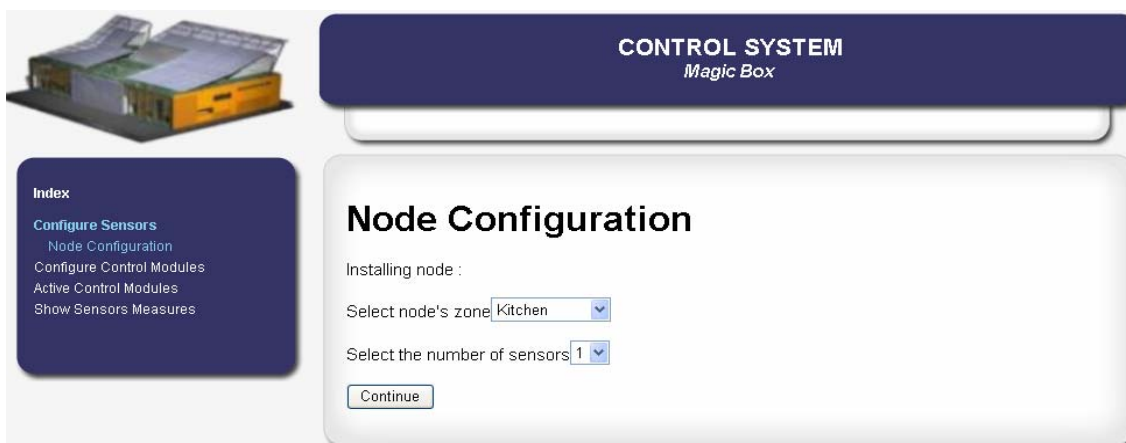
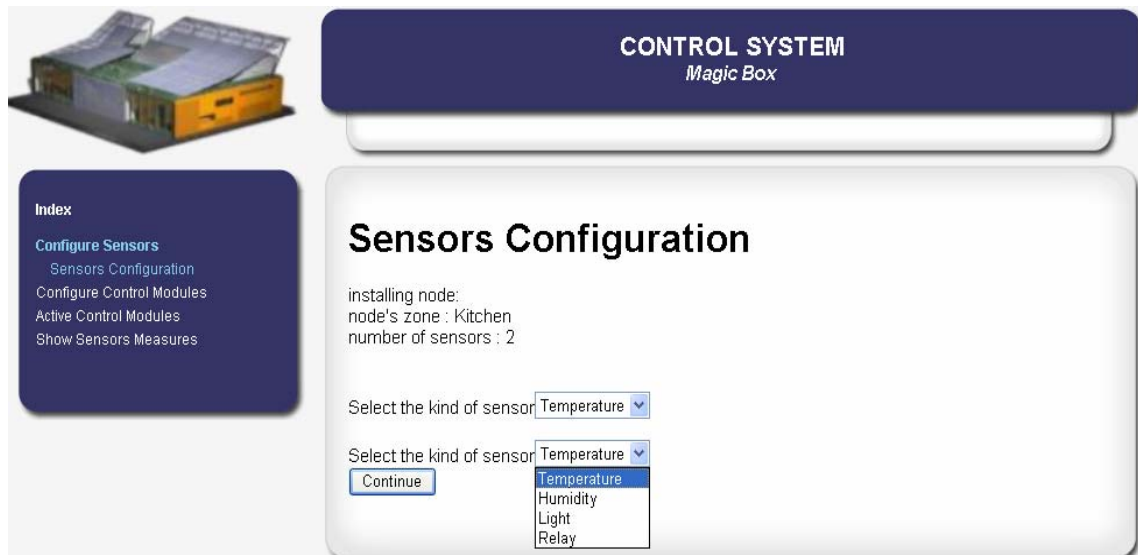


Figura A.2.3 Configuración de Nodos.

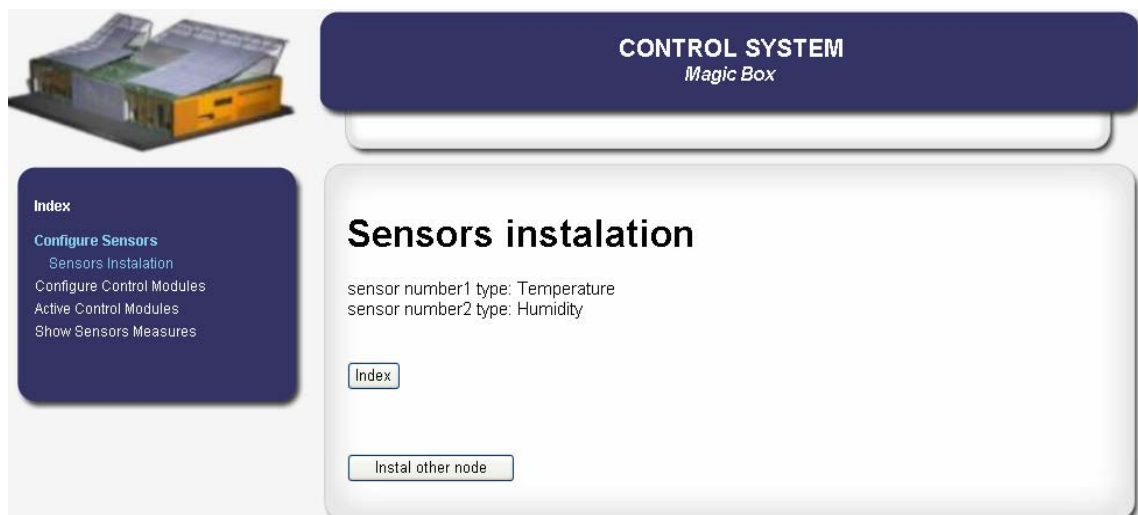
Finalizada la introducción de los datos de los nodos se procede a la instalación de los sensores indicando el tipo de cada uno de ellos.



The screenshot shows a web application titled "CONTROL SYSTEM Magic Box". On the left is a sidebar menu with an "Index" section containing links: "Configure Sensors", "Sensors Configuration", "Configure Control Modules", "Active Control Modules", and "Show Sensors Measures". The main content area is titled "Sensors Configuration" and displays the following information: "installing node:", "node's zone : Kitchen", and "number of sensors : 2". There are two dropdown menus, both labeled "Select the kind of sensor", each with "Temperature" selected. A "Continue" button is located below the first dropdown. A second dropdown menu is open, showing options: "Temperature", "Humidity", "Light", and "Relay".

Figura A.2.4 Configuración de Sensores.

Cuando concluye la instalación del nodo y sus respectivos sensores el usuario puede volver a la configuración de nodos o volver al menú principal.



The screenshot shows the same web application, now titled "Sensors installation". The sidebar menu remains the same. The main content area displays: "sensor number1 type: Temperature" and "sensor number2 type: Humidity". There is an "Index" button and an "Install other node" button at the bottom.

Figura A.2.5 Resultado Instalación de Sensores.

A.3.1.2. Desinstalación de nodos y sensores

En caso de que se desee eliminar algún nodo de la red, antes de proceder a borrar la información del nodo se solicita al usuario la verificación de su deseo de eliminar dicho nodo. Si continúa se eliminará toda la información del nodo, si se cancela la eliminación se vuelve al índice principal.

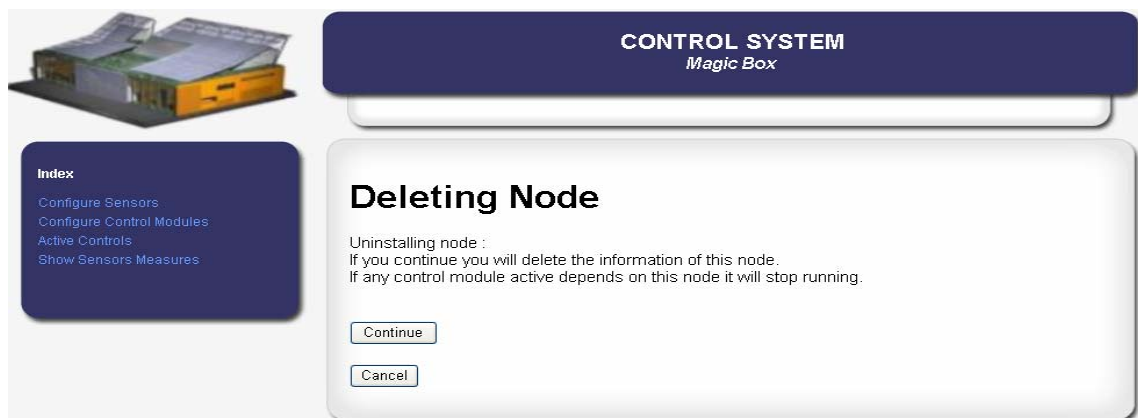


Figura A.2.6 Eliminación de Nodos.

A.3.2. Configuración de los módulos de control

A continuación se explicará el proceso para configurar un módulo de control correctamente. El proceso es secuencial y puede resultar algo complejo en una primera toma de contacto. A lo largo de la configuración de un módulo existe un importante flujo de información, con lo cual es aconsejable que en el caso de que se interrumpa la configuración antes de tiempo o se cometa un error en algún paso se considere ese módulo como incorrecto, se elimine de la lista de módulos configurados y se repita la instalación.

Al seleccionar Configurar Módulos de Control en el Índice Principal (Figura A.1) la aplicación permite instalar o desinstalar módulos de control. Primeramente se analizará la configuración y se concluirá este apartado con el proceso de borrado de módulos.



Figura A.3.1 Configuración de Módulos de Control.

A.3.2.1. Instalación de módulos de control

La primera ventana de navegación para la instalación de un módulo solicita al usuario el nombre que identificará al módulo y una breve descripción de su funcionamiento.

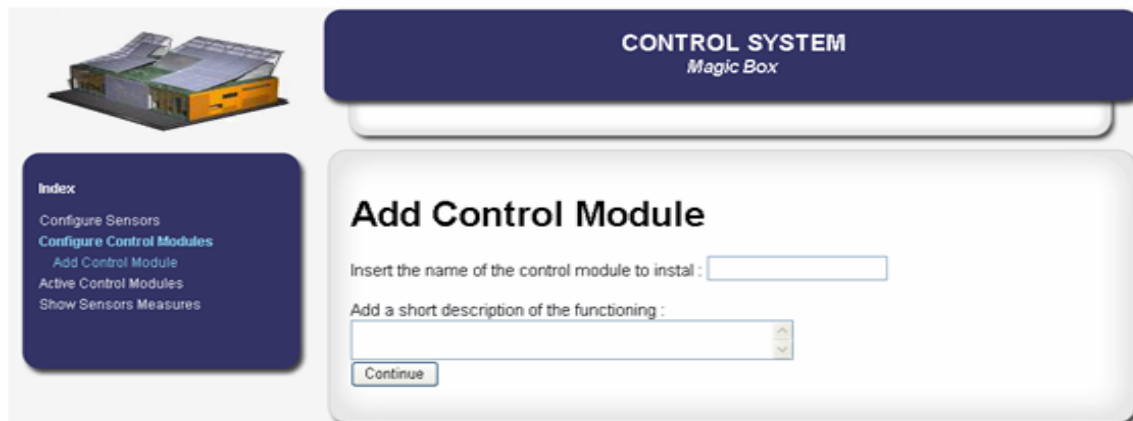


Figura A.3.2 Instalación de un Módulo de Control.

La siguiente ventana solicita el tipo de operación que se desea introducir en la condición. En principio se pueden seleccionar el valor de un sensor o la función media de una serie de sensores.



Figura A.3.3 Selección del tipo de operación.

En caso de que la opción elegida sea un sensor se muestran todos los sensores del sistema para que el usuario seleccione el que desee. (Ver Figura A.3.4)

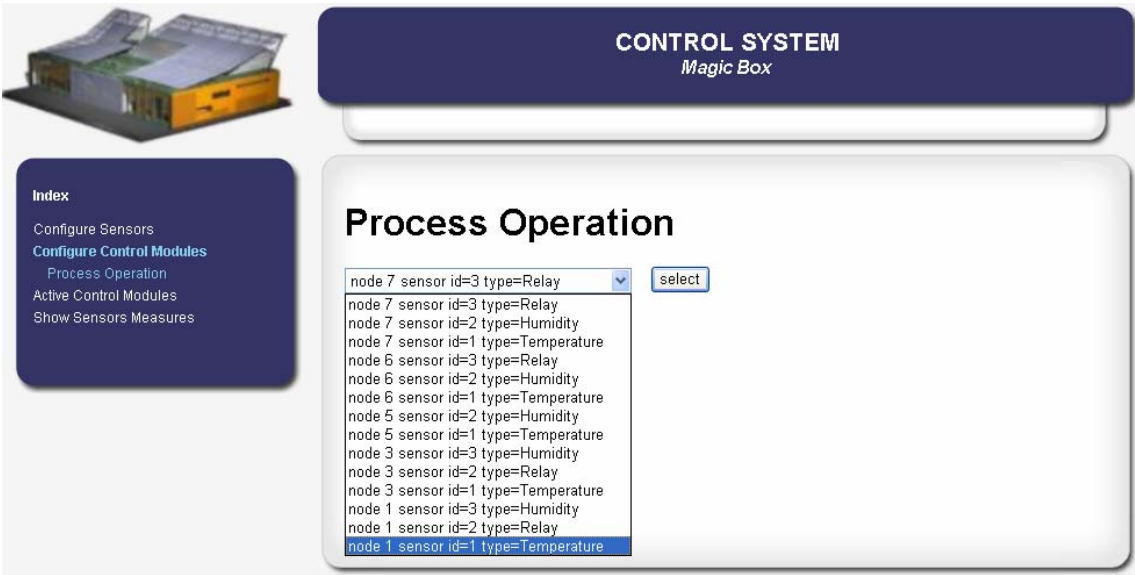


Figura A.3.4 Selección operación tipo sensor.

En el caso de que se decida realizar la media de varios sensores se selecciona la opción average en la Figura, a continuación la aplicación solicita el número de sensores sobre los que realizar la media aritmética.

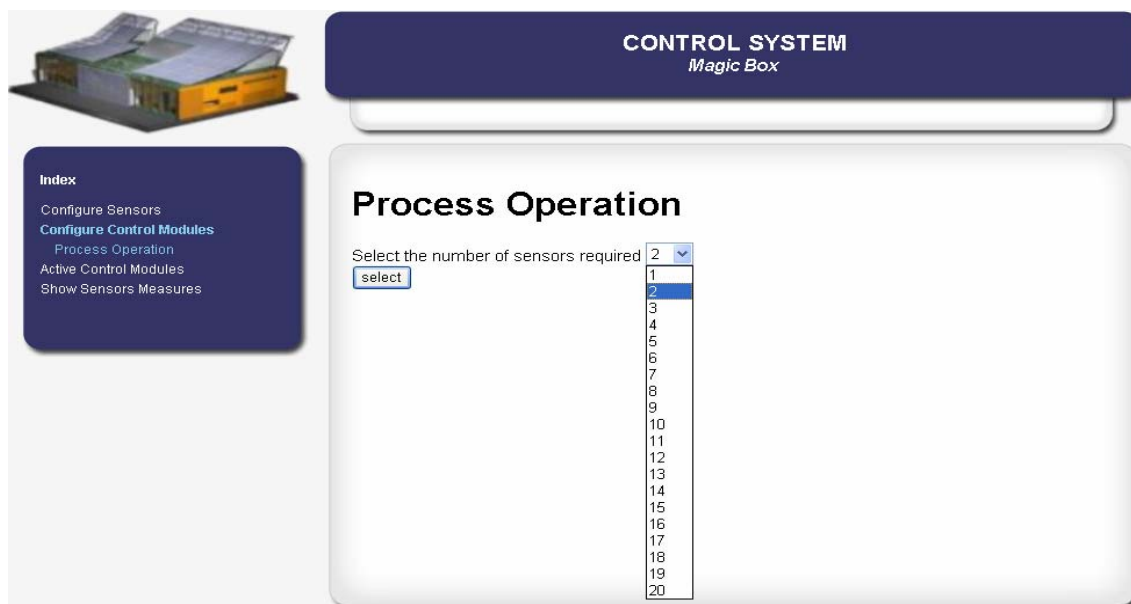


Figura A.3.5 Selección número de sensores para operación media.

Tras seleccionar el número de sensores de la función media la aplicación muestra tantos paneles de selección de sensores como sean necesarios para completar la función media que se desee utilizar. (Ver Figura A.3.6)

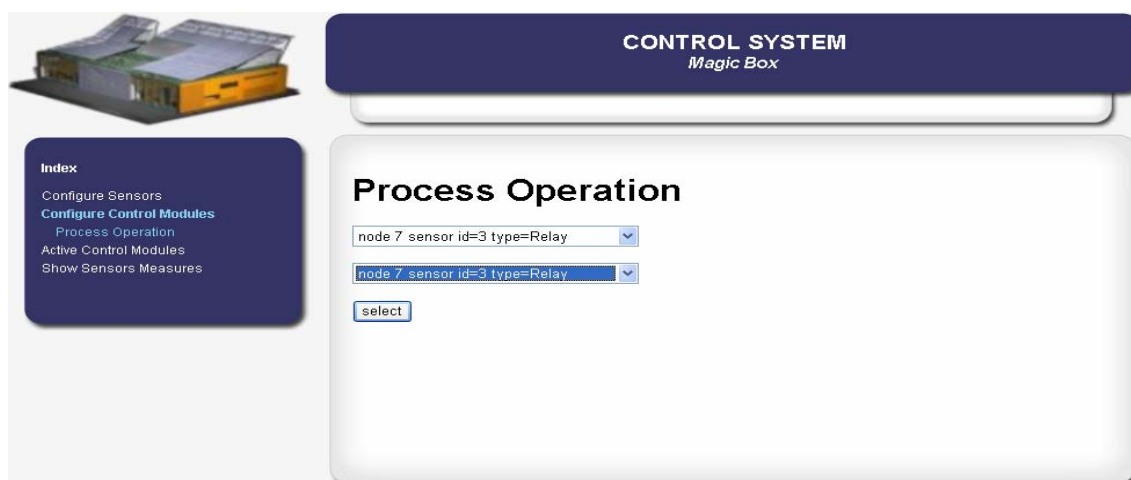


Figura A.3.6 Selección múltiple operaciones tipo sensor.

Una vez se ha seleccionado el sensor o la función media que se desea evaluar el usuario debe seleccionar el tipo de comparador que va a utilizar.



Figura A.3.7 Selección tipo comparador.

Tras el comparador la aplicación solicita al usuario que introduzca el segundo término de la comparación, en éste caso además de los valores ya vistos de sensor y función media de sensores se incorporan también la posibilidad de introducir un valor numérico o un valor de un sensor para una medida inmediatamente anterior a la actual

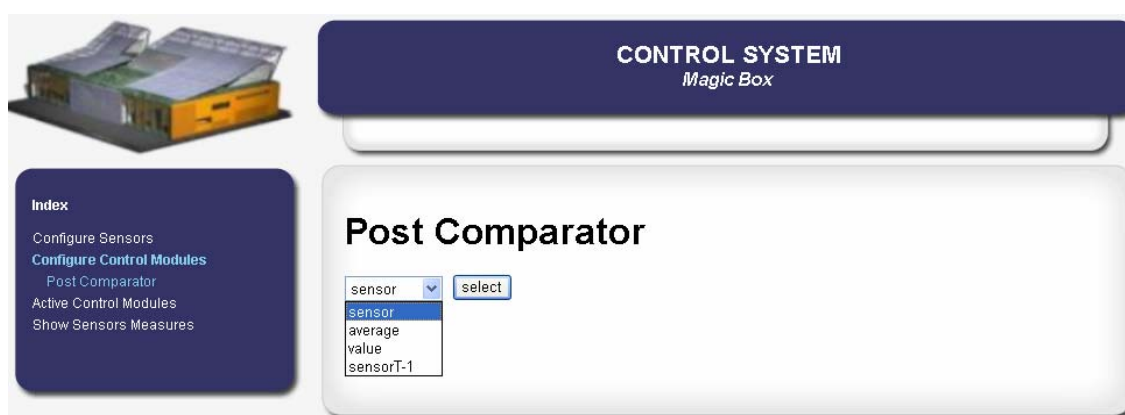


Figura A.3.8 Selección operación posterior al comparador.

Si la operación que se solicita es un valor numérico la aplicación solicita al usuario que introduzca el valor umbral, (Ver Figura A.3.9) el resto de operaciones se tratan de la misma manera que ya se ha visto en las figuras A.3.4, A.3.5 y A.3.6



Figura A.3.9 Selección de Valor Umbral

Tras estas operaciones se ha configurado una condición de control. A continuación la aplicación consulta al usuario si desea anidar más condiciones a la primera configurada.



Figura A.3.10 Selección del concatenador de condiciones

Si se elige un elemento de linkado (*and* u *or* lógicos) entonces la aplicación repetiría el proceso visto anteriormente de generación de una condición simple.

En el caso de que no se haya seleccionado ningún concatenador se da por finalizada la configuración de la condición principal del sistema y se procede a la selección del tipo de acción a realizar, acción si la condición es cierta o acción si la condición es falsa.



Figura A.3.11 Selección de Respuesta Cierta o Falsa

La aplicación contiene una lógica transparente al usuario que detecta con qué condición está relacionada la respuesta. Esto permite configurar varios niveles de indexación de condiciones que posibilitan la construcción de módulos de control complejos. La primera acción seleccionada está automáticamente asociada con la condición principal del sistema. Como se puede apreciar en la Figura A.3.12 la aplicación permite activar o desactivar actuadores del sistema o incorporar nuevas condiciones al módulo de control.

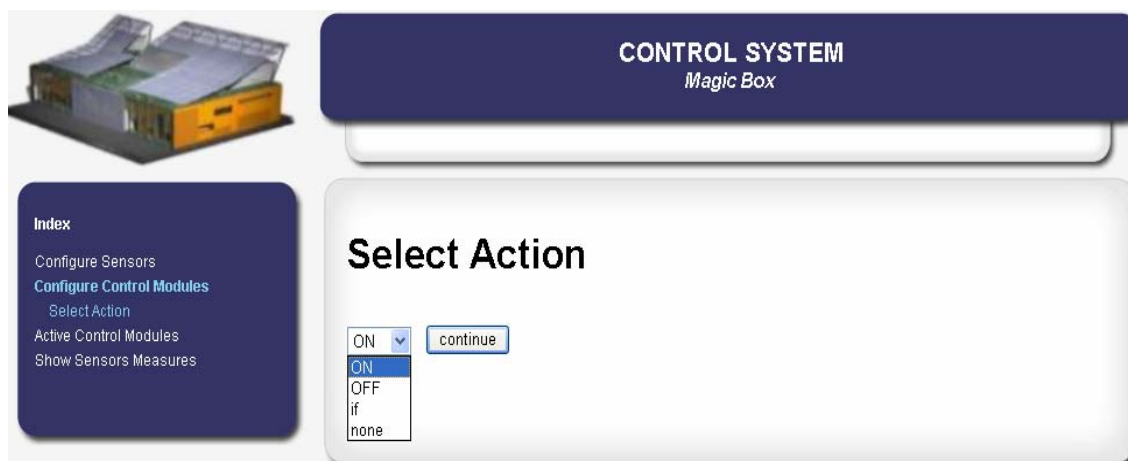


Figura A.3.12 Selección de Acción

En el caso de que la respuesta elegida sea activar o desactivar un actuador, la aplicación muestra los actuadores instalados en el sistema para que el usuario seleccione uno.

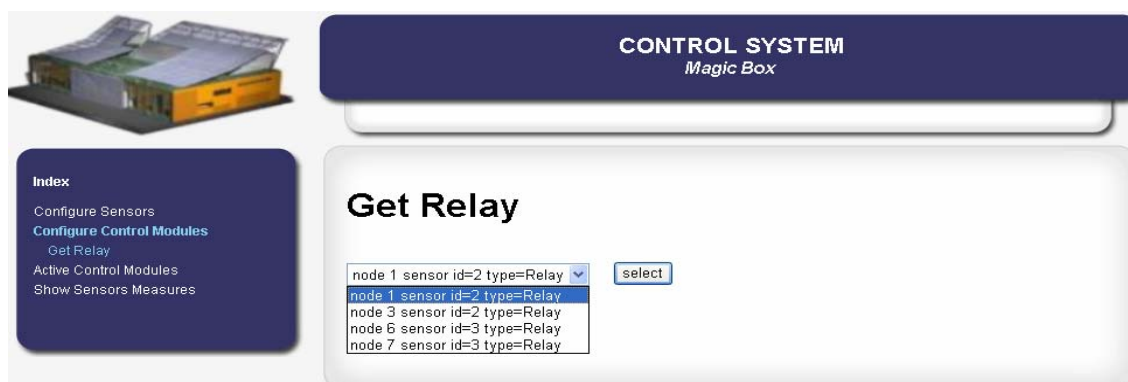


Figura A.3.13 Selección de Actuador

Si se quiere añadir una nueva condición al módulo en la selección de respuestas (Figura A.3.12) se selecciona la opción if. La aplicación incrementará en uno el nivel de indexación, repetirá los pasos vistos ya en este capítulo para la configuración de condiciones, y asignará la primera respuesta al nuevo nivel de indexación. Al concluir la asignación de una respuesta el usuario puede decidir incluir nuevas respuestas en el mismo nivel de indexación en un nivel inmediatamente inferior o en el nivel principal.

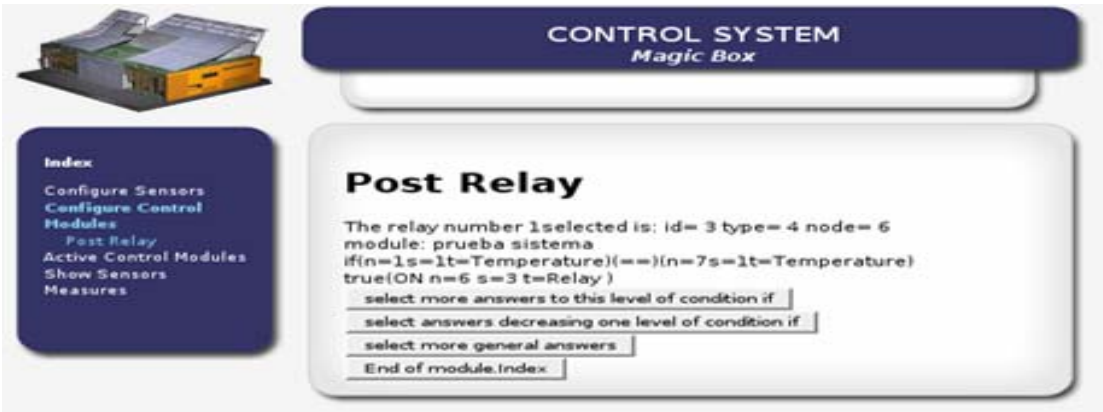


Figura A.3.14 Selección de nuevas respuestas

A.3.2.1. Eliminación de módulos de control

En el caso de que se desee desinstalar un módulo de control y borrar la información asociada a él. Se muestra una tabla con todos los módulos instalados y los datos principales de cada uno de ellos como el nombre, la descripción, el status y las dependencias. (Ver Figura A.3.15)

En caso de que exista un problema con algún nodo de las dependencias, esto es, que el módulo necesite un nodo que no esté instalado o no funcione, esa información se le indica al usuario acompañando la dirección del nodo que genera error con un asterisco.

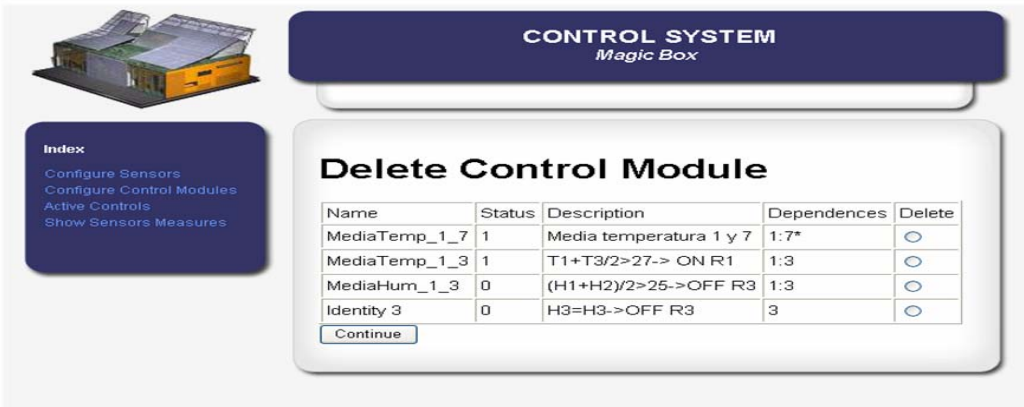


Figura A.3.15 Eliminación Módulo de Control

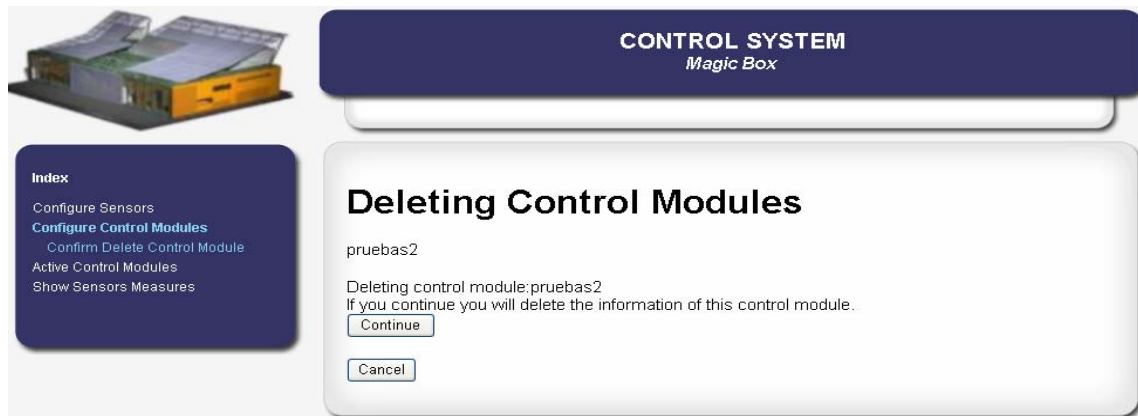


Figura A.3.16 Confirmación Eliminación Módulo de Control

La aplicación permite seleccionar un módulo y proceder a eliminar sus datos o volver al menú principal. Si se selecciona un módulo para ser eliminado se advierte al usuario que la información de ese módulo será borrada y no se podrá recuperar, dándole nuevamente la opción de confirmarlo, o cancelar y volver al menú principal. En caso de seguir adelante con la eliminación del módulo se borran los datos del archivo de control y se regresa al menú principal.

A.3.3. Activación de los módulos de control

Para activar los módulos de control que se desean ejecutar se muestra una tabla con todos los módulos instalados y los datos principales de cada uno de ellos. La aplicación informa al usuario del estado de todos los nodos involucrados en cada módulo de control, de esta forma si se necesita un nodo que no se encuentra instalado se indica la dirección del nodo que genera error por medio de un asterisco.(Ver Figura A.4.1)

El usuario puede seleccionar el estado de todos los módulos de control instalados. Los que se encuentran activos se marcan inicialmente como tal, del mismo modo que los que tienen el status a cero aparecen desactivados. Al finalizar la activación de los módulos se vuelve al índice principal.

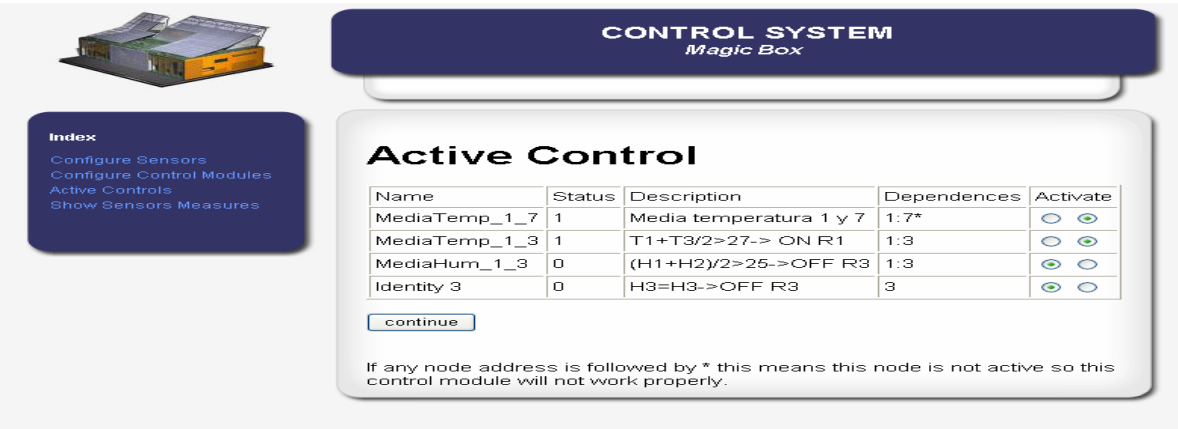


Figura A.4.1 Activación Módulos de Control

A.3.4. Visualización del Estado de la Red.

La última opción de la aplicación permite comprobar el estado de los elementos de la red agrupando los sensores que se deseen visualizar en función de distintas características. Hay cuatro opciones de visualización: (Ver Figura A.5.1)

- Todos los sensores de la red
- Sensores de un mismo tipo
- Sensores de una misma zona
- Sensores de un mismo nodo.

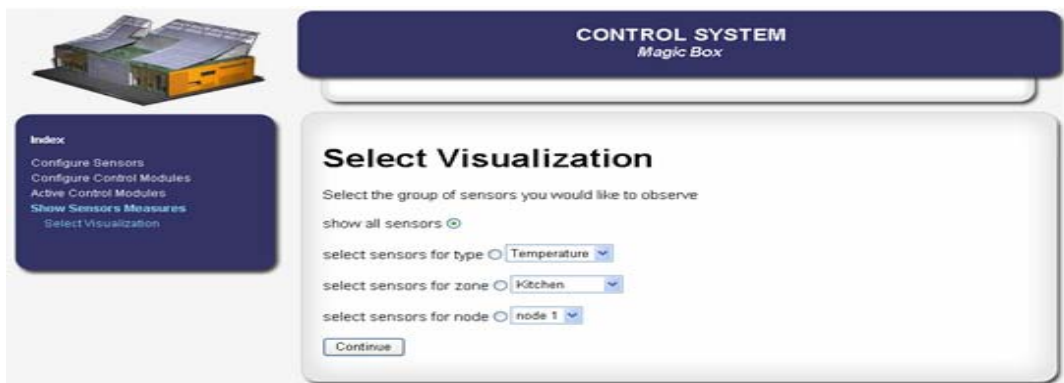
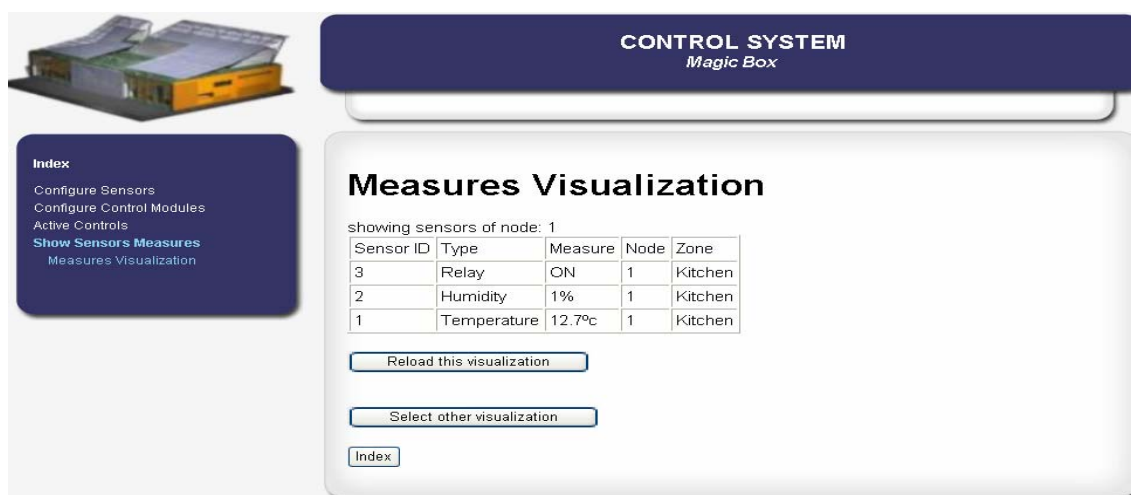


Figura A.5.1 Selección de Visualización

Una vez elegida la visualización se muestra una tabla con la información de los sensores seleccionados. La tabla muestra la ID que identifica al sensor dentro de su nodo, el tipo de sensor, la medida o estado del sensor, el nodo al que pertenece y la zona donde se encuentra ubicado.



CONTROL SYSTEM
Magic Box

Measures Visualization

showing sensors of node: 1

Sensor ID	Type	Measure	Node	Zone
3	Relay	ON	1	Kitchen
2	Humidity	1%	1	Kitchen
1	Temperature	12.7°C	1	Kitchen

Reload this visualization

Select other visualization

Index

Figura 5.2 Visualización sensores nodo 1.

La aplicación permite recargar la misma visualización para poder actualizar los datos que se están mostrando, seleccionar otra visualización distinta o volver al menú principal.