



**UNIVERSIDAD DE JAÉN**  
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

**INTERFAZ GRÁFICA DE USUARIO PARA  
MONITORIZACIÓN Y CONTROL DE UN  
SISTEMA DOMÓTICO**

**Alumno: Miguel Ángel Fernández Ruíz**

**Tutor: D. Luis Miguel Nieto Nieto**

**Depto.: Ingeniería Electrónica y Automática**

**Septiembre, 2014**

# ÍNDICE

<b>1. INTRODUCCIÓN.....</b>	<b>3</b>
<b>2. OBJETIVOS.....</b>	<b>4</b>
<b>3. ESTADO DEL ARTE.....</b>	<b>5</b>
<b>4. DESCRIPCION DEL SISTEMA.....</b>	<b>6</b>
4.1 Estructura del sistema.....	6
4.2 Funcionamiento del sistema.....	7
4.3 Formato de la agenda.....	10
4.4 Comunicaciones con el PC.....	13
<b>5. DESCRIPCION SOFTWARE.....</b>	<b>15</b>
5.1 Manual de usuario.....	15
5.1.1 Requerimientos mínimos.....	15
5.1.2 Instalación.....	15
5.2 Área de trabajo.....	20
5.3 Área de trabajo del acceso remoto.....	37
5.4 Manual de instalación.....	50
5.5 Manual de programación.....	63
5.5.1 Introducción.....	64
5.5.2 Programación multihilo.....	65
5.5.3 Clases.....	66
<b>6. Presupuesto.....</b>	<b>76</b>
<b>7. Conclusiones.....</b>	<b>77</b>
<b>8. Líneas de Futuro.....</b>	<b>78</b>
<b>9. Bibliografía.....</b>	<b>79</b>

## 1.- INTRODUCCIÓN

Este trabajo fin de grado tiene como objetivo principal el desarrollo de una interfaz gráfica que permita el control y monitorización de un sistema domótico de ámbito doméstico. La aplicación debe permitir al usuario configurar el funcionamiento del sistema así como realizar consultas de su estado. Antes de empezar sería conveniente como introducción definir el concepto de domótica, además de los servicios más comunes que debe o puede ofrecer un sistema de este tipo.

- **Definición de domótica**

Hoy en día, es cada vez más frecuente encontrar el término “inteligente” aplicado a muchos productos, incluso a un edificio o vivienda. Por otra parte encontramos otros términos que asimilamos a la inteligencia tales como la informática y las comunicaciones. Es por tanto que se puede denominar a una vivienda o edificio como inteligente cuando incorpore además de las instalaciones tradicionales, nuevas tecnologías como la informática y las comunicaciones para ofrecer ciertos servicios de automatización como gestión energética (climatización), seguridad (intrusos, incendio,...), etc.

Se usan también otros términos, quizás más genéricos, como el de “vivienda automatizada” o “sistemas domésticos” pero sin duda alguna, el término más genérico y global de todos es el de “DOMÓTICA”. Domótica es un término que proviene del francés. Traducido al español sería algo parecido a “vivienda robotizada”. Normalmente se asocia domótica a fracaso, lujo inútil, fallos, problemas y por supuesto CARO, pero sin embargo los grandes avances en informática, comunicaciones y electrónica han hecho que la domótica evolucione rápidamente. Los sistemas más extendidos actualmente son X10 (por corrientes portadoras) y EIB (bus dedicado).

Para concluir, se puede definir domótica como vivienda que integra todos los automatismos en materia de seguridad, gestión de energía, comunicaciones y automatización para ofrecer varios servicios. Para que un sistema se denomine domótico debe ser capaz de integrar los cuatro subsistemas básicos nombrados anteriormente, si falta alguno, no existe domótica.

A continuación se analizan los cuatro subsistemas por separado:

1. La primera categoría es la de *automatización*, un sistema domótico debe permitir servicios de este tipo tales como riego de jardín, control de persianas, temperatura, y todas las tareas que se nos ocurran relacionadas con el confort.
2. Otra categoría es la de *seguridad*. Esta categoría engloba alarmas de incendio, robos, escapes de agua, simuladores de presencia, etc.
3. La tercera categoría es la de *gestión de energía*. Con esto se busca optimizar el consumo energético de la vivienda, por ejemplo se podrían abrir las persianas automáticamente mientras exista luz natural o se podría pensar también en un sistema de gestión de carga eléctrica turnando el funcionamiento de varios aparatos o electrodomésticos para permitir tener una potencia contratada menor.
4. Y finalmente tenemos los servicios de *comunicación*. En todo sistema domótico está presente la comunicación, tanto dentro como fuera de la vivienda. Por ejemplo podemos pensar en poder activar la calefacción con una llamada telefónica, ofrecer servicios de comunicación con un PC, este PC a su vez puede tener un servidor Web o un software de control remoto y ofrecer servicios de telegestión a través de Internet, etc.

## 2.- OBJETIVOS

El objetivo principal de este proyecto es el desarrollo de una interfaz gráfica que permita el control y monitorización de un sistema domótico de ámbito doméstico presentado como proyecto fin de carrera por Víctor Espinosa Gutiérrez en diciembre de 2005, en la EPS de Linares. La aplicación debe contar con un entorno gráfico que permita al usuario configurar el funcionamiento del sistema así como realizar consultas de su estado, empleando un protocolo particular de comunicaciones con el PC basado en RS232. Para el desarrollo de la aplicación debe usarse el entorno de desarrollo Microsoft Visual C++ y la biblioteca de clases Microsoft Foundation Class (MFC). Para el acceso remoto se usará HTML5, hojas de estilo CSS3, y las librerías de Javascript en el lado del cliente, JQuery y JQuery mobile, para permitir mejoras en la interfaz de usuario y páginas web dinámicas. Para la comunicación con el servidor se utilizará AJAX. El servidor será un servidor http embebido con TSL

### 3.- ESTADO DEL ARTE

En la actualidad se comercializan varios sistemas domóticos. Los más representativos del mercado son EIB sobre red de cable fundamentalmente, líder en Europa y X10 que emplea la red eléctrica. EIB es líder en Europa mientras que X10 lo es en USA. EIB es bastante potente, descentralizado e ideal tanto para viviendas como para grandes edificios. X10 es más ideal para viviendas y aunque tiene la ventaja de no necesitar preinstalación, resulta ser un sistema que tiene unos terminales un tanto voluminosos.

A continuación se realiza una comparación de ambas tecnologías además de la utilizada en este proyecto:

Característica	EIB	X10	Tecnología utilizada.
Medio de Transmisión	Usualmente par trenzado	Línea eléctrica (es necesario filtro a la entrada de la vivienda)	(Red RS485) Cable par trenzado
Velocidad transmisión entre terminales	Depende medio físico utilizado. 9600 bps para par trenzado	50bps	9600bps
Nº de nodos	15 áreas con 960 dispositivos por area => 14400 dispositivos	256	255 más la central (cada nodo puede tener varios sensores o actuadores)
Interfaz de usuario	PC, panel empotrado, pulsadores, etc	Mandos a distancia, pulsadores, PC	PC y posibilidad de implementar otras posibilidades mediante módulos

			esclavos (pulsadores, mandos a distancia, teclados, pantallas, voz, ...)
Programable	SI	NO	Si (Agenda de tareas mensual)

Tabla 3.1

Podemos concluir este apartado señalando que aunque la aplicación se basa en una tecnología concreta, se podría extrapolar a cualquiera de las tecnologías existentes en el mercado con solo cambiar la clase encargada de las comunicaciones con el sistema.

#### 4.- DESCRIPCIÓN DEL SISTEMA

A continuación se hará una breve descripción del sistema en el que se basa la aplicación.

##### 4.1.- Estructura del sistema:

El sistema utilizado es un sistema centralizado, es decir, existe un módulo *maestro* (central) y varios *esclavos* hasta un máximo de 255, ya que contamos con 1 byte de dirección y la dirección 0 está reservada para el maestro. Todos los módulos están conectados con un bus según la norma RS485. Los módulos esclavos tendrán sensores o actuadores o también se podría usar una estructura mixta en la que un módulo esclavo tenga tanto sensores como actuadores. El módulo maestro también está conectado a un PC mediante interfaz RS232 y desde la cual la aplicación podrá modificar, programar y monitorizar el sistema.

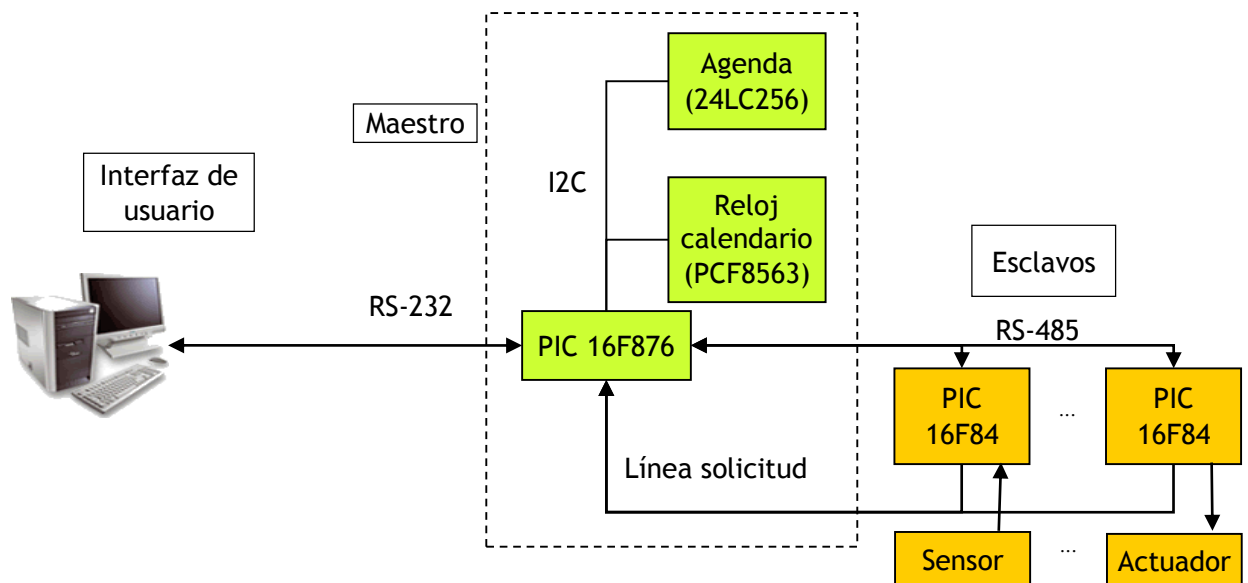


Figura 4.1

#### 4.2.- Funcionamiento del sistema

- Cada módulo esclavo tiene un byte de dirección y ocho bytes de estado de propósito general cuyo significado depende de la aplicación que se ejecute en el esclavo: temperatura, posición de una persiana, configuración del módulo esclavo, etc.
- El módulo maestro tiene un reloj – calendario (con el que puede saber en todo momento la hora, día, mes, etc.) y una agenda de tareas (semanal o mensual).
- Una tarea estará formada por lo siguiente: hora y minuto en la que se ejecutara, dirección del dispositivo, nº de estado (de 0 a 7) y nuevo valor de dicho estado.
- El módulo maestro leerá o cambiará los estados de los módulos esclavos de acuerdo con las tareas que existan en la agenda y lo hará a la hora que se indique en la propia tarea.

- El sistema incluye una línea de control (línea solicitud bus) conectada a todos los esclavos que deseen transferir datos, y por la que solicitan al maestro iniciar una transacción, con independencia del tráfico en el bus RS485. Esta línea también se emplea para notificar alarmas al módulo maestro.
- Los módulos esclavos también pueden cambiar los estados de otros módulos esclavos. De esta forma, se pueden proveer servicios que no sean simples órdenes del maestro a una hora determinada (tales como por ejemplo encender la luz cuando una persona entre a la vivienda, mantener temperatura a un nivel determinado, alarmas de incendio, robo etc...).
- Cuando el maestro recibe la solicitud del bus por parte de algún esclavo, pregunta uno por uno a través del bus RS485 para identificar quién la ha solicitado. A la vez que lo identifica, le da permiso para acceder al bus y cambiar algún estado de otro dispositivo esclavo durante el tiempo que necesite.
- Finalmente tenemos la parte del PC (RS232) que servirá para grabar la agenda semanal con las tareas que queramos así como para controlar y monitorizar el sistema. Es decir, desde el PC se puede leer y cambiar los estados de cualquier dispositivo esclavo. Para conectar el sistema al PC usaremos un cable alargador de puerto serie como los que se pueden encontrar comercialmente para mayor comodidad, tal y como lo muestra la siguiente ilustración.





A continuación se muestra una ficha técnica del sistema:

<b>Característica</b>	<b>Valor</b>
Velocidad comunicación entre terminales	9600bps
Velocidad comunicación con PC	9600bps
Velocidad de los microcontroladores	4Mhz
Tamaño de la memoria de la agenda	32 KB
Número máximo de tareas programables	7984 tareas
Número máximo de días programable	31 días
Periodicidad de la agenda	Diaria, semanal (mensual -> no periódica)
Nº máximo de terminales	255
Nº máximo de sensores, actuadores	Depende del diseño de los terminales esclavos
Interfaz de usuario	PC. Posibilidad de implementarla en módulos esclavos (pulsadores, teclados, pantalla, o incluso voz)
Medio de transmisión entre terminales	Bus RS485 - Cable par trenzado cat5
Longitud máxima del bus RS485	1200m (sin repetidores)

Tabla 4.1

### 4.3.- Formato de la agenda.

La agenda se guarda en una memoria I2C 24LC256 de 256kbits (que equivale a 32kBytes).

La agenda puede ser semanal o mensual lo cual se indicará en el primer byte de memoria. (00 = Semanal, 01 = Mensual). Si esta posición tuviera valor h'FF' indicará que la agenda está vacía o las tareas o datos que contiene no son válidos y no se deben de ejecutar.

Las acciones estarán almacenadas por orden cronológico a partir de la posición h'3F' (63 en decimal).

El formato de las tareas es el siguiente:

Hora	Minuto	Dispositivo	Acción
------	--------	-------------	--------

Figura 4.2

Cada campo ocupa 1 byte, a continuación se detalla para que sirve cada campo:

- Hora y minuto indican la hora a la que se ejecutará la acción. Estos campos están en formato BCD.
- Dispositivo es la dirección RS485 del dispositivo implicado.
- Acción es el valor que se le enviara al dispositivo para modificar alguno de sus estados.

El campo hora y el campo minuto están en formato BCD. Por tanto, los 2 bits más significativos del campo Hora y el bit más significativo del campo minuto se usan para indicar el número de estado con el que se trabajará:

Bit 7 de hora	Bit 6 de hora	Bit 7 de minuto	
0	0	0	Estado 0
0	0	1	Estado 1
.....			.....
1	1	1	Estado 7

Figura 4.3

Las tareas se ejecutan en la hora y minuto indicado. Si se desea ejecutar en un determinado minuto más de una tarea se hará lo siguiente: La primera tarea tendrá la hora y minuto correspondiente y las restantes lo tendrán a 0 (sin contar los bits dedicados para indicar el estado).

Si en un día no hay ninguna tarea a las 23 horas y 59 minutos, se pondrá una tarea con el siguiente formato:

Hora	Minuto	Dispositivo	Acción
23h	59h	00h	00h

Figura 4.4

Esta tarea no ejecutará ninguna acción en ningún dispositivo ya que la dirección 00 es la del maestro. Lo que se pretende con esto es que no exista ningún día vacío y así evitar que tareas de un día determinado se ejecuten en días anteriores.

Para indicar el final de la agenda habrá una acción con el siguiente formato:

Hora	Minuto	Dispositivo	Acción
FFh	FFh	FFh	FFh

Figura 4.5

Para que al conectar el sistema en un momento determinado, este sepa qué tarea toca para ejecutar, se almacenarán desde la posición 1 hasta la 62 las direcciones (punteros) donde comienzan las tareas de cada día (son necesarios 62 porque se direcciona con 2 bytes y hay como máximo 31 días). En el caso de que sea semanal, solo habrá 7 punteros,

es decir, se utilizarán únicamente 14 bytes de los 62. La dirección donde comienza las tareas del primer día siempre será la 63 (decimal).

<b>Dir. memoria (hex / dec)</b>	<b>Contenido</b>
0000 / 0	<b>h'00'</b> si es semanal <b>h'01'</b> si es mensual <b>h'FF'</b> si está vacía
0001 / 1	Dirección de comienzo del día 1 (o Lunes si es semanal) , Byte alto
0002 / 2	Dirección de comienzo del día 1 (o Lunes), Byte bajo
0003 / 3	Dirección de comienzo del día 2 (o Martes), Byte alto
0004 / 4	Dirección de comienzo del día 2 (o Martes), Byte bajo
0005 / 5	Dirección de comienzo del día 3 (o Miércoles), Byte alto
.....	.....
003E / 62	Dirección de comienzo del día 31, Byte bajo
<b>003F / 63</b>	<b>COMIENZO DE LA AGENDA (TAREAS DEL DÍA 1 si es mensual o del Lunes si es semanal)</b>

Tabla 4.2

Si la agenda es semanal se repetirá indefinidamente, es decir, cuando acabe las tareas del domingo, volverá a las del lunes.

En el caso de que sea mensual no se repetirá. Para conseguir esto, se pondrá el primer byte de memoria con el valor h'FF' al finalizar la agenda para indicar que la agenda está vacía y evitar que se ejecute de nuevo.

Si queremos una agenda semanal que no se repita, se utilizará una mensual en la que especificaremos los días que se ejecutarán tareas y los días en los que no habrá ninguna.

Aunque no se indique explícitamente, también se permiten agendas diarias y que se repitan. Para ello basta con hacer una agenda semanal en la que los punteros de todos los días apunten a la dirección h'003F'.

#### 4.4.- Comunicaciones con el PC

Las acciones que el PC puede realizar con el sistema son:

1. **Controlar el sistema (F0):** Para ello debe enviar el byte h'F0'. El PC puede leer o modificar los 8 estados de cualquier dispositivo esclavo. Para ello debe enviar al PIC los campos de una trama RS485 en el siguiente orden: FLAG\_CONTROL, DIRECCIÓN y DATOS. La rutina, por seguridad, solo dejará leer y modificar estados por lo que pone a 0 los bits del campo FLAG\_CONTROL dedicados a otras operaciones para impedir que el PC pueda efectuarlas. La rutina devolverá un valor correspondiente al valor leído, o bien al valor por el que se ha modificado según sea la acción realizada.

1 Bit	7 Bits	1 Byte	1 Byte
Flag	Control	Dirección de destino	Dato

Figura 4.6

FLAG	CONTROL						
0	Y1	Y2	Y3	Y4	X2	X1	X0

Figura 4.7

Bit y quién lo usa ↓	Valor 0	Valor 1
(Maestro) Y1 =	Leer estado	Modificar estado por el valor "datos"
(Esclavo) Y2 =	NACK (Error)	ACK (Todo correcto)
(Esclavo) Y3 =	No quiere acceder al bus	Solicitud de acceso al bus
(Maestro) Y4 =	No permiso bus	Da permiso de acceso al bus
X2,X1,X0 =	Nº de estado con el que se trabajará	

Figura 4.8

2. **Leer hora (F1):** El maestro envía los 16 bytes que posee el reloj-calendario (PCF8563) al PC. El Formato de estos bytes es el siguiente:

Address	Register name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00H	control/status 1	TEST1	0	STOP	0	TESTC	0	0	0
01H	control/status 2	0	0	0	TI/TP	AF	TF	AIE	TIE
0DH	CLKOUT control	FE	x	x	x	x	x	FD1	FD0
0EH	timer control	TE	x	x	x	x	x	TD1	TD0
0FH	timer	<timer countdown value>							

Figura 4.9

Address	Register name	BCD format tens nibble				BCD format units nibble			
		Bit 7 2 <sup>3</sup>	Bit 6 2 <sup>2</sup>	Bit 5 2 <sup>1</sup>	Bit 4 2 <sup>0</sup>	Bit 3 2 <sup>3</sup>	Bit 2 2 <sup>2</sup>	Bit 1 2 <sup>1</sup>	Bit 0 2 <sup>0</sup>
02H	seconds	VL	<seconds 00 to 59 coded in BCD>						
03H	minutes	x	<minutes 00 to 59 coded in BCD>						
04H	hours	x	x	<hours 00 to 23 coded in BCD>					
05H	days	x	x	<days 01 to 31 coded in BCD>					
06H	weekdays	x	x	x	x	x	<weekdays 0 to 6>		
07H	months/century	C	x	x	<months 01 to 12 coded in BCD>				
08H	years	<years 00 to 99 coded in BCD>							
09H	minute alarm	AE	<minute alarm 00 to 59 coded in BCD>						
0AH	hour alarm	AE	x	<hour alarm 00 to 23 coded in BCD>					
0BH	day alarm	AE	x	<day alarm 01 to 31 coded in BCD>					
0CH	weekday alarm	AE	x	x	x	x	<weekday alarm 0 to 6>		

Figura 4.10

3. **Leer agenda (F2):** El PC puede leer la agenda en su totalidad. El maestro envía todos los datos byte a byte al PC. Esta opción no se ha implementado en la aplicación ya que la esta guarda una copia en el disco duro.
4. **Ajustar hora (F3):** El PC puede modificar los parámetros del reloj – calendario PCF8563 para ajustarlos correctamente. Para ello debe enviar al PIC todos los campos en el siguiente orden: segundos, minutos, hora, día, día de la semana, mes y año. Estos campos tienen que estar en el formato que sigue el dispositivo PCF8563, es decir, en BCD.
5. **Grabar agenda (F4):** El PC puede grabar la agenda de tareas (semanal o mensual) en la memoria I2C. El PC debe enviar todos los datos en bloques de 64 bytes (ya que es el tamaño máximo que se pueden escribir en la memoria 24LC256 de una vez debido al buffer de escritura que posee). Cada vez que el PC reciba un byte con valor h'BB' podrá enviar un nuevo bloque. Para terminar debe enviar cuatro bytes h'FF' que es la tarea que indica el fin de la agenda.

## 5.- DESCRIPCIÓN SOFTWARE

En este apartado se explicará el software del cual es objetivo este proyecto. La explicación se dividirá en tres partes:

### 5.1.- Manual de usuario.

#### 5.1.1 *Requerimientos Mínimos*

- Ordenador compatible que conste de:
  - 1 GHz Pentium®/AMD en adelante.
  - Espacio libre en disco duro de 50 MB.
  - Memoria RAM de 128 MB.
- Sistema Operativo *Microsoft Windows XP*.
- Motor de base de datos Microsoft Jet.
- Adaptador puerto RS232

#### 5.1.2 *Instalación*

Para la instalación del programa, se ejecutará el archivo “Setup.exe” contenido en el CD que acompaña al proyecto. Lo primero que hay que indicar es el idioma a utilizar en la instalación:

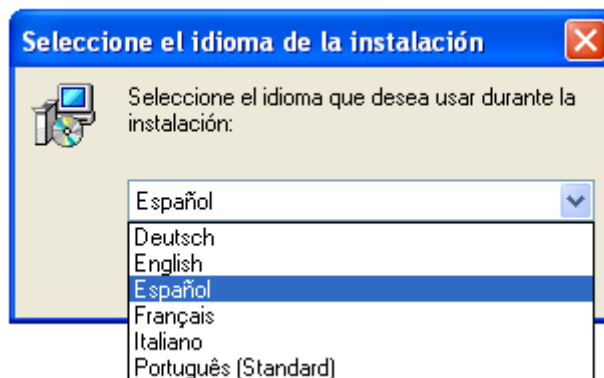


Figura 5.1

Después, se mostrará la siguiente pantalla:

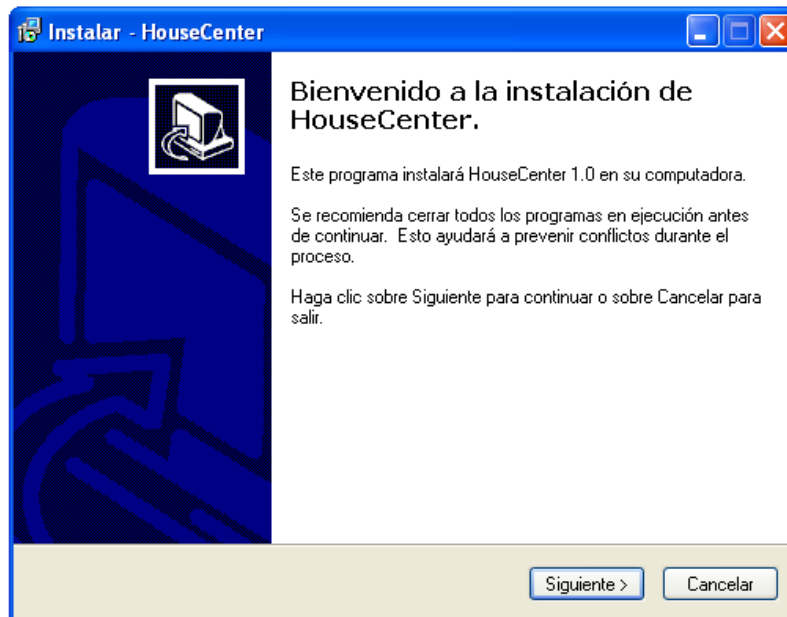


Figura 5.2

El proceso de instalación se podrá cancelar en cualquier momento, para ello sólo se tendrá que pulsar Cancelar, se pedirá una confirmación y se indicará que el proceso de instalación podrá ejecutarse de nuevo para completarla.

Se pulsará Siguiente y se seleccionará la carpeta destino de la aplicación, por defecto esta carpeta se encuentra en:

“C:\Archivos de Programa\HouseCenter”:

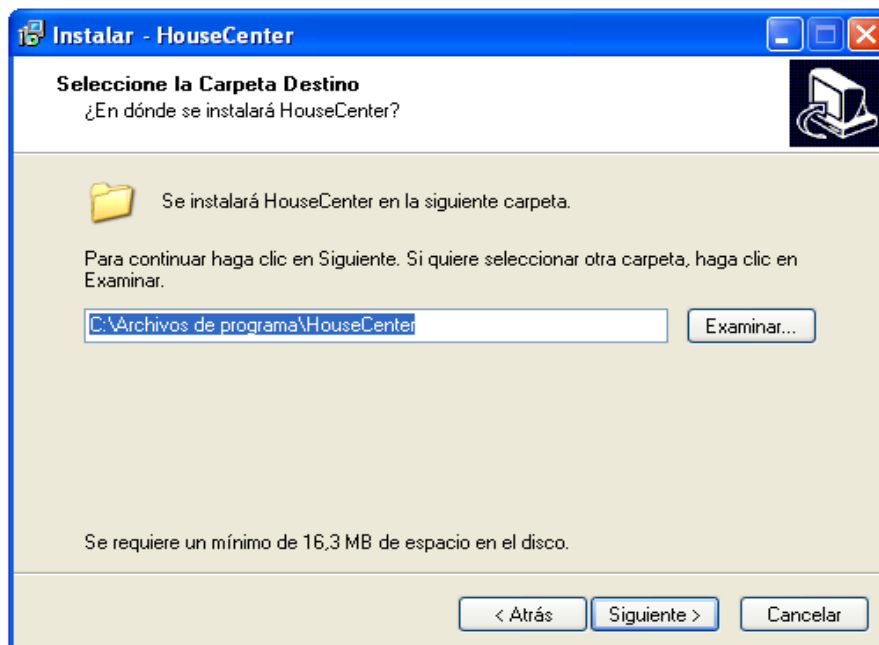


Figura 5.3



Después se indicara si ya existía la carpeta, por si se sobrescribe su contenido:

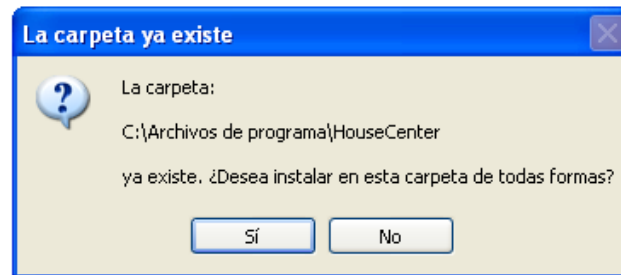


Figura 5.4

Después se pulsará siguiente y se seleccionarán los componentes que se desean instalar. Se distinguirán 2 tipos de instalación:

- **Instalación Completa:** Incluye el programa principal, su ayuda, su código fuente.
- **Instalación Personalizada:** Mediante esta opción se seleccionará manualmente los componentes que se deseen instalar

En esta pantalla además se indicará el espacio necesario en disco para cada tipo de instalación.

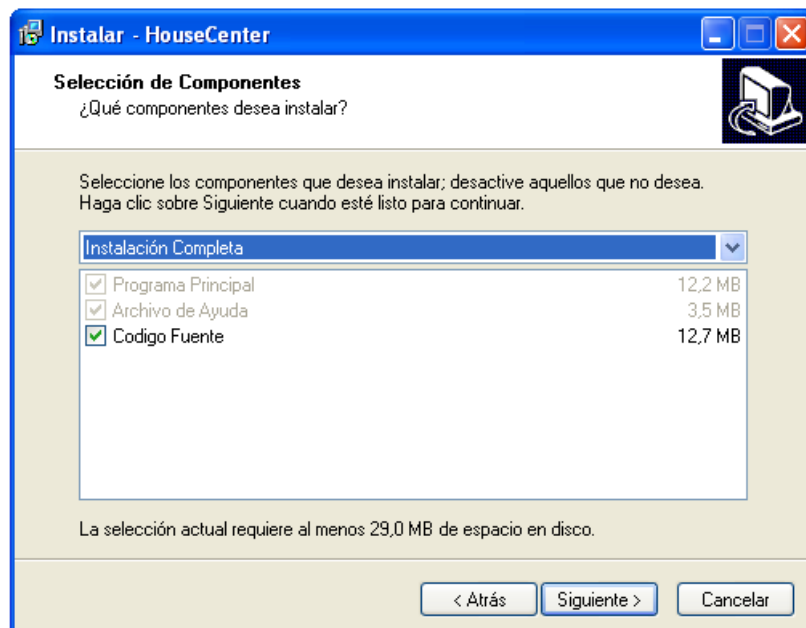


Figura 5.5

El próximo paso será indicar la carpeta del Menú de Inicio donde se crearán los iconos del programa.

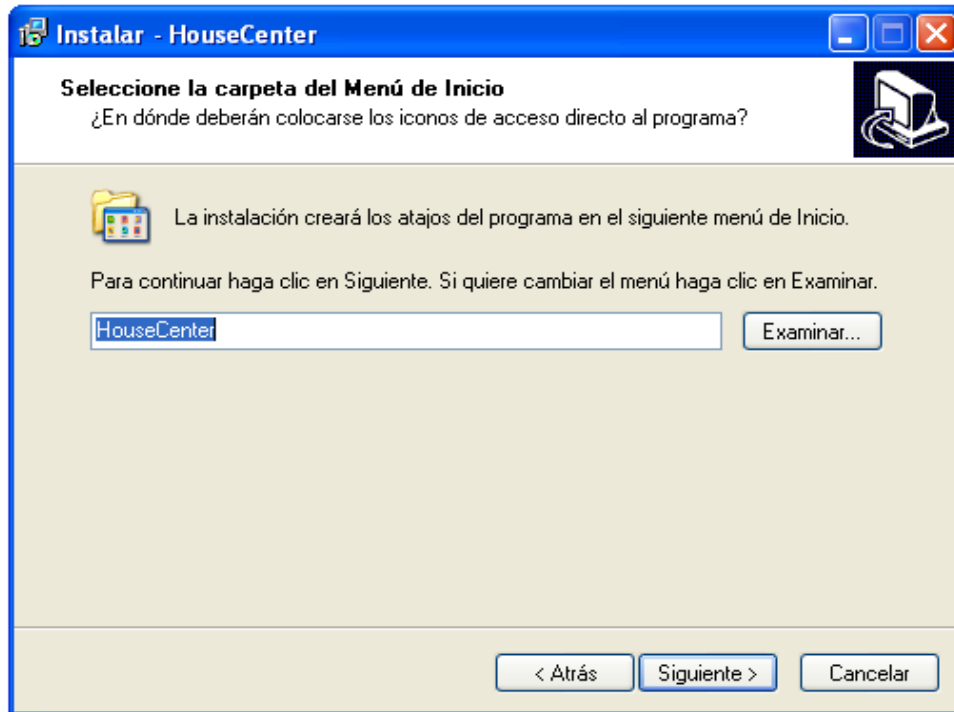


Figura 5.6

La siguiente pantalla dará la opción de crear un acceso directo en el escritorio, para ello sólo se tendrá que marcar la opción “Crear un atajo en el escritorio”.

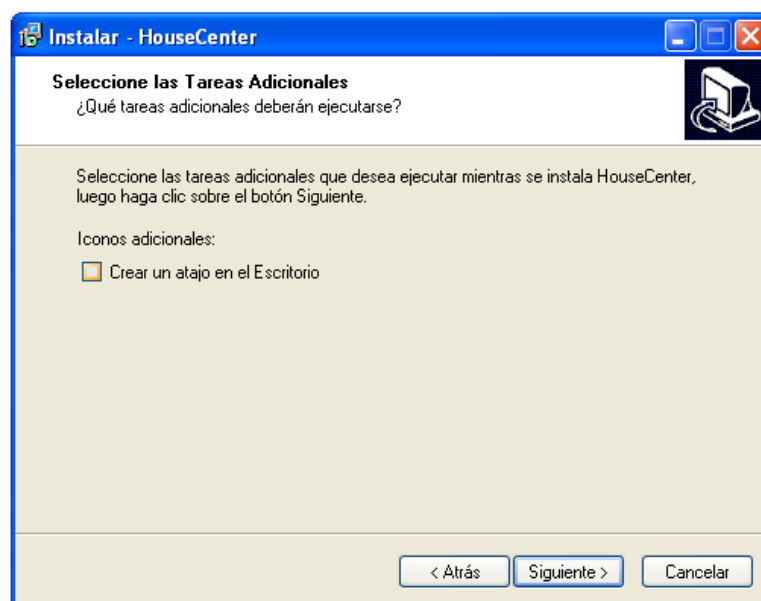


Figura 5.7

Antes de realizar la instalación se mostrará un resumen de las opciones seleccionadas en dicha instalación, de tal forma que puedan confirmarse las elecciones, y en el caso de no estar de acuerdo se podrán modificar hasta conseguir la instalación deseada (Tal y como muestra la figura 5.8)

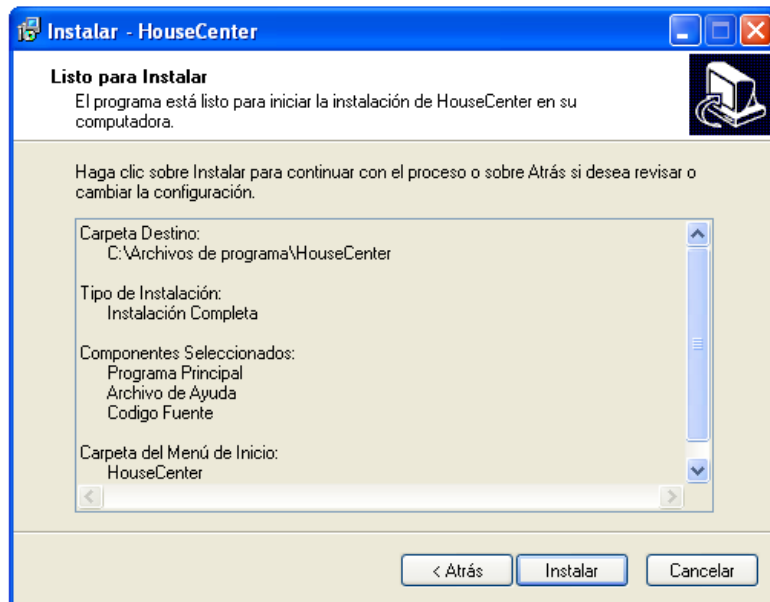


Figura 5.8

Una vez pulsado instalar aparecerá una barra de progreso indicando el proceso de instalación. Cuando este proceso haya acabado se dará la opción de ejecutar automáticamente el programa y aparecerá la siguiente pantalla indicando el fin de la instalación.

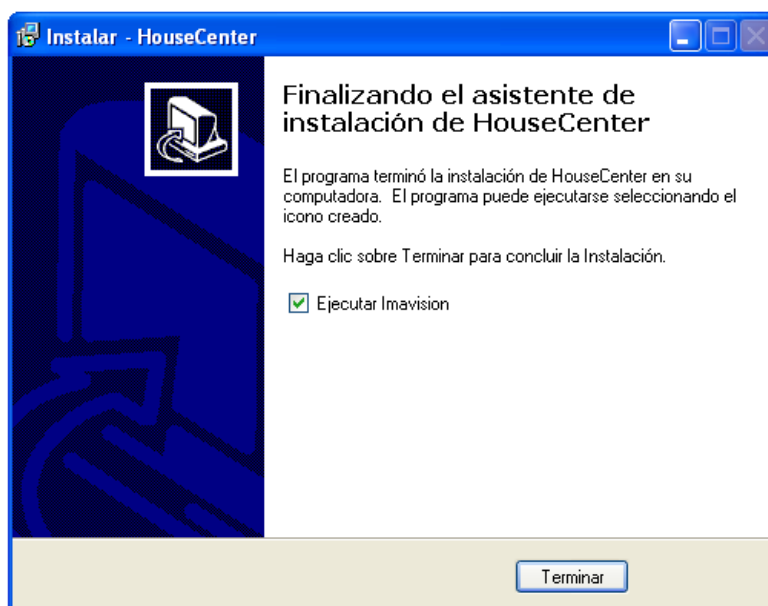


Figura 5.9

## 5.2.- Área de trabajo.

El área de trabajo de la aplicación presenta el aspecto reflejado en la siguiente figura:

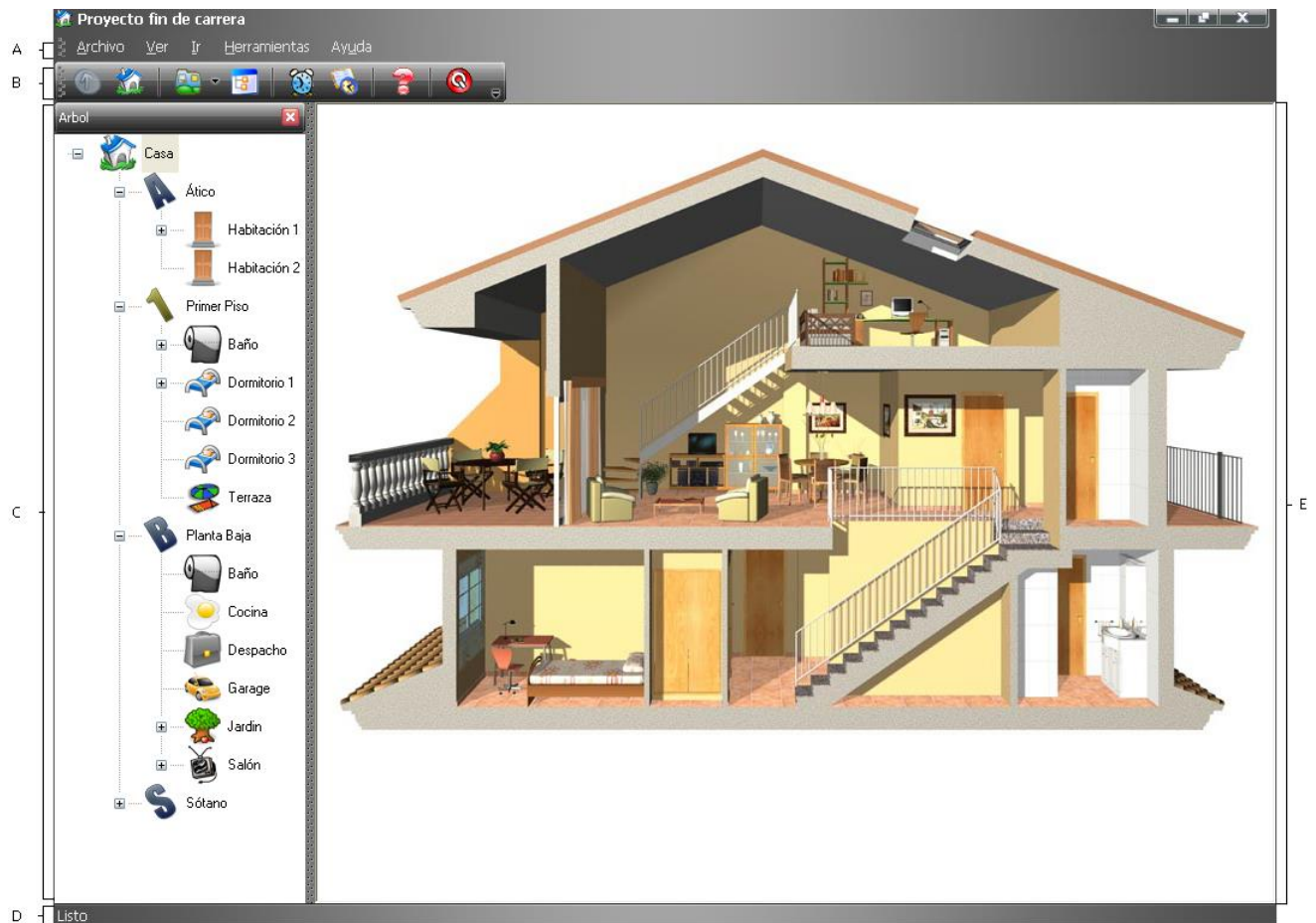


Figura 5.10

Se puede observar en la imagen los distintos componentes que lo forman:

- A. Menú
- B. Barra de herramientas
- C. Árbol
- D. Barra de estado
- E. Vista

- **Acerca del Menú.**

Contiene menús organizados por tareas.



Figura 5.11

**Menú Archivo:**

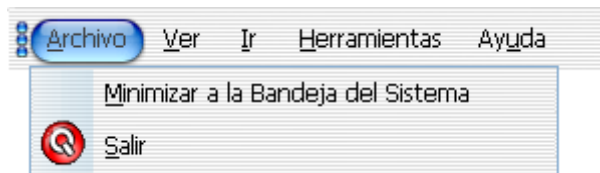


Figura 5.12

**Salir:** Cierra la aplicación.

**Minimizar a la Bandeja del Sistema:** Permite minimizar la aplicación a la bandeja del sistema. Para maximizar la aplicación solo hay que hacer un doble clic sobre el icono situado en la bandeja del sistema.



Figura 5.13

Si se pincha sobre el icono con el botón derecho de la aplicación se mostrara un menú que nos permite cerrar la aplicación, restaurarla o acceder al dialogo Acerca de.

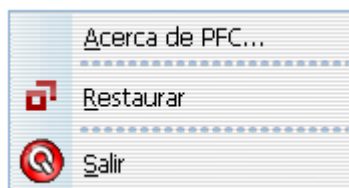


Figura 5.14

### **Menú Ver:**

Permite personalizar el área de trabajo.

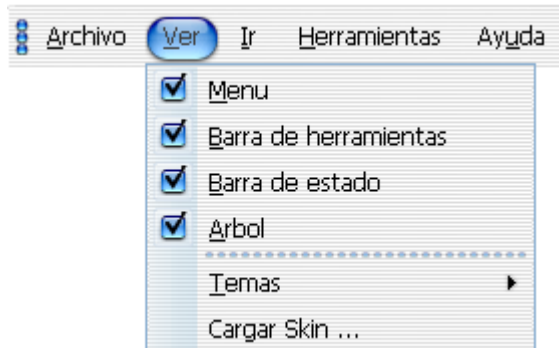


Figura 5.15

**Menú:** Muestra u oculta el menú.

**Barra de herramientas:** Muestra u oculta la barra de herramientas.

**Barra de estado:** Muestra u oculta la barra de estado.

**Árbol:** Muestra u oculta el árbol.

Puede que llegado a este punto se haya dado cuenta de que la captura de la aplicación expuesta al principio de esta sección difiere de las capturas de los menús. Esto es debido a que la aplicación puede cambiar su aspecto. Veamos como:

**Temas:** Permite cambiar el aspecto de la aplicación. Para ello solo hay que elegir uno de los temas disponibles.

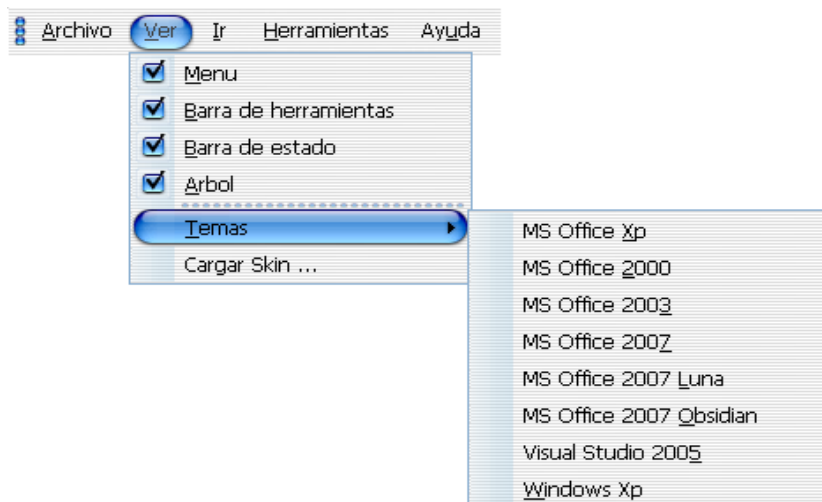


Figura 5.16

**Cargar Skin:** Permite cambiar el aspecto de la aplicación. Para ello, se hará uso de un cuadro de dialogo en el que indicaremos el archivo .skin que deseamos cargar. El skin puede estar en cualquier directorio.

Veamos un ejemplo de algunos temas o skins:

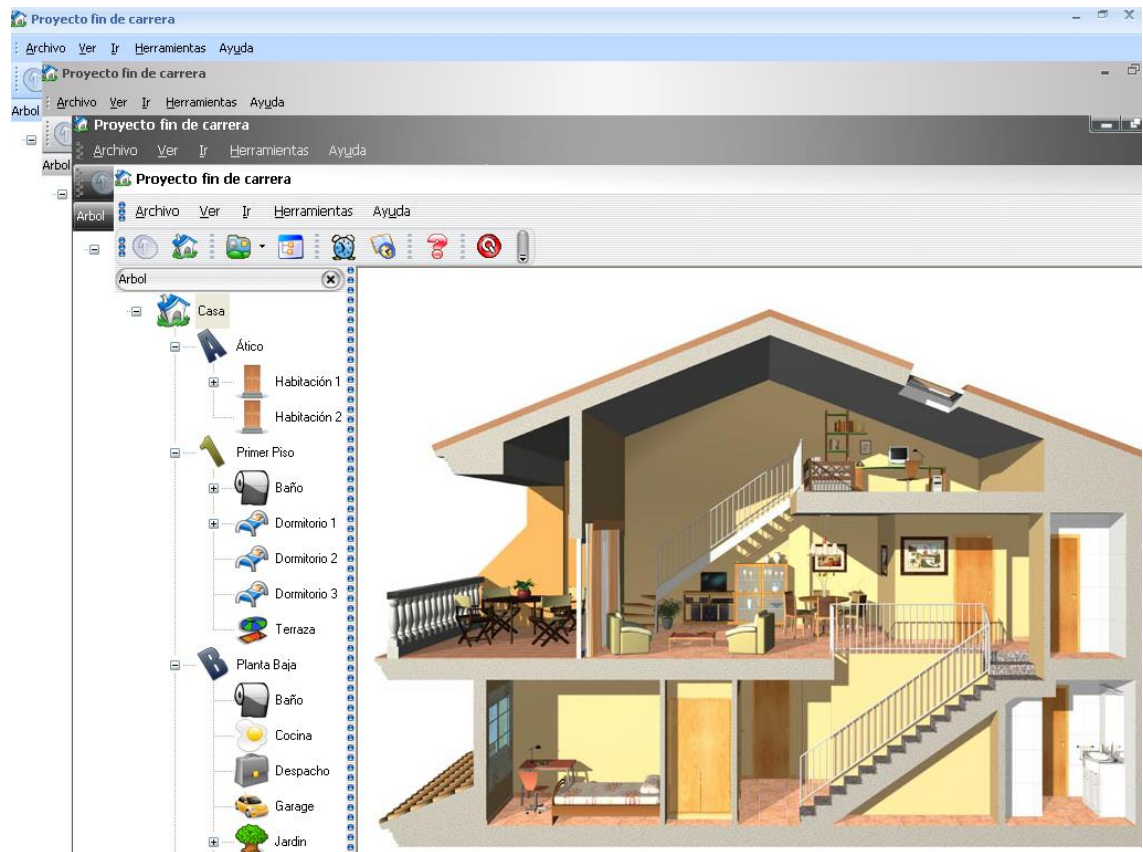


Figura 5.17

### **Menú Ir:**

Permite una fácil navegación por el árbol.

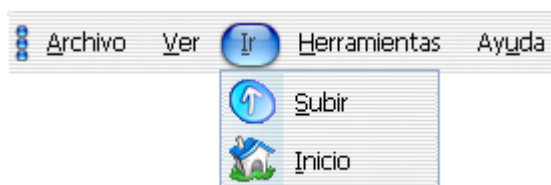


Figura 5.18

**Subir:** Sube un nivel en el árbol.

**Inicio: Selecciona** la raíz del árbol.

### **Menú Herramientas:**

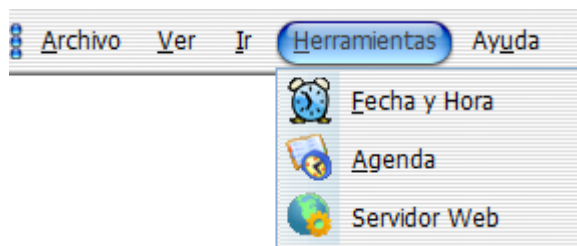


Figura 5.19

**Fecha y Hora:** Permite sincronizar el reloj de la central domótica con el reloj del PC. Para ello haremos uso de un cuadro de dialogo que nos mostrara la hora de la central y así poder decidir si queremos sincronizarla o no.



Figura 5.20

Una vez sincronizada la hora de la central con el PC no se permite una nueva sincronización hasta que no se vuelva a cargar el cuadro de dialogo para así no saturar a la central con múltiples cambios de hora.

**Agenda:** Nos permite poder manipular la agenda. Para ello haremos uso de una hoja de propiedades, que no es más que un cuadro de dialogo con opciones que nos permite acceder a las respectivas paginas pulsando en la lista situada en la



parte izquierda del cuadro de dialogo o pulsando siguiente o atrás. (Tal y como se muestra en la siguiente figura)

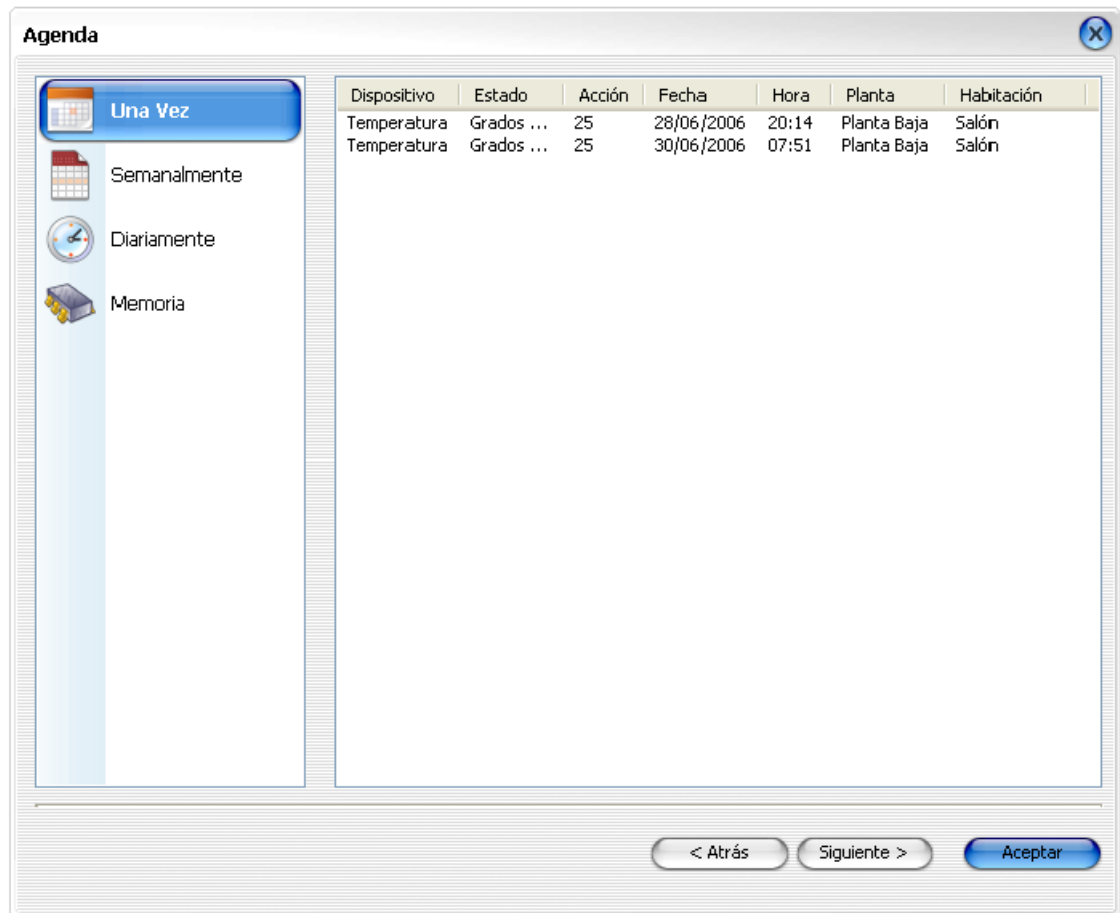


Figura 5.21

En la agenda podemos distinguir cuatro páginas:

- **Una Vez:** Aquí se listaran las tareas que se repiten solo una vez, es decir, la tarea programada solo se ejecutara en el día y hora programada. Cuando seleccionamos una o más tareas de la lista y pulsamos el botón derecho del ratón lanzaremos un menú que nos permitirá borrar o ejecutar inmediatamente todas las tareas que han sido seleccionadas.
- **Semanalmente:** Aquí se listaran las tareas que se repiten semanalmente, es decir, que si tenemos una tarea programada para los días Lunes, Martes, y Miércoles la tarea se ejecutara esos días todas las semanas. Aquí, al igual que en la anterior página, también

dispondremos de un menú que nos permita borrar o ejecutar inmediatamente las tareas que han sido seleccionadas.

- **Diariamente:** Aquí se listan las tareas que se repiten diariamente, es decir, las tareas se ejecutarán a la hora programada todos los días. De forma similar a lo explicado anteriormente, se dispondrá también de un menú que nos permita borrar o ejecutar inmediatamente las tareas que han sido seleccionadas.
  
- **Memoria:** Esta página nos permite grabar la memoria hardware del sistema. Para ello dispondremos de cuatro opciones:
  1. Mensual: La agenda durará cuatro semanas. Para generarla se tienen en cuenta las tareas que se ejecutan una sola vez, las que se ejecutan semanalmente y las tareas que se ejecutan diariamente.
  2. Semanal: La agenda de la memoria será semanal. Para generarla se tienen en cuenta las tareas que se ejecutan una sola vez, las que se ejecutan semanalmente y las tareas que se ejecutan diariamente.
  3. Semanal Repetitiva: La agenda será semanal y se repetirá indefinidamente. Para generarla solo se tienen en cuenta las tareas que se ejecutan semanalmente.
  4. Diaria Repetitiva: La agenda será diaria y se repetirá todos los días indefinidamente. Para generarla solo se tienen en cuenta las tareas que se ejecutan diariamente.

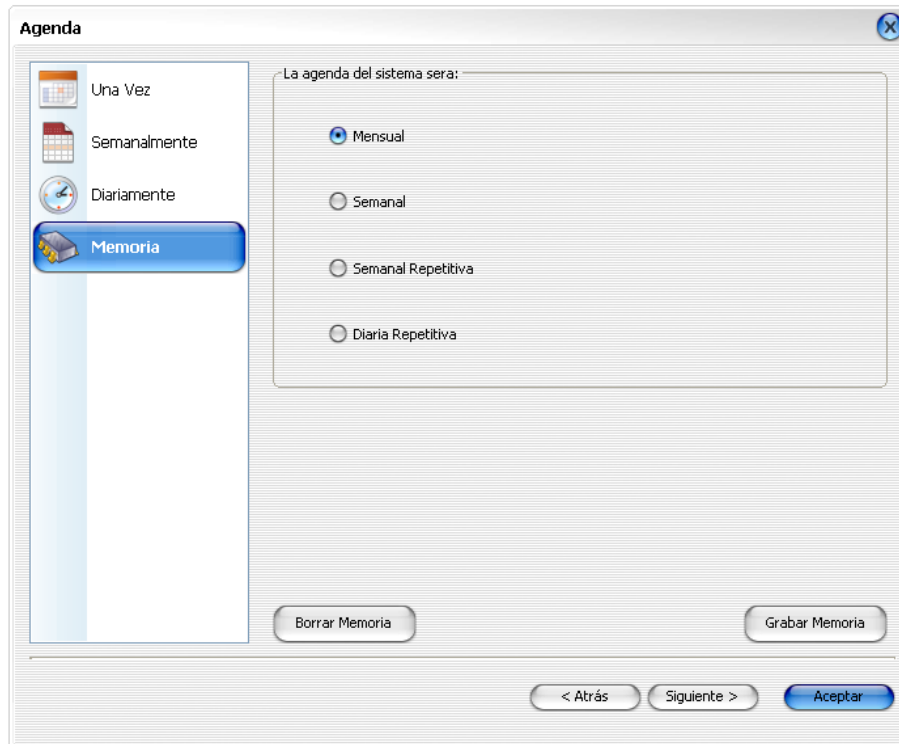


Figura 5.22

**Servidor Web:** Permite lanzar el servidor web, establecer el puerto donde escuchara, así como el usuario y clave que necesitara el usuario para conectarse remotamente a la aplicación. Para ello haremos uso de un cuadro de dialogo en el que podremos modificar cada uno de estos parámetros.



Figura 5.23

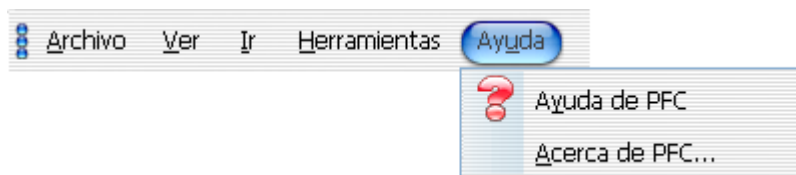
**Menú Ayuda:**

Figura 5.24

**Ayuda:** Esta opción permitirá abrir el archivo de ayuda que informará sobre el uso del programa.

**Acerca de:** Esta opción lanza un cuadro de dialogo que informa de la versión del programa, así como del nombre del autor.



Figura 5.25





- **ACERCA DE LA BARRA DE HERRAMIENTAS**






Permite tener un acceso rápido a las opciones más importantes del menú.



Figura 5.26

Las diferentes opciones que podemos encontrar son:

-  *Subir:* equivale a Menú Ir | Subir.
-  *Inicio:* equivale a Menú Ir | Inicio.
-  *Temas:* permite cambiar la apariencia de la aplicación.
-  *Árbol:* equivale a Menú Ver | Árbol.

-  Fecha y Hora: equivale a Menú Herramientas | Fecha y Hora.
-  Agenda: equivale a Menú Herramientas | Agenda.
-  Servidor Web: equivale a Menú Herramientas | Servidor Web
-  Ayuda: equivale a Menú Herramientas | Ayuda.
-  Salir: equivale a Menú Archivo | Salir.

- **ACERCA DEL ARBOL**

Permite una fácil navegación por las diferentes partes de la vivienda.

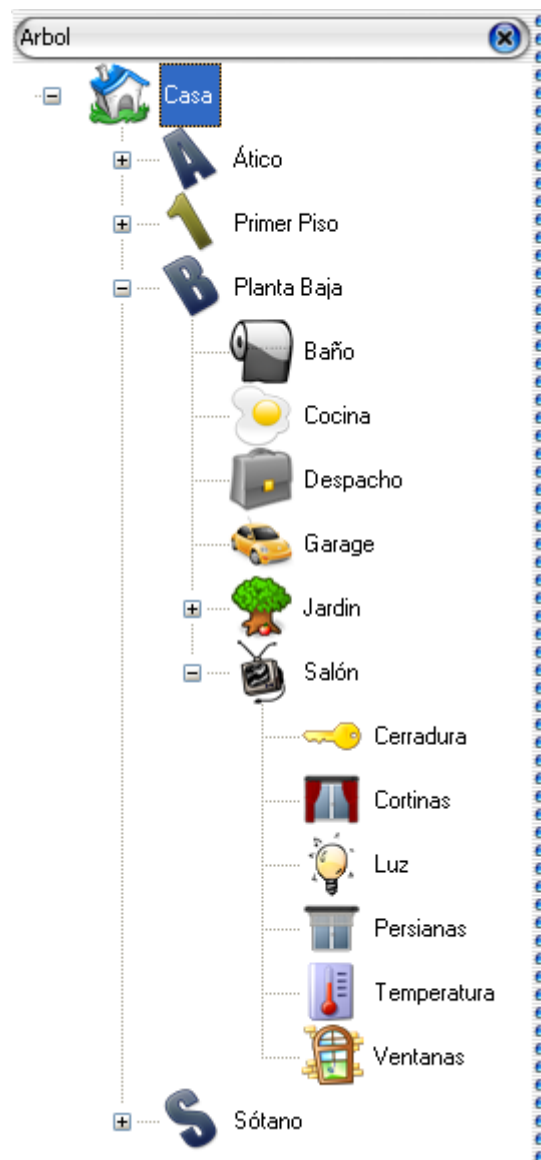


Figura 5.27

En el árbol se refleja las diferentes partes que nos podemos encontrar en una casa así como los componentes instalados en ella. El primer nivel del árbol estará formado por las diferentes plantas de la que puede constar la vivienda. En el segundo nivel tendremos las habitaciones que hay en cada planta y por último, el tercer nivel constara de los dispositivos que hay instalados en la vivienda. Así pues, observando el árbol de la captura anterior podemos ver que la casa, en la que se basa el árbol, tiene cuatro plantas. Que en la planta baja hay un baño, una cocina, un salón,..., y que en el salón hay instalados un control para la luz, las temperatura, las ventanas etc...

Al seleccionar un ítem del árbol pueden ocurrir dos cosas: O bien se muestra en la **vista** el plano correspondiente a esa parte de la vivienda o una foto de la habitación con los componentes que hay instalados en ella. De esta manera, si pulsamos en una planta observaremos su plano correspondiente. En cambio, si pulsamos en una habitación o en un componente se mostrara la foto de la habitación.

El árbol también nos permite acceder a la hoja de propiedades de un determinado componente. Dicha hoja nos permite realizar consultas de los estados del componente, modificarlos o programar las tareas. Para acceder a la hoja de propiedades tendremos que pulsar la tecla enter cuando este seleccionado el componente o realizar un doble click sobre el componente. El funcionamiento de la hoja de propiedades se explicara posteriormente cuando se vea el funcionamiento de la vista.

- **ACERCA DE LA BARRA DE ESTADO**

La barra de estado, situada en la parte inferior de la ventana de la aplicación, muestra instrucciones breves acerca del uso de la herramienta activa.

- **ACERCA DE LA VISTA**

Muestra el plano activo o la foto de la habitación activa.



Figura 5.28

Además de mostrar el plano o la foto correspondiente, la vista tendrá las mismas funciones del árbol. Es decir, podremos movernos por las diferentes partes de la vivienda con solo un clic del ratón, y lanzar la hoja de propiedades del dispositivo sin tener que usar el árbol. Veamos como:

Si lo mostrado en la vista es un plano, al mover el puntero del ratón por encima de la vista, veremos que aparecen unos tooltips que nos indicaran el destino al que nos dirigiríamos si pulsásemos el botón izquierdo del ratón. Por ejemplo, en la imagen anterior podemos ver que aparece un tooltip en el que se puede leer “*Planta Baja*”, pues bien, si pinchamos en esas coordenadas de la vista, en el árbol se seleccionara el ítem “*Planta Baja*” y en la vista se mostrara el plano correspondiente a la planta baja. Así nos podemos mover por la vivienda sin necesidad de utilizar el árbol.

Cuando lo mostrado en la vista corresponde a una habitación, la vista mostrara, además de la foto, los distintos dispositivos que hay instalados en la

habitación. Así pues, para acceder a la hoja de propiedades de un dispositivo, solo tendremos que pulsar su botón correspondiente.

Si se pulsa el botón derecho del ratón cuando el puntero esta sobre la vista podremos subir un nivel en el árbol y en la vista se mostrara el plano correspondiente a ese nivel. (Ilustración 5.1).



Ilustración 5.1

### ***Hoja de propiedades del dispositivo.***

La hoja de propiedades del dispositivo nos permite ver cuál es su estado, programar las tareas que queramos para ese dispositivo, o modificar sus estados. La hoja se podrá lanzar desde el árbol o la vista. Se lanzara desde el árbol cuando hagamos un doble clic sobre el dispositivo o cuando pulsemos enter cuando el dispositivo este seleccionado. Y se lanzara desde la vista cuando pinchemos sobre el botón correspondiente al dispositivo.

En la hoja de propiedades podemos distinguir tres páginas:



- En la primera podremos ver cuál es el estado del dispositivo en este preciso instante así como poder cambiar cada uno de los estados. El título de la página dependerá de cual sea el nombre del dispositivo. Para cambiar los estados tendremos tres tipos de controles. Un checkbox, un slider que va acompañado de un control edit, y por ultimo una lista desplegable. También se ha proporcionado un control static para aquellos estados que sean de solo lectura. Hay que tener en cuenta que si el dispositivo solo tuviese estados de solo lectura, las páginas de “opciones” y “Fecha y Hora” no estarán habilitadas ya que en un dispositivo de este estilo no se permite la programación de tareas. Una vez elegido los valores que queremos asignarle a los estados, solo tendremos que pulsar el botón “aplicar” para aplicar los cambios inmediatamente. (Figura 5.29)

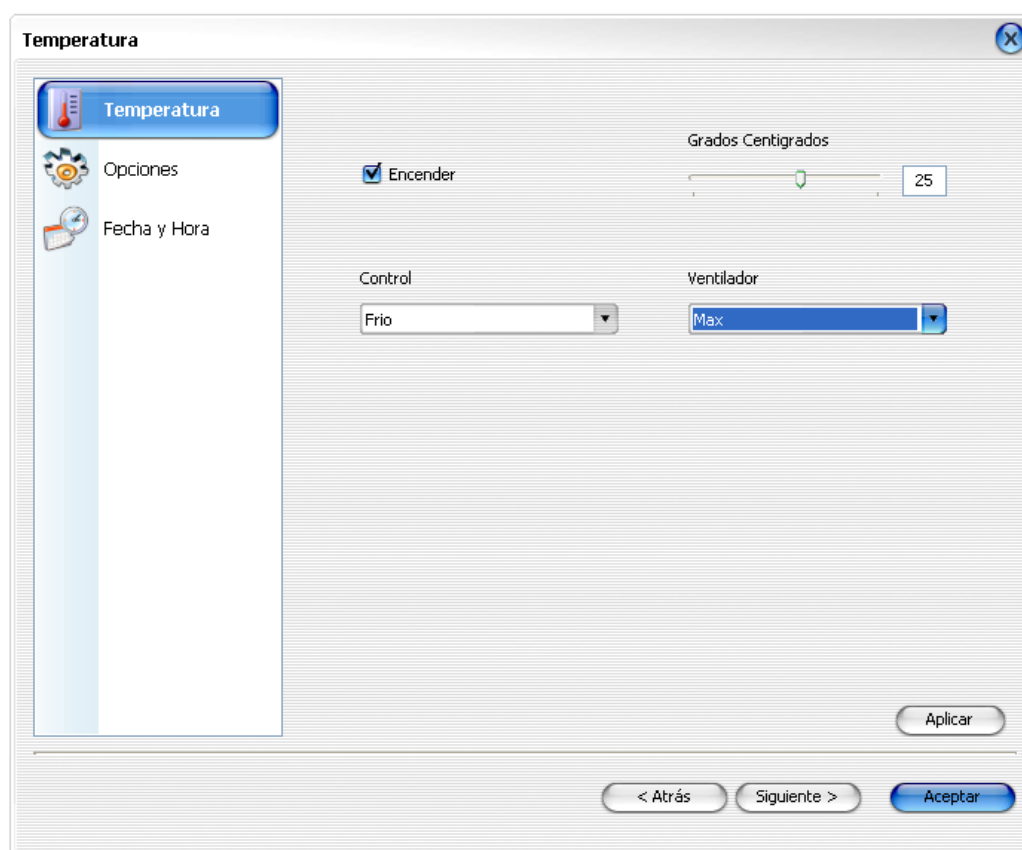


Figura 5.29

- Opciones: En esta página podremos elegir cuáles serán los estados para los que queremos programar las tareas de la agenda y como queremos que se ejecuten. Para elegir un estado solo tenemos que marcar el checkbox correspondiente. Y para elegir el modo de ejecución tenemos tres opciones (Figura 5.30):

- Una vez: la tarea programada solo se ejecutara en el día y hora programada.
- Diariamente: la tarea se ejecutara diariamente en el horario indicado.
- Semanalmente: la tarea se ejecutara semanalmente en el horario establecido y en los días seleccionados.

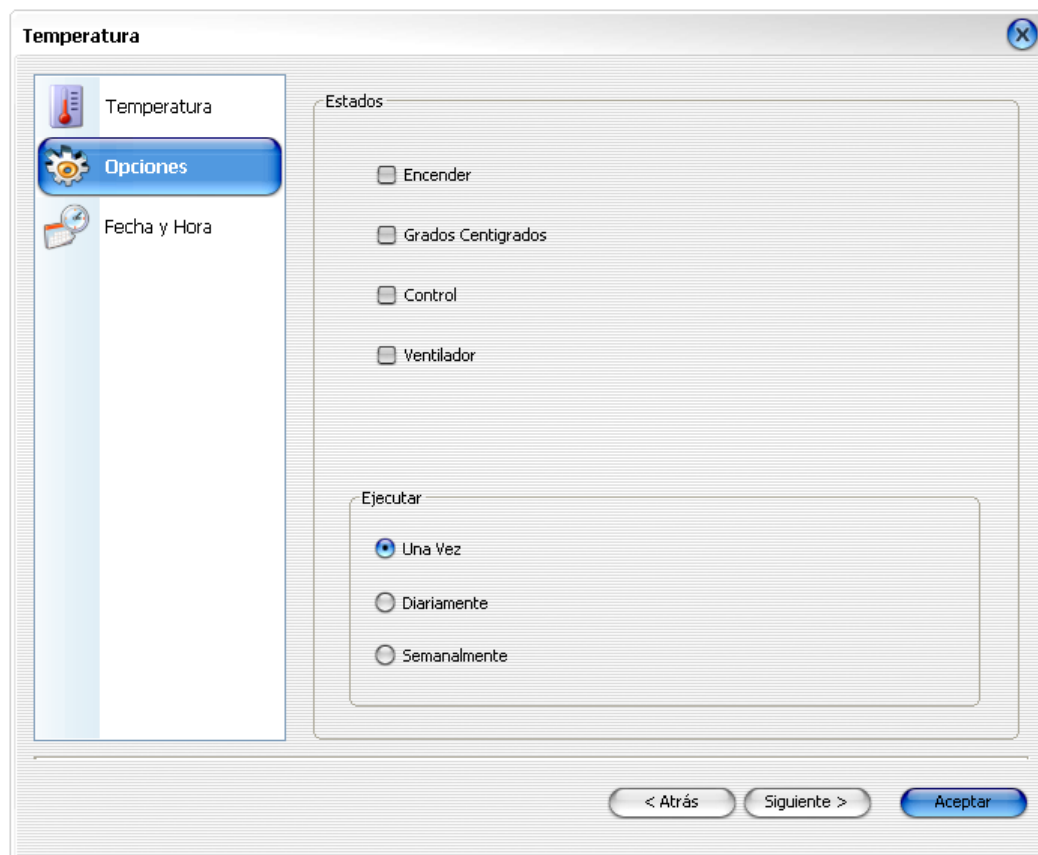


Figura 5.30

- Fecha y hora: Aquí indicaremos la fecha y hora de ejecución de las tareas. Esta página será diferente dependiendo de la opción de ejecución que hayamos elegido en la página de opciones.

Hay que tener en cuenta que el botón “Agregar Tarea” estará deshabilitado a menos que hayamos elegido algún estado en la página de opciones. Si la opción de ejecución fue “Semanalmente” aparte de elegir un estado, habrá que seleccionar al menos un día.

La hora y fecha ingresadas en las tareas que se ejecutan una sola vez, tienen que ser mayor que la hora y fecha actual.

A continuación se mostrara las tres clases de páginas que nos podemos encontrar:

Una vez:

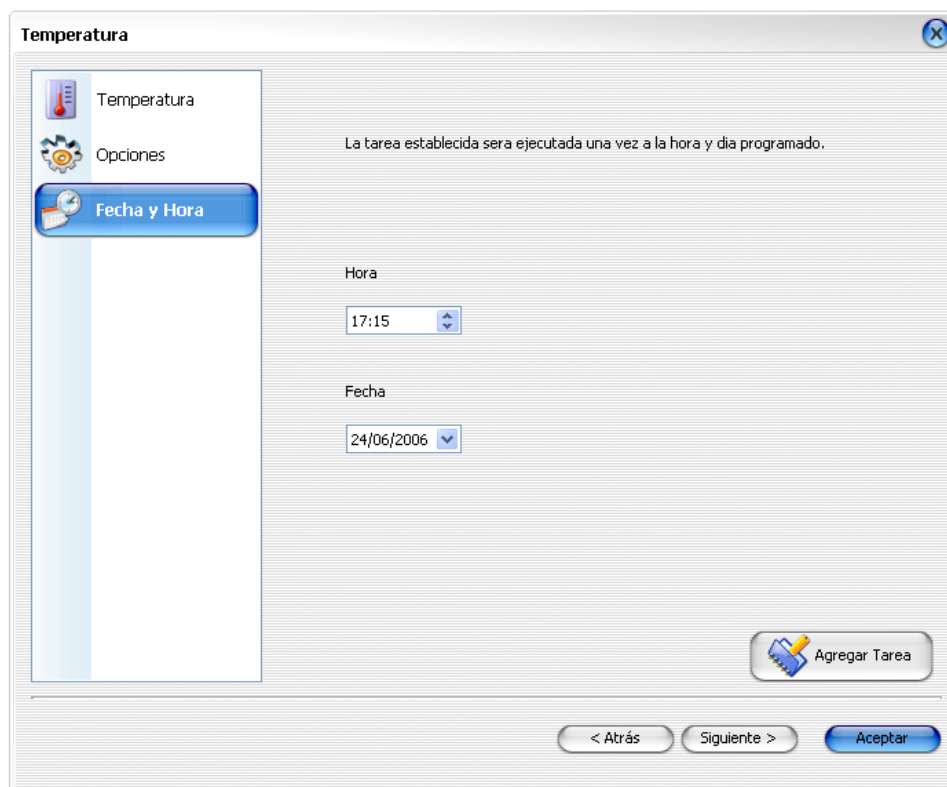


Figura 5.31

Diariamente:

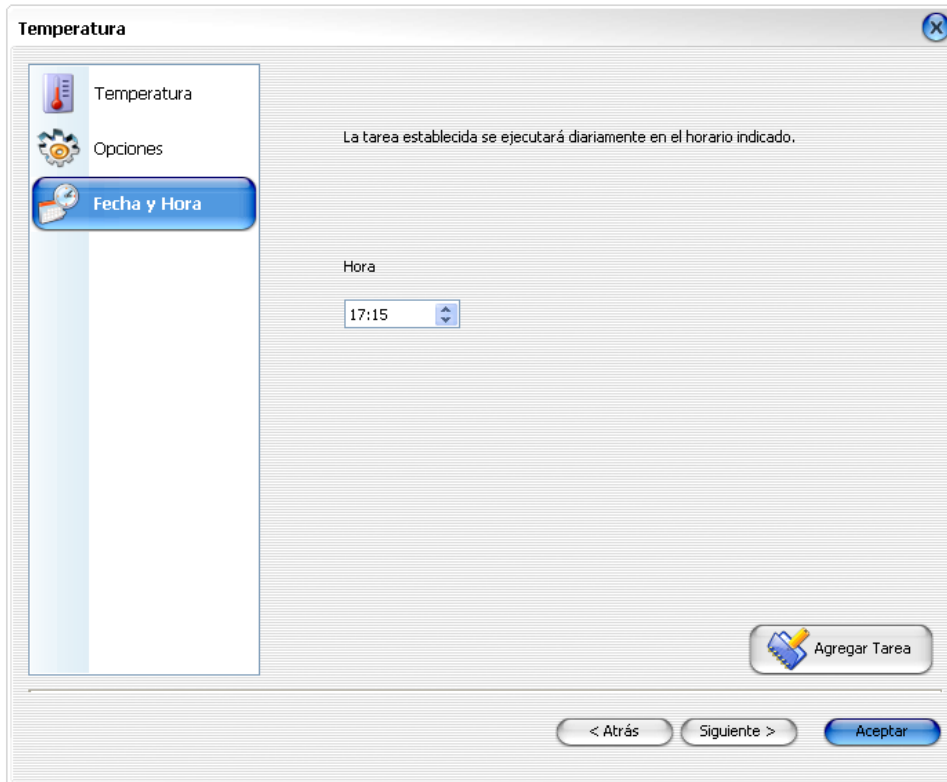


Figura 5.32

Semanalmente:

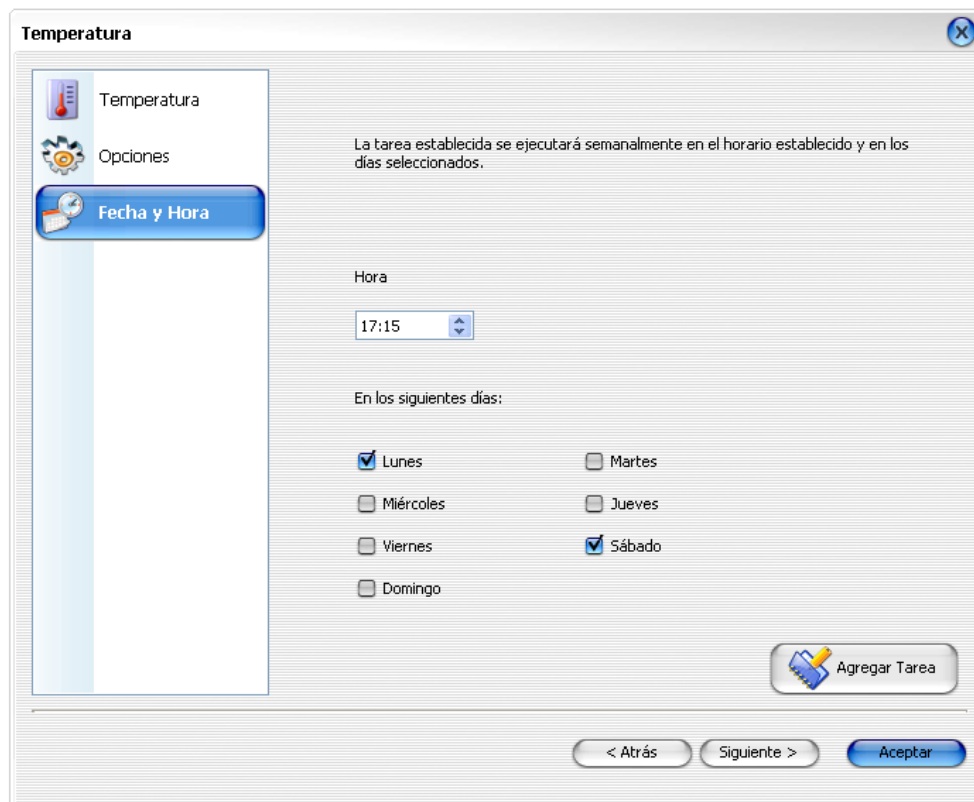


Figura 5.33

### 5.3.- Área de trabajo del acceso remoto

El acceso remoto nos va a proporcionar las mismas funcionalidades que la aplicación de escritorio. Para acceder remotamente a la aplicación tendremos que indicar en nuestro navegador la dirección ip y puerto donde se encuentra escuchando el servidor. (Figura 5.34)

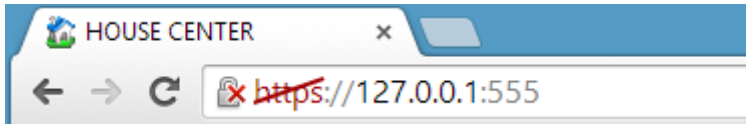


Figura 5.34

Cuando la conexión ha sido establecida con el servidor, este nos requerirá un nombre de usuario y una contraseña para poder acceder.

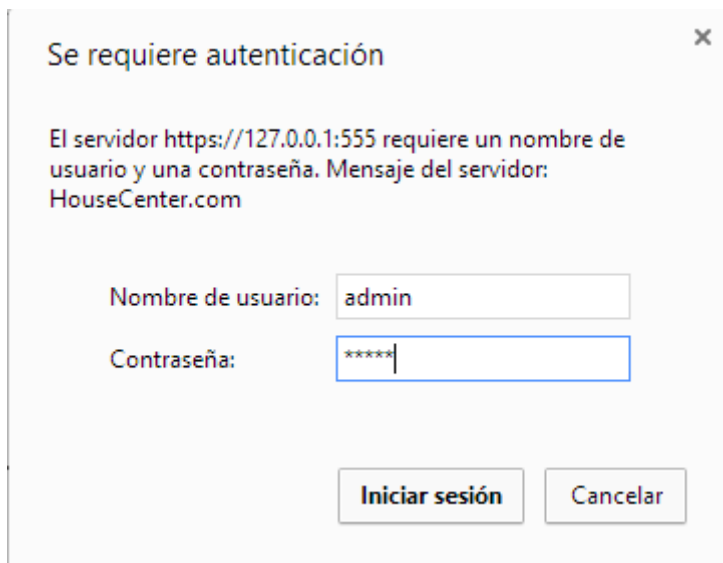


Figura 5.35

Una vez que nos hemos conectado con el servidor, el área de trabajo de la aplicación presenta el siguiente aspecto: (Figura 5.36)

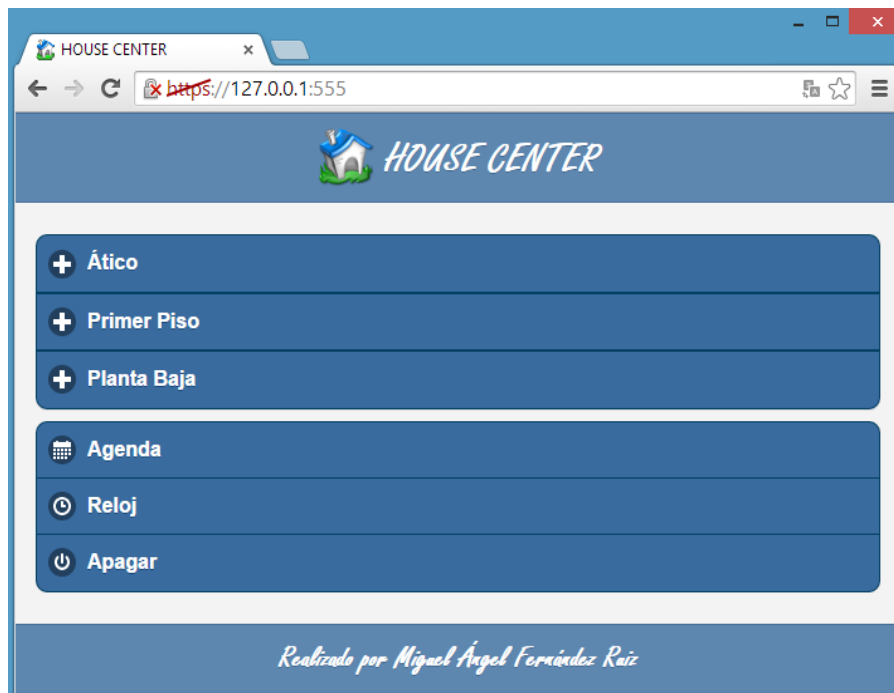


Figura 5.36

Se puede observar en la imagen las distintas opciones que se nos presentan, pudiendo desplazarnos por las distintas plantas del edificio, y por las distintas habitaciones, viendo en cada una de estas habitaciones que sensores o actuadores se encuentran localizados. (Figura 5.37)

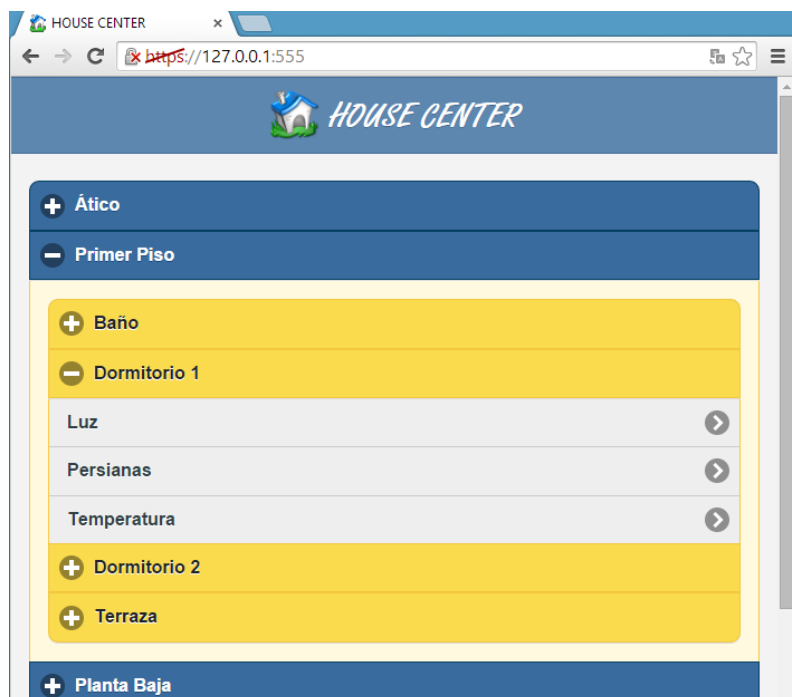


Figura 5.37

### Consulta y control de dispositivos:

Una vez seleccionado un componente en cuestión, podremos modificar sus propiedades de igual forma que podríamos realizarlo con la aplicación de escritorio. (Figura 5.38)

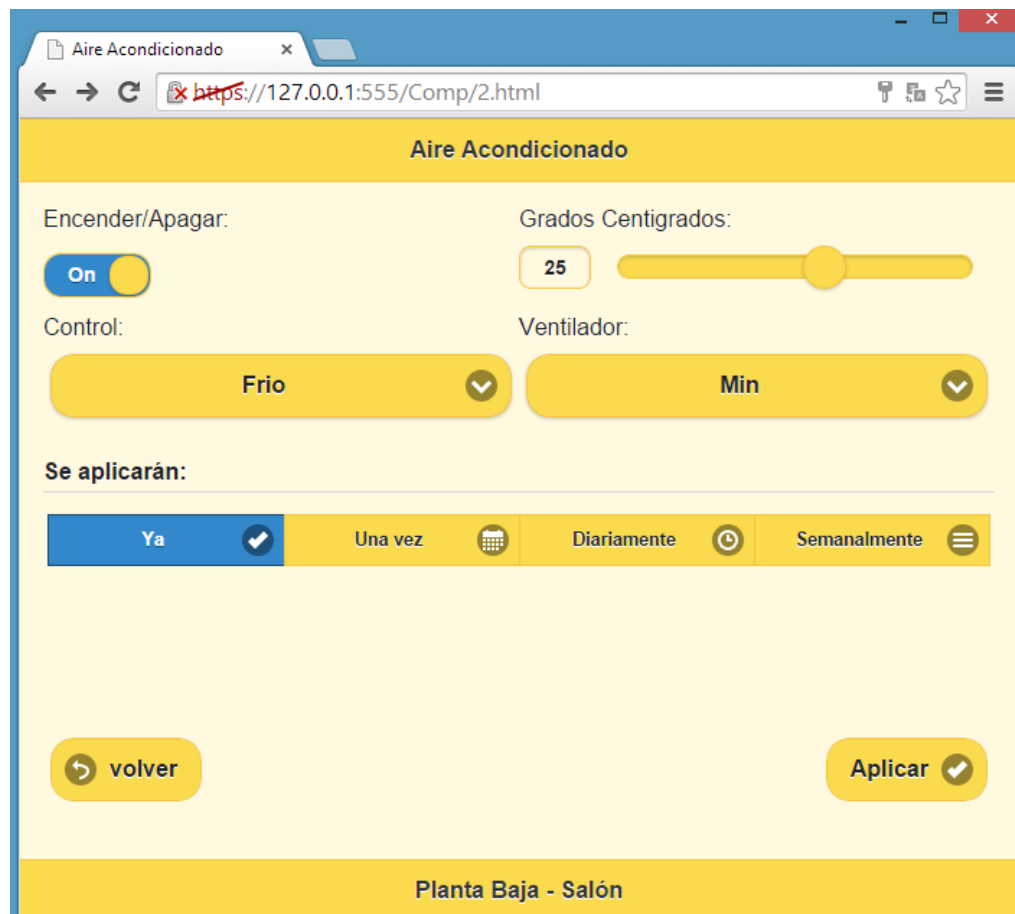


Figura 5.38

Así pues, podremos ver cuál es el estado del dispositivo en este preciso instante así como poder cambiar cada uno de los estados. El título de la página dependerá de cual sea el nombre del dispositivo, y en el pie de página tendremos la localización física del dispositivo. Para cambiar los estados tendremos tres tipos de controles. Un checkbox, un slider que va acompañado de un control edit, y por ultimo una lista desplegable. (Figura 5.39)



Figura 5.39

También se ha proporcionado un control static para aquellos estados que sean de solo lectura. Hay que tener en cuenta que si el dispositivo solo tuviese estados de solo lectura, las opciones “Ya”, “Una Vez”, “Diariamente” y “Semanalmente” no estarán disponibles ya que en un dispositivo de este estilo no se permite su modificación ni la programación de tareas. (Figura 5.40)

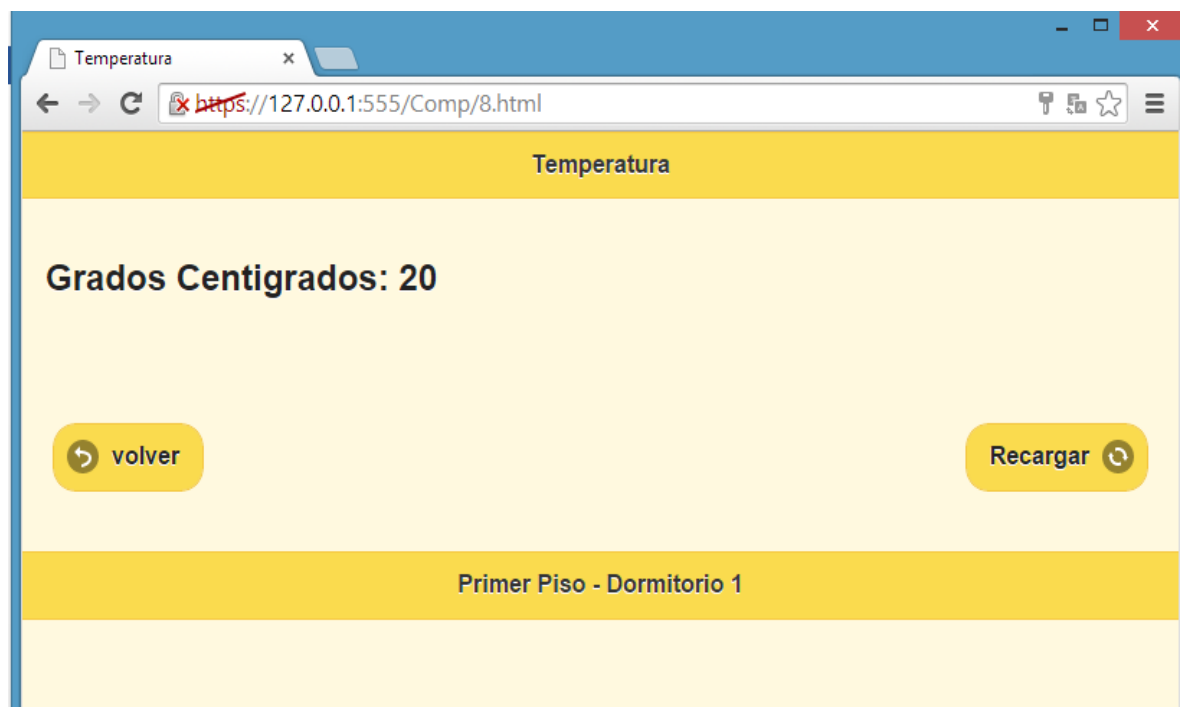


Figura 5.40

Una vez elegidos los valores que queremos asignarle a los estados, solo tendremos que seleccionar la opción “Ya” y pulsar el botón “aplicar” para ejecutar los cambios inmediatamente.



Para poder programar tendremos que elegir cuáles serán los estados para los que queremos programar las tareas de la agenda y como queremos que se ejecuten. Para elegir un estado solo tenemos que marcar el checkbox correspondiente. Y para elegir el modo de ejecución tenemos tres opciones:

- Una vez: la tarea programada solo se ejecutara en el día y hora programada.
- Diariamente: la tarea se ejecutara diariamente en el horario indicado.
- Semanalmente: la tarea se ejecutara semanalmente en el horario establecido y en los días seleccionados

Una vez: (figura 5.41)

The screenshot shows a user interface for 'Aire Acondicionado' (Air Conditioning) with the following elements:

- Encender/Apagar:** A toggle switch set to 'On'.
- Grados Centigrados:** A temperature slider set to 25.
- Control:** A dropdown menu set to 'Frio'.
- Ventilador:** A dropdown menu set to 'Min'.
- Los estados:** A row of four buttons: 'Encender/Apagar' (highlighted in blue), 'Grados Centigrados', 'Control', and 'Ventilador'.
- Se aplicarán:** A row of four buttons: 'Ya' (with a checkmark), 'Una vez' (highlighted in blue with a calendar icon), 'Diariamente' (with a clock icon), and 'Semanalmente' (with a list icon).
- A la fecha y hora:** A text input field containing '02/09/2014 07:58'.
- Navigation:** 'volver' (return) and 'Aplicar' (apply, with a checkmark) buttons at the bottom.

Figura 5.41

Diariamente: (figura 5.42)

### Aire Acondicionado

Encender/Apagar:

On

Grados Centigrados:

25

Control:

Frio ▼

Ventilador:

Min ▼

**Los estados:**

Encender/Apagar

Grados Centigrados

Control

Ventilador

**Se aplicarán:**

Ya

Una vez

Diariamente

Semanalmente

**A la hora:**

07:58

↶ volver

Aplicar

Figura 5.42

Semanalmente: (Figura 5.43)

### Aire Acondicionado

Encender/Apagar:

On

Grados Centigrados:

25

Control:

Frio ▼

Ventilador:

Min ▼

**Los estados:**

Encender/Apagar

Grados Centigrados

Control

Ventilador

**Se aplicarán:**

Ya

Una vez

Diariamente

Semanalmente

**A la hora:**

07:58

**Los días:**

Lunes

Martes

Miércoles

Jueves

Viernes

Sábado

Domingo

↶ volver

Aplicar

Figura 5.43

**Fecha y Hora:** Permite sincronizar el reloj de la central domótica con el reloj del PC. Para ello haremos uso de un cuadro de dialogo que nos mostrara la hora de la central y así poder decidir si queremos sincronizarla o no.



Figura 5.44

Una vez sincronizada la hora de la central con el PC no se permite una nueva sincronización hasta que no se vuelva a cargar el cuadro de dialogo para así no saturar a la central con múltiples cambios de hora.

**Agenda:** Nos permite poder manipular la agenda del centro domótico. Para ello haremos uso de una hoja de propiedades, que nos permitirá acceder a las respectivas páginas pulsando en la lista situada en la parte inferior de la página.

En la agenda podemos distinguir cuatro páginas:

- **Memoria:** Esta página nos permite grabar la memoria hardware del sistema. Para ello dispondremos de cuatro opciones: (Figura 5.45)
  1. Mensual: La agenda durara cuatro semanas. Para generarla se tienen en cuenta las tareas que se ejecutan una sola vez, las que se ejecutan semanalmente y las tareas que se ejecutan diariamente.
  2. Semanal: La agenda de la memoria será semanal. Para generarla se tienen en cuenta las tareas que se ejecutan una sola vez, las que se ejecutan semanalmente y las tareas que se ejecutan diariamente.
  3. Semanal Repetitiva: La agenda será semanal y se repetirá indefinidamente. Para generarla solo se tienen en cuenta las tareas que se ejecutan semanalmente.
  4. Diaria Repetitiva: La agenda será diaria y se repetirá todos los días indefinidamente. Para generarla solo se tienen en cuenta las tareas que se ejecutan diariamente.



Figura 5.45

- **Una Vez:** Aquí se listarán las tareas que se repiten solo una vez, es decir, la tarea programada solo se ejecutará en el día y hora programada. Podremos borrar todas las tareas que creamos oportunas, simplemente pulsando en el botón eliminar situado a la derecha de cada una de las tareas.

AGENDA							
Dispositivo	Estado	Acción	Fecha	Hora	Planta	Habitación	Eliminar
Aire Acondicionado	Control	Frio	02/09/2014	15:30	Planta Baja	Habitación	
Aire Acondicionado	Encender/Apagar	Encender	02/09/2014	15:30	Planta Baja	Habitación	
Aire Acondicionado	Ventilador	AUTO	18/09/2014	08:04	Planta Baja	Habitación	
Aire Acondicionado	Encender/Apagar	Encender	10/10/2014	20:36	Planta Baja	Salón	
Aire Acondicionado	Encender/Apagar	Encender	17/10/2014	19:55	Planta Baja	Salón	

[← Volver](#)

Memoria Una Vez Semanal Diaria

Figura 5.47

- **Semanalmente:** Aquí se listarán las tareas que se repiten semanalmente, es decir, que si tenemos una tarea programada para los días Lunes, Martes, y Miércoles la tarea se ejecutará esos días todas las semanas. Aquí, al igual que en la anterior página, también dispondremos en cada una de las tareas de un botón que nos permitirá eliminar las tareas que deseemos.

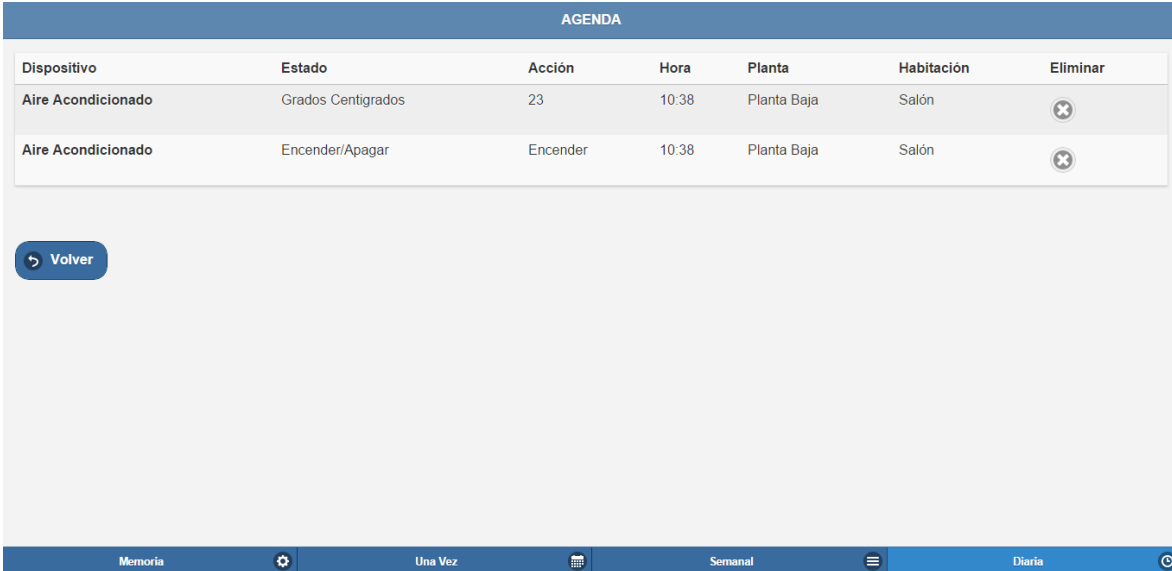
AGENDA							
Dispositivo	Estado	Acción	Hora	Días	Planta	Habitación	Eliminar
Aire Acondicionado	Encender/Apagar	Encender	00:02	Miércoles, Jueves, Domingo	Planta Baja	Salón	
Temperatura	Control	Calor	01:44	Lunes, Viernes, Sábado	Planta Baja	Salón	
Temperatura	Encender/Apagar	Encender	01:44	Lunes, Viernes, Sábado	Planta Baja	Salón	
Aire Acondicionado	Encender/Apagar	Encender	19:43	Lunes	Planta Baja	Salón	
Aire Acondicionado	Grados Centígrados	25	19:43	Jueves	Planta Baja	Salón	
Aire Acondicionado	Encender/Apagar	Encender	19:46	Sábado	Planta Baja	Salón	
Aire Acondicionado	Control	Calor	19:55	Lunes, Domingo	Planta Baja	Salón	

[← Volver](#)

Memoria Una Vez Semanal Diaria

Figura 5.48

- **Diariamente:** Aquí se listan las tareas que se repiten diariamente, es decir, las tareas se ejecutaran a la hora programada todos los días. De forma similar a lo explicado anteriormente, también se dispondrá de un botón de eliminar tarea.



The screenshot shows a mobile application interface titled 'AGENDA'. It features a table with the following columns: 'Dispositivo', 'Estado', 'Acción', 'Hora', 'Planta', 'Habitación', and 'Eliminar'. There are two rows of data. Below the table is a blue button labeled 'Volver' with a left-pointing arrow. At the bottom, there is a navigation bar with icons for 'Memoria', 'Una Vez', 'Semanal', and 'Diaria'.

Dispositivo	Estado	Acción	Hora	Planta	Habitación	Eliminar
Aire Acondicionado	Grados Centigrados	23	10:38	Planta Baja	Salón	
Aire Acondicionado	Encender/Apagar	Encender	10:38	Planta Baja	Salón	

Figura 5.49

Para cada una de estas páginas, si el tamaño de pantalla del dispositivo impide que se pueda ver la totalidad de las columnas horizontalmente, automáticamente cambiará a una vista vertical agrupadas por grupos de tareas: (Figura 5.50)



Figura 5.50

**Apagar:** Nos permite poder apagar el servidor web remotamente. Para ello se hace uso de un cuadro de dialogo que nos proporcionara la funcionalidad: (Figura 5.51)





Figura 5.51

#### 5.4.- Manual de instalación

En esta sección se explicara cómo se deben configurar y preparar los diferentes elementos que utiliza el programa. La configuración del programa la deberá realizar el instalador del sistema, ya que es este el que posee los conocimientos necesarios para ello.

##### **Paso 1 – Preparación de la lista de imágenes.**

El programa utiliza para su funcionamiento una lista de imágenes. Esta lista es utilizada por la vista de árbol, los botones que lanzan la hoja de propiedades, y la misma hoja de propiedades.

La lista estará formada por imágenes de dimensión 32 píxeles de alto por 32 píxeles de ancho.

Una vez que tengamos preparadas las imágenes que queramos añadir a la lista, estas se guardaran consecutivamente de izquierda a derecha en el fichero “Arbol.bmp”.

El color que tomara el programa como transparente será el RGB(255,0,255).

A cada imagen de la lista se le asignara un número, así, la imagen situada más a la izquierda tendrá el número 0, la siguiente el 1, y así sucesivamente. La imagen con el número 0 será la imagen utilizada como raíz del árbol.

Al no estar incluida la lista de imágenes en el archivo de recursos de la aplicación, el instalador podrá incluir o eliminar las imágenes que desee sin necesidad de generar otra copia del programa. (Figura 5.52)



Figura 5.52

##### **Paso 2 - Preparación de los planos.**

Aquí se explicara cómo se debe preparar los planos que se visualizaran en la vista.

La vista se comporta como un mapa de bits sensible. Así cuando lo mostrado en la vista es un plano y movemos el puntero del ratón por encima de la vista, veremos que aparecen unos tooltips que nos indicaran el destino al que nos dirigiríamos si pulsásemos el botón izquierdo del ratón.

Pues bien, para conseguir esto se utilizan dos imágenes. Una que será la imagen que se verá en la vista, y otra que estará en segundo plano pero que será la que utilizara el programa para saber hacia dónde dirigirse. Así, que si queremos que una determinada región del plano sea sensible a un clic del ratón, solo tendremos que pintar esa región en la imagen que está en segundo plano de un determinado color.

Hay que tener en cuenta que las dos imágenes tienen que tener las mismas dimensiones, y que los colores asignados a cada región deben ser diferentes. Los colores utilizados para cada región serán del tipo RGB, lo que ofrece hasta 16.7 millones de posibilidades por lo que nunca se podrá agotar la gama de colores que identifican la distintas zonas del plano.

Veamos esto con un ejemplo:



Figura 5.53

Supongamos que tenemos el plano de la figura 5.53. Si queremos que el baño, el salón, la cocina y el dormitorio sea sensibles al movimiento y al click del ratón, solo tendremos que asignar a cada una de esas regiones un color diferente.(Figura 5.54).

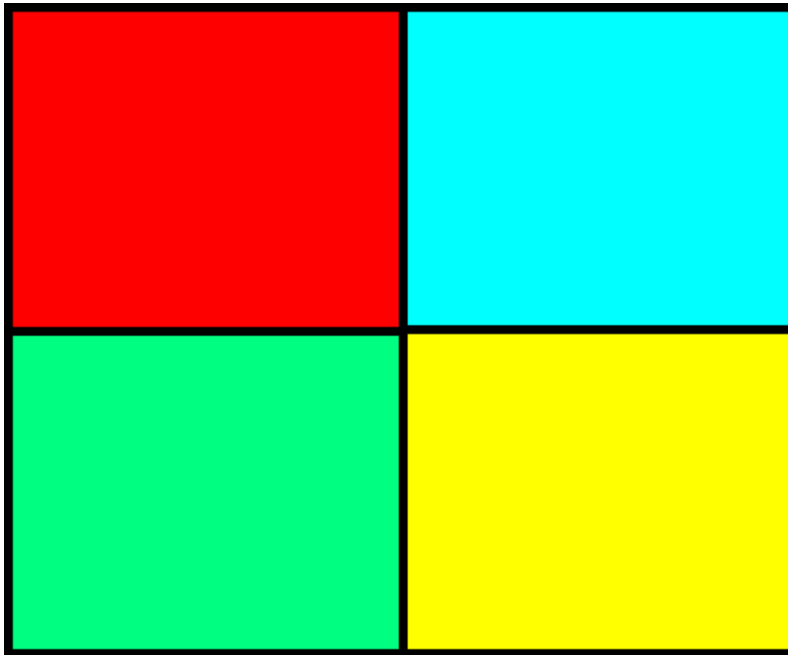


Figura 5.54

Los planos que se hayan preparado se guardaran en el directorio “Planos” situado en el directorio de la aplicación.

Los planos serán imágenes del tipo “bmp” y se guardaran con el nombre de la planta a la que pertenecen si son el plano que se mostrara en la vista, y con el nombre de la planta seguido de un “1” si son las imágenes que están es segundo plano. Por ejemplo, si estamos preparando el plano de la “Planta Baja”, las imágenes se guardaran en los ficheros “Planta Baja.bmp” y “Planta Baja1.bmp”.

### **Paso 3 - Preparación de las fotos de las habitaciones.**

Se deberá preparar para cada habitación de la vivienda una foto, que se mostrara en la vista cada vez que se seleccione en el árbol la habitación. Las fotos de las habitaciones se guardaran en el mismo directorio en el que se guardan los planos con el nombre que desee el instalador.

#### **Paso 4 - Obtención de la direcciones RS485 de los dispositivos.**

Es decir, la dirección que le corresponde a cada dispositivo en el bus RS485. La dirección es un numero entre 1 – 255, ya que contamos con 1 byte de dirección y la dirección 0 está reservada para el módulo *maestro* (central).

#### **Paso 5 – Introducción de la información en las bases de datos.**

El programa utiliza para su funcionamiento bases de datos del tipo Microsoft Access. Estas bases de datos se utilizaran para obtener información de la configuración de la vivienda, así como para introducir y manipular datos relativos a la agenda del sistema.

- ***BaseD.mdb***

En esta base de datos se guardara la información relativa a la configuración del sistema, es decir, tiene información de donde se encuentran situados los dispositivos, sus direcciones en el bus RS485, la estructura de la casa, los planos y fotos que debe cargar la vista, etc...

La base de datos está formada por las tablas “Casa”, “Componentes”, y las restantes tablas dependerán de cuantas plantas tenga la vivienda.

- **Tabla “Casa”:** (Tabla 5.1)

En esta Tabla se almacenara:

- **Plantas (Tipo Texto):** El nombre de cada una de las plantas que tiene la vivienda. Ejemplo: Planta Baja, Primer Piso etc.
- **Icono (Tipo número):** El icono que queremos asignarle a la planta y que se verá en la vista del árbol. Este icono se elegirá de la lista de imágenes (Ver Paso 1). Por ejemplo, si tenemos una planta baja, y a esa planta le queremos asignar el icono de la lista de imágenes que es una B, tendremos que introducir en la base de datos el

número 12, ya que es el número que le corresponde en la lista de imágenes. (Figura 5.55)



Figura 5.55

- Por ultimo habrá que introducir el color que se le asignó a la planta en la preparación del plano, es decir, que si a la “Planta Baja” se le asignó el color rojo ( RGB(255,0,0)) cuando se preparó los planos, se deberá introducir en el campo “rojo” el número 255, y en el “verde” y “azul” el número 0. El campo será tipo número.(Tabla 5.1)

	Plantas	Icono	Rojo	Verde	Azul
▶	Atico	11	255	0	0
	Planta Baja	12	0	0	255
	Primer Piso	2	0	255	0
	Sótano	13	255	255	0
*		0	0	0	0

Registro: 1 de 4

Tabla 5.1

- A continuación se agregara tantas tablas como plantas haya en la vivienda.

En estas tablas se almacenara:

- Habitaciones (Tipo Texto):** El nombre de cada una de las habitaciones que haya en la planta.
- PathImagen (Tipo Texto):** El nombre del archivo donde podemos encontrar la imagen de la habitación.
- Icono (Tipo número):** El icono que queremos asignarle a la planta y que se verá en la vista del árbol.

- Y por último habrá que introducir el color que se le asignó a la habitación en la preparación del plano. Cada campo será del tipo número.

Ejemplo:

Habitaciones	PathImagen	Icono	Rojo	Verde	Azul
Baño	Baño.bmp	16	255	0	0
Cocina	Cocina.bmp	14	0	255	0
Despacho	Despacho.bmp	17	0	0	255
Garage	Garage.bmp	22	128	128	255
Jardín	Jardin.bmp	18	255	128	64
Salón	Salón.bmp	15	255	255	0
*		0	0	0	0

Registro: 1 de 6

Tabla 5.2

#### - Tabla “Componentes”: (Tabla 5.3)

En esta tabla se almacenara toda la información relativa a los dispositivos que hay instalados en la vivienda:

- **Dirección (Tipo número):** Dirección RS485 del dispositivo.
- **Componente (Texto):** Nombre del dispositivo.
- **Planta (Texto):** Planta en la que está instalado el dispositivo.
- **Habitación (Texto):** Habitación en la que está instalado el dispositivo.
- **Icono (Número):** Numero del icono que queremos asignarle al dispositivo.
- **EjeX (Número):** Coordenada del eje de abscisas de la vista donde se mostrara el icono.
- **EjeY (Número):** Coordenada del eje de ordenadas de la vista donde se mostrara el icono.

- **Estado 0,..., Estado 7 (Texto):** Aquí se deberá de introducir la información que la primera página de la hoja de propiedades del dispositivo usara al crear los controles, para ello, la información introducida tiene que seguir un determinado formato dependiendo del control a utilizar. Los estados que no se utilicen de deberán dejar en blanco. Disponemos de 4 controles:

- **Slider:** El estado será controlado por un control slider acompañado de un control edit. El formato que debe seguirse es:

Nombre del Estado,Tipo de control, Valor maximo,Valor mínimo

Nombre del estado: Indica cual es el nombre del estado.

Tipo de control: Indica el control a utilizar, en este caso es sl.

Valor máximo: El valor máximo que tomara el control slider.

Valor mínimo: Valor mínimo que tomara el control slider.

Ejemplo: Grados centigrados,sl,18,30

- **CheckBox:** El estado será controlado por un check box.  
Formato: Nombre del estado,Tipo de control,Vncheck,Vcheck

Nombre del estado: Indica cual es el nombre del estado.

Tipo de control: En este caso ch.

Vncheck: Valor usado cuando el check box no este marcado.

Vcheck: Valor usado para marcar el checkbox.

Ejemplo: Encender/Apagar,ch,15,50

- **Lista desplegable:** El estado será controlado por una lista desplegable. Formato: Nombre del estado,Tipo de control,Tabla

Nombre del estado: Indica cual es el nombre del estado.



Tipo de control: li.

Tabla: Indica el nombre de la tabla de la base de datos

“Lista.mdb” que contiene la información de la lista.

Esta base de datos se explicara posteriormente.

Ejemplo: Control,li,Persianas

- **Static:** Este control deberá utilizarse para los estado que sean de solo lectura, ya que su única función será la de mostrar información.

Formato: Nombre del estado, Tipo de control

Nombre del estado: Indica cual es el nombre del estado.

Tipo de control: En este caso et.

Ejemplo: Grados centigrados,et

Direccion	Componente	Planta	Habitacion	Icono	EjeX	EjeY	Estado 0	Estado 1	Estado
	1 Luz	Planta Baja	Salón	24	100	100		Nivel Intensidad,sl,0,20	
	2 Temperatura	Planta Baja	Salón	27	140	140	Encender,ch,10,25	Grados Centigrados,sl,18	Control
	3 Persianas	Planta Baja	Salón	29	180	180		Subir/Bajar,sl,0,30	
	4 Luz	Sótano	Habitación 1	24	100	100			
	5 Luz	Sótano	Habitación 2	24	100	100		Encender,ch,10,25	
▶	6 Luz	Primer Piso	Baño	24	100	100	Encender,ch,10,25		
	7 Luz	Primer Piso	Dormitorio 1	24	100	100			Encender
	8 Temperatura	Primer Piso	Dormitorio 1	27	150	150		Grados Centigrados,et	
	9 Luz	Ático	Habitación 1	24	100	100	Encender,ch,10,25		
	10 Cortinas	Planta Baja	Salón	28	220	220		Abrir Cerrar,sl,0,30	
	11 Ventanas	Planta Baja	Salón	26	300	300			
	12 Cerradura	Planta Baja	Salón	25	350	350	Abrir/Cerrar,ch,25,14		
	13 Persianas	Primer Piso	Dormitorio 1	29	200	200		Abrir/Cerrar,sl,0,14	
	14 Riego	Planta Baja	Jardín	31	100	100	Encender Aspersores,ct		
*	0			0	0	0			

Tabla 5.3

- **Lista.mdb**

En esta base de datos se guarda la información que se utilizara en la creación de las listas desplegables. Se ha decidido no crear las tablas en la base “BaseD.mdb” para tener una mayor claridad.

Ya que los estados trabajan con valores numéricos, se ha creído conveniente dar la opción de asociar a cada valor que puede tomar un estado un nombre, ya que para una persona es más sencillo recordar un nombre que un número. Así que si se quiere utilizar esta opción solo habrá que utilizar un control “lista desplegable” y rellenar la tabla correspondiente.

Cada tabla estará formada por:

“Valor”: Campo de tipo número. En él se guarda los valores que se quieren asociar.

“Estado0”,...,“Estado7”: Campos de tipo texto. En estos campos se almacenara el nombre que se le asigna al valor. Si a un determinado valor no se le quisiera asignar un nombre, el nombre se dejara en blanco.

Un ejemplo de una posible tabla podría ser este: (Tabla 5.4)

	Valor	Estado2	Estado3
▶	0	AUTO	AUTO
	10	Frio	Max
	25	Calor	Min
	30	Humedad	Low
	40	Ventilador	
*	0		

Registro: 1 de 5

Tabla 5.4

Donde se puede ver como a cada valor se le asocia un nombre.

- **Agenda.mdb**

Base de datos encargada del almacenamiento y manipulación de la agenda. Aunque el instalador no deberá introducir ninguna información en ella, ya que de esto se ocupara internamente el programa, es conveniente definir su estructura.

La base de datos estará formada por cuatro tablas:

- **Tabla “Una”:** En ella se guardan las tareas que se repiten una sola vez. Está formada por los siguiente campos: (Tabla 5.5)

**Dirección (Tipo número):** Dirección RS485 del dispositivo.

**Planta (Texto):** Planta en la que está instalado el dispositivo.

**Habitación (Texto):** Habitación en la que está instalado el dispositivo.

**Componente (Texto):** Nombre del dispositivo.

**Nombre Estado (Texto):** Nombre del estado.

**Número Estado (Numero):** Numero de estado.

**Acción (Texto):** Acción que se debe ejecutar.

**Fecha (Fecha/ Hora):** Fecha y hora en la que se ejecutara la tarea.

	Dirección	Planta	Habitación	Componente	Nombre Estado	Número Estado	Acción	Fecha
▶	2	Planta Baja	Salón	Temperatura	Grados Centigrados	1	25,25	30/06/2006 7:51:00
	2	Planta Baja	Salón	Temperatura	Control	2	Calor,25	27/06/2006 21:12:00
*	0					0		

Tabla 5.5

- **Tabla “Diaria”:** En ella se guardan las tareas que se repiten diariamente. Está formada por los siguiente campos: (Tabla 5.6)

**Dirección (Numero):** Dirección RS485 del dispositivo.

**Planta (Texto):** Planta en la que está instalado el dispositivo.

**Habitación (Texto):** Habitación en la que está instalado el dispositivo.

**Componente (Texto):** Nombre del dispositivo.

**Nombre Estado (Texto):** Nombre del estado.

**Número Estado (Número):** Numero de estado.

**Acción (Texto):** Acción que se debe ejecutar.

**Hora (Hora corta):** Hora en la que se ejecutara la tarea.

	Dirección	Planta	Habitación	Componente	Nombre Estado	Número Estado	Acción	Hora
▶		2	Planta Baja	Salón	Temperatura	Grados Centigr:	1 25,25	1:01
*		0				0		

Registro: 1 de 1

Tabla 5.6

- **Tabla “Semanal”:** En ella se guardan las tareas que se repiten semanalmente. Está formada por los siguiente campos: (Tabla 5.7)

**Dirección (Numero):** Dirección RS485 del dispositivo.

**Planta (Texto):** Planta en la que está instalado el dispositivo.

**Habitación (Texto):** Habitación en la que está instalado el dispositivo.

**Componente (Texto):** Nombre del dispositivo.

**Nombre Estado (Texto):** Nombre del estado.

**Número Estado (Numero):** Numero de estado.

**Acción (Texto):** Acción que se debe ejecutar.

**Hora (Hora corta):** Hora en la que se ejecutara la tarea.

**Días (Texto):** Días de la semana en los que se ejecutara la tarea.

Dirección	Planta	Habitación	Componente	Nombre Estado	Número Estado	Acción	Hora	Dias
2	Planta Baja	Salón	Temperatura	Grados Centigr:	1	25,25	4:01	Lunes,Sábado,Domingo
*	0				0			

Tabla 5.7

- **Tabla “Buffer”**: Esta tabla se utiliza como un punto de transición antes de enviar los datos a la memoria Hardware. Debe de tener cuatro campos: “Dirección”, “NumEstado”, “Acción”, “Fecha” todos ellos de tipo numérico. (Tabla 5.8)

Dirección	NumEstado	Acción	Fecha

Tabla 5.8

#### 5.4.1.- Creación del certificado digital

Debido a que la aplicación utiliza un servidor HTTP con SSL para poder proporcionar un acceso seguro al centro domotico, es requerido la creacion de un certificado digital que se alojara en el mismo directorio de la aplicación con el nombre de archivo “server-key-cert.pem”

Un certificado SSL es un certificado digital utilizado por el protocolo para el encriptamiento de la información. Este certificado es proporcionado por un proveedor autorizado (Verisign, Thawte, Comodo, etc...) y es enviado a el cliente por el servidor con quien estamos estableciendo una conexión segura.

El certificado que vamos a crear tiene el mismo nivel de encriptamiento que cualquiera de estos proveedores autorizados puede entregar, simplemente no aparecemos en la lista preestablecida en nuestro navegador, y por lo tanto desconfiará del mismo.

## **Paso 1: Instalar OPENSSL**

Lo primero que debemos hacer es descargarnos la distribución [de OpenSSL para Windows](#). [OpenSSL](#) es un software open source para libre descarga que incluye un paquete de herramientas de administración y bibliotecas relacionadas con la criptografía, que suministran funciones criptográficas a otros paquetes como OpenSSH y navegadores web para acceso seguro a sitios con el protocolo HTTPS, herramientas que ayudan al sistema a implementar el Secure Sockets Layer (SSL). (Figura 5.56)

File	Type	Description
<a href="#">Win32 OpenSSL v1.0.1i Light</a>	1MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v1.0.1i (Recommended for brave users by the creators of <a href="#">OpenSSL</a> ). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
<a href="#">Visual C++ 2008 Redistributables</a>	1.7MB Installer	Having problems with error messages when trying to run OpenSSL? This will likely fix the problem. Only works with Windows 2000 and later. Although there is a "newer version" of this installer, this is the <b>correct</b> version to install.
<a href="#">Visual C++ 2008 Redistributables for Windows 9x/NT4</a>	4.6MB ZIP file	Having problems with error messages when trying to run OpenSSL? This will likely fix the problem. Only use under Windows 95, 98, Me, and NT4. Install this, then install "OpenSSL Light" using the '/bin' option, and then run FixSSL_9xNT4.bat contained within the /bin directory.
<a href="#">Win32 OpenSSL v1.0.1i</a>	16MB Installer	Installs Win32 OpenSSL v1.0.1i (Recommended for software developers by the creators of <a href="#">OpenSSL</a> ). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
<a href="#">Win64 OpenSSL v1.0.1i Light</a>	1MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v1.0.1i (Only install this if you need 64-bit OpenSSL for Windows. Only installs on 64-bit versions of Windows. Note that this is a default

Figura 5.56

## **Paso 2: Generar una Clave Privada**

La llave privada nos será útil para la generación del certificado. Una vez creado nuestro certificado SSL dependerá de esta llave para la implementación del mismo en cualquier servicio que requiera una conexión segura.

Para nuestra aplicación vamos a crear una llave de 1024 bits.

```
openssl genrsa -out server.key 1024
```

## **Paso 3: Generar un CSR (solicitud de firma de certificado)**

Una vez que hemos creado la clave privada, se puede generar una solicitud de firma de certificado. El CSR se puede utilizar entonces de dos maneras. Idealmente, el CSR se enviará a una autoridad de certificación, tal como Thawte o Verisign, quien verificará la

identidad del solicitante y expedirá un certificado firmado. La segunda opción es la de autofirmar el CSR, que se demostrará a continuación.

Durante la generación del CSR, se pedirá varia información. Estos son los atributos X.509 del certificado. Uno de los mensajes nos pedirá el Common Name. Es importante que este campo se rellene con el nombre de dominio completo del servidor que va a estar protegido por SSL. En nuestro caso será `https://HouseCenter.com`. Para generar el CSR hay que ejecutar el siguiente comando, en el cual uno de los parámetros es la llave privada que acabamos de crear::

```
openssl req -new -key server.key -out server.csr
```

#### **Paso 4: Crear un certificado auto-firmado**

En este punto ya podemos generar el certificado auto-firmado. Hay que decir que, evidentemente, este certificado generará un error en el navegador del cliente en el sentido de que la autoridad de certificado de firma es desconocida y no confiable.

Para generar un certificado válido para 365 días, ejecutaremos el siguiente comando:

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

#### **Paso 5: concatenar el certificado auto-firmado**

Una vez obtenido el certificado y la clave, concatenaremos ambos ficheros en uno solo archivo llamado "server-key-cert.pem" el cual se alojara en el mismo directorio de la aplicación.

### *5.5.- Manual de Programación*

#### **5.5.1.- Introducción.**

En el desarrollo de la aplicación ha tenido dos líneas de desarrollo completamente diferentes que veremos a continuación:

#### **Aplicación de escritorio**

Para la programación de la aplicación de escritorio se ha escogido el entorno de programación Visual C++ para desarrollar el proyecto, por ser la herramienta más poderosa y preferida por los programadores en C++. Visual C++ combina la programación orientadas a objetos (C++) y el sistema de desarrollo diseñado especialmente para crear aplicaciones gráficas para Windows (SDK).

Aunque las aplicaciones Windows son sencillas de utilizar, el desarrollo de las mismas no es una tarea fácil. Por ello, para hacer más asequible esta tarea, Visual C++ incluye, además de las herramientas que lo convierten en un generador de programas C++, un conjunto completo de clases ( Microsoft Foundation Class, MFC ) que permite crear de una forma intuitiva las aplicaciones para Windows, y manejar los componentes de Windows según su naturaleza de objetos. Esto es, MFC es una biblioteca de clases que encapsula las funciones de la API de Windows, para crear y manipular objetos Windows.

Así pues, podemos concluir indicando que Visual C++ nos ofrece un entorno muy profesional y cómodo de usar para el desarrollo de proyectos, en donde las aplicaciones resultan fáciles y agradables al programador debido al gran número de componentes existentes.

### **Acceso Remoto**

Debido a que el uso de dispositivos móviles está creciendo a un ritmo imparable, dispositivos como tablets y smartphones han incrementado sus ventas en los últimos años y la navegación en Internet mediante estos dispositivos es cada vez más común, la premisa que se ha seguido en el desarrollo del acceso remoto es que fuera accesible desde cualquier tipo de dispositivo, para así evitar tener que desarrollar múltiples aplicaciones específicas dependiendo de la plataforma (Windows, Linux, IOS, Android,...) o el tipo de dispositivo (tabletas, smartphones, libros electrónicos, portátiles, PCs,...) que se utilice.

Al ser tan heterogéneas las posibilidades de acceso, (plataformas, dispositivos, resoluciones, tamaños de pantalla,...), se ha desarrollado la aplicación teniendo en cuenta la filosofía de **diseño web adaptable** o **adaptativo**, conocido por las siglas **RWD** (del inglés, *Responsive Web Design*) con lo que nuestra aplicación estará adaptada al dispositivo que se esté utilizando para visualizarla.

Por último, y debido a que la seguridad era un premisa sumamente importante, ya que la aplicación otorga el control total de un hogar, se ha implementado un servidor http



embebido con TSL (*Transport Layer Security*) con su correspondiente certificado digital en el lado del servidor, para otorgar a la aplicación de una seguridad robusta.

Para lograr todos estos objetivos se ha optado por desarrollar el acceso remoto utilizando HTML5, estilos CSS en su versión número 3, así como JavaScript en el lado del cliente (client-side), para permitir mejoras en la interfaz de usuario y páginas web dinámicas. Para la comunicación con el servidor se ha utilizado **AJAX**( *Asynchronous JavaScript And XML* ) *permitiendo* la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones. También se ha utilizado **jQuery**, que es una biblioteca de JavaScript , que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Otras de las grandes ventajas de jQuery es que permite discriminar entre los diversos navegadores, por lo que el código Javascript funcionara igual en cada navegador. Con esta librería ya no necesitamos preocuparnos por si el navegador del usuario es Explorer, Chrome, Firefox, etc. sino que la propia librería hará el trabajo "sucio" por nosotros y ejecutará el código que sea compatible con el software del cliente que está accediendo a nuestra web. Para ello usaremos las funciones que jQuery nos proporciona, dentro de un grandísimo abanico de funcionalidades que además se extiende por medio de miles de plugins que ofrece la comunidad para implementar cualquier tipo de comportamiento.

Para lograr el **diseño web adaptativo** se ha usado la librería **jQuery Mobile** que nos proporcionara un conjunto de herramientas que simplificará el proceso de crear páginas para móviles, desde la escritura del propio código HTML, la maquetación con CSS y la creación de efectos dinámicos con Javascript.

#### 5.5.2.- *Programación Multihilo*

Para poder tener acceso simultáneo al centro domótico desde la aplicación y el acceso remoto, se ha realizado una programación multihilo, en el cual el hilo principal es la aplicación de escritorio y para el servidor se ha creado un hilo independiente que permanece a la espera de peticiones por parte de usuarios que accedan remotamente. Se ha tenido especial cuidado con la sincronización entre los hilos a la hora de acceder a las bases de datos y al puerto serie, utilizando para ellos secciones críticas para evitar problemas indeseados.

Un problema importante que me encontrado, es que la librería dao de MFC no permite acceso seguro a las bases de datos en hilos diferentes. Para solventar este problema es sumamente importante la buena aplicación de las secciones críticas en el programa.

#### 5.5.3.- Clases:

La programación que se ha seguido en la aplicación de escritorio ha sido una programación orientada a objetos, de ahí que pasemos a describir algunas de las clases más importantes que forman la aplicación, cada una de ellas con sus métodos y propiedades correspondientes:

##### **CPFC2App:**

Clase que crea la aplicación Windows.

##### **CAboutDlg:**

Clase que define el cuadro de dialogo que contiene información de la versión del programa, así como el nombre del autor.

##### **CMainFrame:**

Clase que encapsula la funcionalidad de la ventana principal.

Métodos más relevantes:

*CMainFrame*: Establece la apariencia que debe tener la aplicación al inicio de la misma.

*OnCreate*: Método donde se crea el menú de la aplicación, la barra de herramientas, la barra estado y el árbol.

*OnBarCheck*: Muestra u oculta el menú, el árbol, la barra de estado, y la barra de herramientas.

OnTheme2000: Método que cambia la apariencia de la aplicación al estilo MS Office 2000.

OnTemasWindowsxp: Cambia la apariencia de la aplicación al estilo Windows XP.

OnTheme2003: Cambia la apariencia de la aplicación al estilo MS Office 2003.

OnTheme2005: Cambia la apariencia de la aplicación al estilo MS Office 2005.

OnTemasMsoffice2007luna: Cambia la apariencia de la aplicación al estilo MS Office 2007.

OnThemeLoadSkinBin: Cambia la apariencia de la aplicación utilizando para ello el skin cargado.

(Para una mayor información de los métodos y de la librería que permite cambiar la apariencia de la aplicación consultar la ayuda de la librería disponible en el CD-ROM)

OnClose: Minimiza la aplicación a la bandeja del sistema.

OnSalir: Cierra la aplicación.

MostrarNotifylcon: Muestra u oculta el icono en la bandeja del sistema.

OnTrayNotify: Método que captura los mensajes procedentes de la bandeja del sistema.

OnRestaurar: Maximiza la aplicación desde la bandeja del sistema.

OnPosicion: Carga las posiciones relativas del menú, la barra de herramientas, y el árbol.

OnAyudaDePfc: Abre la ayuda de la aplicación.

### **CPFC2View:**

La Vista es la ventana donde se muestra los planos y las fotos de las habitaciones.

Métodos más relevantes:

IniciarArbol: Inicializa el árbol con los valores leídos de la base de datos.

CargarMapa: Carga un mapa en memoria.

Vermapa: Método que visualiza los planos y las fotos en la vista.

InsertarBoton: Función que crea un botón en tiempo de ejecución.

EliminarBotones: Elimina los botones creados en tiempo de ejecución.

NotifyDisplayTooltip: Función que se encarga de mostrar los tooltips en la vista.

OnSetCursor: Función que cambia el puntero del ratón.

LanzarDlgConsulta: Abre la hoja de propiedades del dispositivo.

OnHerramientasFecha: Abre el dialogo que permite cambiar la hora del sistema domótico.

OnHerramientasAgenda: Abre la hoja de propiedades que permite al usuario manipular la agenda y grabar la memoria del sistema domótico.

CrearPaginasWeb: Crea de manera dinamica las páginas web que utilizara el acceso remoto según la información que haya en la base de datos

GuardarPagina: Guarda en el disco duro las páginas web creadas

**CButtonST:**

Clase que define los botones utilizados en la vista.

**AgendaDlg:**

Hoja de propiedades que nos permite acceder a las páginas que manipulan y graban la agenda del sistema domótico.

Métodos:

OnInitDialog: Método donde se crean las distintas páginas de propiedades.

OnBnClickedAtras: Carga la página anterior.

OnBnClickedSiguiente: Carga la página siguiente.

OnSelchangeList: Carga una página.

**CPagUnaVez:**

Página de la agenda que permite manipular las tareas que son ejecutadas una sola vez.

Metodos:

OnInitDialog: Método que lee de la base de datos "Agenda.mdb" las tareas que se ejecutan una sola vez y las carga en el control lista.

OnNMRclickList1: Abre el menú que permite borrar o ejecutar tareas.

OnEliminar: Elimina las tareas seleccionadas.

OnEjecutar: Ejecuta las tareas seleccionadas.

OnLvnKeydownList1: Borra las tareas seleccionadas al pulsar el botón suprimir.

### **CPagSemanal:**

Página de la agenda que permite manipular las tareas que se repiten semanalmente.

Metodos:

OnInitDialog: Método que lee de la base de datos "Agenda.mdb" las tareas que se ejecutan semanalmente y las carga en el control lista.

OnNMRclickList1: Abre el menú que permite borrar o ejecutar tareas.

OnEliminar: Elimina las tareas seleccionadas.

OnEjecutar: Ejecuta las tareas seleccionadas.

OnLvnKeydownList1: Borra las tareas seleccionadas al pulsar el botón suprimir.

### **CPagDiaria:**

Página de la agenda que permite manipular las tareas que se repiten diariamente.

Metodos:

OnInitDialog: Método que lee de la base de datos "Agenda.mdb" las tareas que se ejecutan diariamente y las carga en el control lista.

OnNMRclickList1: Abre el menú que permite borrar o ejecutar tareas.

OnEliminar: Elimina las tareas seleccionadas.

OnEjecutar: Ejecuta las tareas seleccionadas.

OnLvnKeydownList1: Borra las tareas seleccionadas al pulsar el botón suprimir.

### **CPagMemoria:**

Clase que permite grabar la memoria del sistema domótico.

Metodos:

Mensual: Graba la agenda mensual en el sistema domótico.

SemanaInR: Graba la agenda semanal no repetitiva en el sistema domótico.

SemanaR: Graba la agenda semanal repetitiva en el sistema domótico.

DiariaR: Graba la agenda diaria repetitiva en el sistema domótico.

OnBnClickedButton2: Borra la memoria del sistema domótico.

### **CHoraDlg:**

Dialogo que permite cambiar la hora del sistema domótico.

OnInitDialog: Lee la hora del sistema domótico.

OnBnClickedSincronizar: Sincroniza la hora del sistema con el Pc.

### **CDlgDispositivo:**

Hoja de propiedades que nos permite acceder a las páginas que controlaran y monitorizaran el sistema domótico, además de permitirnos programar las tareas que queramos para ese dispositivo.

Metodos:

OnInitDialog: Método donde se crean las distintas páginas de propiedades.

OnBnClickedAtras: Carga la página anterior.

OnBnClickedSiguiente: Carga la página siguiente.

OnSelchangeList: Carga una página.

### **CDlgConsulta:**

Página, de la hoja de propiedades del dispositivo, que permite monitorizar y controlar el sistema domótico.

Métodos:

CrearControles: Crea en tiempo de ejecución los diferentes controles que tendrá la página. El determinado tipo de control que se creara para un determinado estado dependerá del valor leído en el campo correspondiente al estado, en la base de datos "BaseD.mdb". Para más información dirigirse al paso 5 del manual de instalación.

EliminarCtrl: Elimina los controles creados dinámicamente.

GenerarLista: Inicializa todas las listas desplegadas presente en la página.

### **CPagOpciones:**

Página donde se podrá elegir los estados para los que se programaran las tareas y el modo de ejecución de estas.

Métodos:

RadioOp: Indica el modo de ejecución elegido.

IsCheckEstado: Indica si el estado ha sido seleccionado.



IsAnyCheck: Indica si hay algún estado seleccionado.

### **CPagFecha:**

Página donde se introducirá la fecha y hora de ejecución de las tareas. Esta página será diferente dependiendo de la opción de ejecución que hayamos elegido en la página de opciones.

Métodos:

OnInitDialog: Método que inicializa y crea los controles donde se introducirá la fecha y la hora de ejecución de las tareas.

OnShowWindow: Método que determina que controles deben mostrarse según la opción elegida en la página de opciones.

EliminarCtrl: Elimina los controles creados dinámicamente.

OnBnClickedAgregar: Agrega la tarea a la agenda.

### **CPuertoSerie:**

Clase encargada de las comunicaciones por el puerto serie.

Métodos:

Envia: Envía un Byte por el puerto serie.

Recibe: Lee un Byte del puerto serie.

### **CArbol:**

Clase donde se define el árbol utilizado.

Métodos:

InsertarItem: Inserta un ítem en el árbol.

OnTvSelchanged: Indica que un ítem ha sido seleccionado, y le indica a la vista que plano o foto debe mostrar.

OnNMDblclk: Método que abre la hoja de propiedades del dispositivo que esta seleccionado. Este método se ejecutara cuando se realice un doble clic sobre el árbol.

OnKeyUp: Método que abre la hoja de propiedades del dispositivo que esta seleccionado. Este método se ejecutara al pulsar la tecla "enter" cuando el dispositivo este seleccionado.

### **CLista:**

Clase que define las listas desplegables utilizadas en la aplicación.

AddList: Añade un elemento a la lista.

SetCur: Selecciona un determinado elemento de la lista.

GetCur: Devuelve el nombre y valor del elemento seleccionado de lista.

### **CPPToolTip:**

Clase que define los tooltips utilizados en la vista.

### **CCriticalSectionBD:**

Clase que permite a los hilos el acceso a la bases de datos de manera sincronizada

### **CHiloWebServer:**

Clase que permite la creación de un hilo donde se ejecutara el servidor web que proporcionara el acceso remoto.

Metodos:

AgregarTarea: Permite guardar tareas en la agenda

Eliminar: Permite borrar las tareas seleccionadas de la agenda

Mensual: Graba la agenda mensual en el sistema domótico.

SemanalNR: Graba la agenda semanal no repetitiva en el sistema domótico.

SemanalR: Graba la agenda semanal repetitiva en el sistema domótico.

DiariaR: Graba la agenda diaria repetitiva en el sistema domótico.

Borrar: Borra la memoria del sistema domótico.

event\_handler: Método que procesa las peticiones de los usuarios

InitInstance : Inicializa el servidor web

InicializarVariables: Inicializa las variables utilizadas para la ejecución del servidor web

### **WebServer:**

Conjunto de métodos que permiten la creación de un servidor web

Programación del acceso remoto

El desarrollo en javascript se basa en capturar los eventos que se generan en la interacción del usuario con la página web. Por todo ello, debido a los numerosos eventos que se capturan, y en diferentes páginas, es mejor remitirse al código fuente para poder comprender mejor su funcionamiento.

El programa utiliza para la creación de la página web, plantillas que el instalador puede modificar para cambiar el aspecto y funcionalidades. Se ha optado por esta opción para no tener que modificar el código de la aplicación de escritorio cada vez que se quiera cambiar algo en el acceso remoto.

## 6.- PRESUPUESTO

### MANO DE OBRA

Mano de Obra	Horas	Coste (€/hora)	Total
Graduado en ingeniería telemática	300	100 €	30000 €

Tabla 6.1

### MATERIALES

Materiales	Subtotal
Visual Studio 2005	679 €
Ordenador Personal (Hardware)	900 €
Encuadernación	90 €
Materiales Consumibles	60 €

Tabla 6.2

Costes Total de los materiales: 1729 €

Total Presupuesto.....31729 €  
 16 % (IVA).....6663.09 €

**TOTAL: 38392,09 €**

El presupuesto del proyecto “Interfaz gráfica de usuario para monitorización y control de un sistema domótico” asciende a la cantidad de **treinta y ocho mil trescientos noventa y dos euros con nueve céntimos**.

Linares, a 2 de septiembre de 2014

Fdo. Miguel Ángel Fernández Ruiz.  
 Graduado en Ingeniería telemática

## 7.- CONCLUSIONES

Entre los distintos tipos de lenguajes de programación de los que disponíamos: Java, Builder C++, etc., la elección de Visual C++, se debe no solo a sus características como el uso del lenguaje C++, el cual está orientado a objetos, sino que además, contiene un gran número de componentes que lo hace potente y flexible, al lograrse usar en distintos tipos de aplicaciones. Su fácil utilización es interesante desde el punto de vista del programador. Por todo esto y con la aplicación resultante, vemos acertada la elección de Visual C++ como lenguaje de programación.

Desde un punto de vista práctico, el empleo de la librería especializada en interfaces gráficas "Prof-UIS", ha permitido el desarrollo de una interfaz gráfica vistosa e intuitiva que da mucho juego a la hora de cambiar la apariencia de la aplicación.

Además de cumplir con los objetivos solicitados respecto al control y monitorización del sistema domótico, se ha implementado un acceso remoto que tiene las mismas funcionalidades que la aplicación de escritorio y que nos permite controlar y monitorizar el centro domótico.

Visto todo lo anterior podemos comprobar que se han cumplido satisfactoriamente todos los objetivos tanto académicos como personales establecidos al principio y no sólo eso, sino que además se han incorporado una serie de mejoras que complementan la aplicación, aumentando sus prestaciones.

Así pues, tenemos una aplicación con un entorno rico, amigable y económico en comparación con realizaciones basadas en interfaces hardware (pantallas murales), más habituales, costosas y limitadas tanto en las funciones ofrecidas al usuario como en su expansibilidad y facilidad para su actualización.

En cuanto a los inconvenientes, cabe destacar que cuando se graba una agenda mensual, dicha agenda será de 28 días en vez de 31, ya que la estructura que sigue la agenda del sistema domótico hace que si se utilizara la de 31 días y se procediera a grabar la agenda el día 30, solo ejecutaría un día, ya que la agenda se invalidaría el día uno, con la pérdida de eficiencia que eso conlleva. Este inconveniente ha sido subsanado grabando en la agenda 28 días y no direccionando el puntero del día uno a la dirección '3F' de la agenda.

En cuanto a la asignación de un color determinado a las partes del plano que se desea que sean sensibles tanto al movimiento como al clic del ratón, se optó por esta opción en lugar de utilizar coordenadas, ya que un plano puede tener habitaciones que no sean rectangulares lo que llevaría a la pérdida de precisión al no poder distinguir de forma correcta las diferentes partes del plano.

Por último, reseñar que este proyecto se complementa con otro proyecto denominado "Terminales domóticos basados en pic" presentado como proyecto fin de carrera por Víctor Espinosa Gutiérrez en diciembre de 2005, en la EPS de Linares.

## **8.- LÍNEAS DE FUTURO**

Esta aplicación está diseñada bajo una estructura modular, que nos permitirá incorporar nuevas funcionalidades sin afectar por ello al funcionamiento normal de nuestro programa.

Algunas funcionalidades interesantes a incorporar en un futuro podrían ser aquellas que permitan la creación y ejecución de plantillas de tareas, que permitan la ejecución de un grupo de tareas dependiendo del periodo estacional, eventos, situaciones, etc...

## 9.- BIBLIOGRAFÍA

[1] Fco. Javier Ceballos, "Microsoft Visual C++, Aplicaciones para Win32", RA-MA, 1999

[2] Stephen Gilbert, Bill McCarty, "Programación en Visual C++ 6", Paraninfo, 1999

[3] Kate Gregory, "Microsoft Visual C++ 6", Prentice Hall, 1999

[4] Introducción al entorno de desarrollo Visual C++,

<http://www.dcp.com.ar/>

[5] Cursos de C++,

<http://www.conclase.net>

[6] Cursos, tutoriales y código fuente,

<http://www.codeproject.com/>

[7] Cursos, tutoriales y código fuente,

<http://www.codeguru.com/>

[8] Librerías graficas

<http://www.prof-uis.com/>

[9] Tutorial HTML5, CSS3, Javascript y JQuery

<http://www.w3schools.com/>

[10] Tutorial JavaScript

<http://librosweb.es/javascript/>

[11] Tutorial AJAX

<http://librosweb.es/ajax/>

[12] Tutorial JQUERY

<http://librojquery.com/>

[13] Página oficial JQUERY

<http://jquery.com/>

[14] Tutorial Jquery Mobile

<http://www.w3schools.com/jquerymobile/>

[15] Tutorial Jquery Mobile

<http://jquerymobile.com/>

[16] Recursos Web Server

<https://code.google.com/p/mongoose/>