

# UNIVERSIDAD POLITÉCNICA DE VALENCIA



*PROYECTO FINAL DE CARRERA*

## **Control domótico**

**TITULACIÓN: I.T. Informática de Gestión**

**Realizado por**

Jose Antonio Sanchis Hernández

**Dirigido por**

Dr. Lenin G. Lemus Zúñiga



# Índice

Introducción .....	7
Objetivo .....	7
Capítulo 1. Definiciones básicas .....	9
1.1. Domótica .....	9
1.2. Aportaciones de la domótica .....	10
1.2.1 Gestión energética .....	10
1.2.2 Accesibilidad .....	10
1.2.4 Seguridad .....	10
1.2.5 Confort .....	10
1.2.6 Centralización de la comunicación .....	11
Capítulo 2. Estándares utilizados en la domótica.....	13
2.1. X10 .....	13
2.2. KNX/EIB .....	14
2.2.1 Especificaciones .....	15
Capítulo 3. Selección del estándar para realizar la aplicación Web.....	17
Capítulo 4. Diseño de la BBDD .....	19
4.1. Tablas .....	19
4.1.1. Tabla Edificio .....	19
4.1.2. Tabla Planta .....	19
4.1.3. Tabla Estancia .....	20
4.1.4. Tabla Aparato .....	20
4.1.5. Tabla grupo . .....	20
4.1.5. Tabla grupo_aparato .....	20
4.2. Diagrama Entidad-Relación .....	21
Capítulo 5. Estructura e implementación de la aplicación Web .....	22
5.1. Arquitectura .....	22
5.1.1. Capa Presentación .....	23
5.1.2. Capa lógica .....	25
5.1.3. Capa persistencia .....	28
Capítulo 6. Programación de los componentes .....	29
6.1. Componentes del circuito.....	29
6.1.1. Actuador .....	29
6.1.2. Fuente de alimentación .....	30
6.1.3. Detector de movimiento.....	30
6.1.4. Pulsador .....	30
6.1.5. Sensor de luminosidad. ....	31
6.1.6. Acoplador de bus .....	31
6.1.7. Interface .....	31
6.1.8. Enchufe estándar .....	32
6.1.9. Interruptores diferenciales.....	32

6.1.9.1. Interruptor diferencial CFI6, tipo CA .....	32
6.1.9.2. Interruptor diferencial DG62 C1 .....	33
6.2. Programación de los aparatos .....	33
6.2.1. Preparación de <i>ETS3</i> .....	34
6.2.2. Configuración del edificio .....	34
6.2.3. Inserción de aparatos .....	35
6.2.4. Configurando las direcciones de grupo .....	36
Capítulo 7. Configuración y puesta en marcha del servidor.....	39
7.1 Servidor Web .....	39
7.2. eibcontrol.....	39
7.2.1. Instalación.....	39
7.2.2. Configuración.....	40
7.2.3. Encendido.....	41
7.2.4. Funcionamiento .....	41
Conclusiones.....	43
Trabajo Futuro .....	43
Referencias Bibliográficas.....	45

# Índice de Figuras

Figura 1. Estructura de la base de datos de la aplicación .....	21
Figura 2. Arquitectura de 3 capas de nuestro sistema.....	23
Figura 3. Actuador A6F16H Serie eibSOLO.....	29
Figura 4. Fuente alimentación NT160-4.....	30
Figura 5. Detector de movimiento (5WG1 255-2A12) .....	30
Figura 6. Pulsador (5WG1 243-2AB11).....	30
Figura 7. Sensor de luminosidad (5WG1 254-4AB01) .....	31
Figura 8. Acoplador de bus (5WG1 110-2AB03).....	31
Figura 9. Interface (5WG1 148-1AB04) .....	31
Figura 10. Enchufe Estándar.....	32
Figura 11. Interruptor diferencial CF16.....	32
Figura 12. Interruptor diferencial DG62.....	33
Figura 13. Detalle creación edificio en ets3 .....	34
Figura 14. Estructura del edificio en ets3.....	34
Figura 15. Direcciones de grupo en ets3.....	37
Figura 16. Ventana programación en ets3.....	37

# Índice de listados de código

Listado 1. Formulario alta de un nuevo edificio.....	24
Listado 2. Fragmento código, redirección a la página de operación .....	24
Listado 3. Código operationNuevoEdificio.php pasando datos a la capa lógica.....	25
Listado 4. Fragmento formulario de formNuevaEstancia.php, solicitando datos al sistema .....	25
Listado 5. Fragmento de código de sistema.php .....	25
Listado 6. Función obtenerEstadoAparato de la clase sistema.....	27
Listado 7. Fragmento de formConmutarGrupo.php .....	27
Listado 8. Función conmutar de la clase sistema.....	27
Listado 9. Fragmento de eibdb.xml.....	41



# Introducción

En la actualidad la domótica ofrece a hogares y a empresas una mejora en la calidad de vida y una mejora en la eficiencia del consumo energético. La domótica permite dar respuesta a los requerimientos que plantean estos cambios sociales y las nuevas tendencias de nuestra forma de vida, facilitando el diseño de casas y hogares más humanos, más personales, polifuncionales y flexibles.

Este proyecto pretende crear un sitio web para el control de instalaciones domóticas basadas en el estándar EIB. Para el desarrollo del mismo, se ha hecho uso de un circuito que simula una instalación EIB real prestado por el departamento de ingeniería de sistemas y computadores (**DISCA**) de la Universidad Politécnica de Valencia (**UPV**).

# Objetivo

Diseñar una aplicación Web para realizar el control de dispositivos EIB desde una interfaz amigable para el usuario.





# Capítulo 1

## Definiciones básicas

Para iniciar con la exposición del proyecto vamos a introducir el concepto de domótica y las ventajas y oportunidades que esta ofrece a sus usuarios e instaladores.

### 1.1. Domótica <sup>1</sup>

La domótica es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, además de aportar seguridad, confort, y comunicación entre el usuario y el sistema.

El funcionamiento básico de un sistema domótico se basa en la capacidad de recoger información proveniente de unos sensores o entradas, procesarla y emitir órdenes a unos actuadores o salidas. El sistema puede acceder a redes exteriores de comunicación o información.

La domótica aplicada a edificios no destinados a vivienda, es decir oficinas, hoteles, centros comerciales, de formación, hospitales y terciario, se denomina, inmótica.

El sector de la domótica ha evolucionado considerablemente en los últimos años, y en la actualidad ofrece una oferta más consolidada. Hoy en día, la domótica aporta soluciones dirigidas a todo tipo de viviendas, incluidas las construcciones de vivienda oficial protegida. Además, se ofrecen más funcionalidades por menos dinero, más variedad de producto, y gracias a la evolución tecnológica, son más fáciles de usar y de instalar. En definitiva, la oferta es mejor y de mayor calidad, y su utilización es ahora más intuitiva y perfectamente manejable por cualquier usuario. Paralelamente, los instaladores de domótica han incrementado su nivel de formación y los modelos de implantación se han perfeccionado. Asimismo, los servicios posventa garantizan el perfecto mantenimiento de todos los sistemas.

En definitiva, la domótica de hoy contribuye a aumentar la calidad de vida, hace más versátil la distribución de la casa, cambia las condiciones ambientales creando diferentes escenas predefinidas, y consigue que la vivienda sea más funcional al permitir desarrollar facetas domésticas, profesionales, y de ocio bajo un mismo techo.

La red de control del sistema domótico se integra con la red de energía eléctrica y se coordina con el resto de redes con las que tenga relación: telefonía, televisión, y tecnologías de la información, cumpliendo con las reglas de instalación aplicables a cada una de ellas. Las distintas redes coexisten en la instalación de una vivienda o edificio. La instalación interior eléctrica y la red de control del sistema

---

<sup>1</sup> <http://www.cedom.es/que-es-domotica.php>

domótico están reguladas por el Reglamento Electrotécnico para Baja Tensión (**REBT**). En particular, la red de control del sistema domótico está regulada por la instrucción **ITC-BT-51** Instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad para viviendas y edificios.

## **1.2. Aportaciones de la domótica**<sup>2 3</sup>

La domótica contribuye a mejorar la calidad de vida del usuario. Esta mejora la podemos clasificar en cinco factores de los cuales los usuarios se pueden beneficiar:

### **1.2.1 Gestión energética**

Los sistemas domóticos permiten gestionar inteligentemente la iluminación, climatización, agua caliente, sistemas de riego, electrodomésticos, etc., aprovechando mejor los recursos naturales, utilizando las tarifas horarias de menor coste, y reduciendo por tanto de esta manera la factura energética. Además, mediante la monitorización de consumos, se obtiene la información necesaria para modificar los hábitos y aumentar el ahorro y la eficiencia.

### **1.2.2 Accesibilidad**

Personas con algún tipo de discapacidad tienen una gran ayuda en el manejo de los elementos del hogar, permitiendo adaptarlos a sus necesidades y ofreciéndoles un grado de autonomía. Por ejemplo, gestos tan sencillos como el control de la iluminación pueden ser controlados por sensores de presencia o incluso de voz. Los destinatarios de estas tecnologías son todas las personas, ya que por enfermedad o envejecimiento, todos somos o seremos discapacitados, más pronto o más tarde.

### **1.2.4 Seguridad**

Gracias al control que ofrece la domótica podemos mejorar la seguridad de personas y bienes de nuestro hogar, gracias al uso de controles de intrusión, alarmas, cierres centralizado de puertas y persianas, técnicas que permiten detectar incendios, fugas de gas o inundaciones de agua, etc. Además todos estos aspectos se pueden controlar de una forma centralizada.

### **1.2.5 Confort**

Si lo deseamos tenemos la posibilidad de automatizar los distintos equipos dotándolos de un control eficiente y de fácil manejo, haciendo de nuestro hogar un sitio más confortable. Tenemos la posibilidad de multi-programar los aparatos a nuestras necesidades, cambiando su comportamiento dependiendo del momento del día o del uso que le queramos dar.

---

<sup>2</sup> <http://es.wikipedia.org/wiki/Domotica>

<sup>3</sup> <http://www.domotica-maiher.com/ventajas/>

### **1.2.6 Centralización de la comunicación**

“Hablar con la casa” ahora es posible, gracias a los sistemas domóticos obtenemos la posibilidad de tener en casa un único sistema de comunicaciones con el cual comunicar nuestras órdenes directamente a nuestro hogar, además de poder enlazarlo vía internet para facilitar esta comunicación. Por ejemplo podríamos utilizar nuestro televisor, móvil o pc para controlar todo el edificio, obteniendo en vivo el estado de la casa, avisos de anomalías y el estado de funcionamiento de nuestros equipos.



# Capítulo 2

## Estándares utilizados en la domótica <sup>4</sup>

La necesidad de tener dispositivos compatibles entre sí en el mercado domótico y la inexistencia de un estándar internacional, ha fomentado el desarrollo de diferentes estándares que se han ido implantando en el mercado a lo largo del tiempo. En la actualidad existe un gran número de protocolos de control de dispositivos domóticos, algunos de ellos propietarios. Esto ha dificultado enormemente la labor de integración y la mezcla de diferentes marcas en una misma instalación, así como la adquisición de experiencia por parte de instaladores, ingenieros, usuarios, etc. En nuestros días, se están imponiendo las soluciones flexibles y abiertas.

Aunque actualmente se está trabajando para desarrollar un estándar común, que facilite la competencia entre fabricantes y la posibilidad de elección de los clientes, esto aún está relativamente lejano y los productos que actualmente están disponibles para la automatización de edificios siguen determinados estándares, por lo que la elección inicial de un estándar, marcará la posterior evolución del sistema, ya que todos los dispositivos de las ampliaciones futuras deberán funcionar y ser compatibles a dicho estándar.

Entre toda la gran variedad de estándares existentes en domótica e inmótica, vamos a considerar tan sólo dos de los más implantados en Europa, ya que fueron los ofrecidos inicialmente para el desarrollo del proyecto, y sobre los que se ha tenido que elegir para la implementación del mismo.

A continuación, mostraremos resumidamente las características principales de cada uno de dichos estándares y concluiremos mostrándolos en una tabla comparativa.

### 2.1. X10 <sup>5 6</sup>

**X10** fue la primera tecnología domótica en aparecer. Fue desarrollado en 1975 por **Pico Electronics of Glenrothes**, Escocia, para permitir el control remoto de los dispositivos domésticos. Es de código abierto y el más difundido de todos los estándares domóticos.

La filosofía fundamental de diseño de **X-10** es la de que los productos puedan interoperar entre ellos, y la compatibilidad con los productos anteriores de la misma gama. El sistema **X-10** ha sido desarrollado para ser flexible, fácil de usar y fácil de instalar por el mismo usuario sin necesidad de conocimientos especiales. Es un sistema escalable, es decir se puede empezar utilizando un producto en particular, por ejemplo

---

<sup>4</sup> <http://www.buscaclima.es/descargas/domotica/standares%20de%20domotica.pdf>

<sup>5</sup> <http://es.wikipedia.org/wiki/X10>

<sup>6</sup> <http://www.domotica.es/x10>

el control de una luz, y expandir luego el sistema para incluir la seguridad o el control con el ordenador con componentes fáciles de instalar y que no requieren cableados especiales. Esta sencillez a su vez se convierte en una dificultad para crear un auténtico proyecto domótico enlazando todos los productos, ya que los productos están diseñados principalmente para su uso individual.

El protocolo **X10**, utiliza la línea eléctrica (220V o 110V) para transmitir señales de control entre equipos de automatización del hogar en formato digital y por lo tanto no existe la necesidad de instalar un nuevo cableado, suponiendo un ahorro importante para el cliente. Como desventaja encontramos que al usar la red eléctrica este sistema se convierte en poco fiable frente a ruidos eléctricos.

Las señales de control de **X10** se basan en la transmisión de ráfagas de pulsos de **RF** (120 kHz) que representan información digital. A su vez, cada orden se transmite 2 veces, con lo cual toda la información transmitida tiene cuádruple redundancia. Cada orden involucra 11 ciclos de red (220 ms para 50 Hz y 183,33, para 60Hz).

Hoy en día, **X-10** es un estándar y a la vez un fabricante de estos mismos productos y productos compatibles con **X-10** (alarmas, televisiones, contestadores, interfaces de ordenador, etc.).

Podemos concluir que **X10** es una opción recomendable en proyectos en los que la vivienda ya está construida y por lo tanto una instalación cableada tendría un coste excesivo y también sería útil utilizar este tipo de tecnología para proyectos sencillos.

## 2.2. KNX/EIB <sup>7</sup>

**KNX** es la iniciativa de tres asociaciones europeas (**EIBA**, **BCI** y **EHSA**) con el objetivo de aunar los esfuerzos de todos los fabricantes de sistemas domóticos del mercado europeo, para que con el apoyo de los gobiernos y del resto de industrias europeas, pueda crearse un único estándar europeo para la automatización de viviendas y oficinas. Es decir el estándar **KNX** es el paso evolutivo lógico que trata de concentrar toda la experiencia y conocimientos de los principales estándares europeos en un único estándar común, abierto y con dispositivos a precios suficientemente competitivos, como para que los propietarios de viviendas empiecen a cuestionarse la instalación de un sistema de este tipo.



El protocolo **KNX** se basa en su mayoría en el protocolo **EIB**. **EIB** fue promovido por la **EIBA** con el objetivo de crear un estándar europeo, con el suficiente número de fabricantes, instaladores y usuarios, que permita la comunicación de todos los dispositivos de una instalación eléctrica. La **EIBA** es una asociación de más de 115 empresas europeas líderes en el mercado eléctrico (Todos sus miembros cubren más del 80% de la demanda de equipamiento eléctrico en Europa) que se unieron en 1990 con el fin de impulsar la creación de un estándar domótico europeo.

---

<sup>7</sup> [http://www.ingenieria.eu/html/pagina.php?par\\_idioma=es&par\\_opcion=2&va\\_tipo=30](http://www.ingenieria.eu/html/pagina.php?par_idioma=es&par_opcion=2&va_tipo=30)

**EIB**, es un protocolo de red abierto, que se fortalece del aporte de todas y cada una de la infinidad de empresas que lo suscriben. Son tantas que la cantidad y la diversidad de dispositivos existentes supera la de cualquier otro sistema en la actualidad. Según datos de la **EIBA**, a principios del año 2003, había unos 10 millones de dispositivos **EIB** instalados por todo el mundo, unas 70.000 instalaciones, una gama de 4.500 productos diferentes, y más de 70.000 instaladores cualificados.

### 2.2.1 Especificaciones

El funcionamiento del **EIB** es el mismo que el del **KNX**: En un principio solo se contemplo como medio físico un único cable de dos hilos conductores o bus de cable, a través del cual se comunican entre sí los participantes del bus. Este sistema sobre par trenzado funciona a 9.6 Kbps. La alimentación de 24 Vdc se suministra a los componentes a través de los propios hilos conductores de este.

Cada dispositivo **EIB** tiene dos direcciones de 16 bits, la física y la lógica. Ambas se asignan al preparar la instalación pero tienen funciones diferentes, y son mutuamente excluyentes, (es decir se usa solo una de las dos). En realidad la dirección física solo se utiliza mientras se da de alta la instalación o se configuran los dispositivos de la instalación. Su misión es identificar cada uno de los dispositivos, diferenciándolos de los demás. Está definida en función de la posición del dispositivo de en la matriz de conexión. De esta forma cada dispositivo que conectemos tendrá su dirección física compuesta de: zona (4 bits) + línea (4 bits) + dispositivo (8 bits); en total 64.000 posibilidades de direccionamiento. Por otro lado, la dirección lógica o de grupo de 16 bits, es con la que el dispositivo trabajará realmente en la instalación mientras esta operando de forma normal, y no tiene por qué ser única (varios dispositivos pueden compartir una misma dirección lógica). Por lo tanto sirve para definir grupos de dispositivos desde un punto de vista funcional (del tipo de luces, interruptores, sensores de alarma, etc...).

La topología o forma en la que se extiende el bus del cable al que se conectan los distintos dispositivos, pueden tener un aspecto que podríamos denotar por matricial. En primer lugar se forman líneas, cada una de las cuales puede tener conectados hasta 255 dispositivos, utilizando cada uno de ellos un dispositivo denominado acoplador al bus. Cada línea puede tener una longitud máxima de 1.000 metros en total (sumando todas las ramas y respetando ciertas limitaciones en cuanto a la distribución de los dispositivos respecto a la fuente). Después, en un extremo de la línea se puede conectar un aparato denominado acoplador de línea, que hace las veces de puente y que, en este caso, permite colgar ésta y otras líneas hasta un máximo de 16, de una principal para formar una matriz denominada zona. Siguiendo la misma estructura que antes, las zonas también se pueden conectar entre sí, hasta un máximo de 16. Para ello, cada zona se conecta a un cable general, mediante un dispositivo denominado acoplador a la red troncal. Finalmente el sistema completo podrá conectarse a otros sistemas iguales o diferentes, mediante la pasarela correspondiente.

La característica fundamental del **EIB** es su organización descentralizada, es decir todos los componentes del sistema pueden comunicarse entre sí sin necesidad de una unidad centra de control.

También se pretendió que el **EIB** funcionase sobre otros medios físicos, aunque en la práctica sólo el par trenzado ha conseguido una implantación masiva:

- **EIB.PL** Corrientes portadoras sobre 230 Vac y 50 Hz a 1200/2400 bps.
- **EIB.net** Ethernet a 10 Mbps.
- **EIB.RF** Radiofrecuencia.
- **EIB.IR** Infrarrojo.
- **EIB.bluetooth**.

Por último, es importante destacar que **EIB** contempla una serie de herramientas software basadas en el PC. Por ejemplo el **ETS3** es un programa sobre plataforma Windows que facilita el desarrollo de proyectos de ingeniería EIB, como el diseño y la configuración, con independencia del fabricante de los productos.



# Capítulo 3

## Selección del estándar para realizar la aplicación Web<sup>8</sup>

Tras estudiar las dos posibilidades ofrecidas para el desarrollo del proyecto, se ha elegido el estándar de **KNX/EIB**, por considerarse un firme candidato a convertirse a largo plazo en el único estándar a nivel Europeo. Las razones que se quieren destacar por las cuales se considera esto son las siguientes:

- Se trata de un estándar europeo abierto, por lo que no existe una dependencia de un fabricante en concreto. Esto implica una competencia de mercado con la consecuencia de una progresión hacia precios cada vez más competitivos y una calidad cada vez más alta.
- La cantidad y la diversidad de dispositivos existentes supera la de cualquier otro sistema en la actualidad. Esta amplia variedad de dispositivos disponibles cubrirán adecuadamente las necesidades de funcionalidad de cualquier tipo de proyecto domótico.
- En las instalaciones tradicionales cada función requiere una línea eléctrica propia, y cada sistema de control precisa una red separada. Por el contrario, con el **EIB** se pueden controlar, comunicar y vigilar todas las funciones de servicio y su desarrollo, con una única línea común. Con esto se puede dirigir la línea de energía sin desvíos, directamente hasta el aparato consumidor.
- Además del ahorro en el cableado se presentan adicionalmente otras ventajas: La instalación en un edificio se puede realizar de un modo más sencillo desde el principio, y después se puede ampliar y modificar sin problemas. Ante cambios de uso o reorganización del espacio, el **EIB** consigue una adaptación rápida y sin problemas, mediante una fácil ordenación (cambio de parametrización) de los componentes del bus, sin necesidad de un nuevo cableado.
- El **EIB** se puede conectar mediante las correspondientes interfaces con los centros de control de otros sistemas de automatización de edificios o con una red digital de servicios integrados (**RDSI**). De este modo el uso del **EIB** en una vivienda unifamiliar resulta tan rentable como en hoteles, escuelas, bancos, oficinas o edificios del sector terciario.

Por todas estas razones, se ha querido aprovechar la oportunidad de realizar este proyecto como una apuesta de futuro aprendiendo de un estándar que está en auge liderando los mercados y esperemos que cada vez generando más puestos de trabajo.

---

<sup>8</sup> <http://www.mundomotica.es/web1/knx1.htm>



# Capítulo 4

## Diseño de la BBDD

Durante la realización del proyecto, llegado al punto en el que nuestro circuito esta correctamente programado y se puede controlar a través del programa eibcontrol como veremos más adelante, surge un nuevo problema. En los enlaces usados por eibcontrol debe estar especificada la dirección de grupo de cada uno de los aparatos que queremos controlar, esto supone saber con antelación cuales son las direcciones de cada aparato, limitando el diseño de la web a este circuito en concreto y no pudiendo ser usado en otros proyectos con direcciones y aparatos distintos.

Para solucionar esta barrera en el proyecto se ha incluido el uso de una sencilla base de datos que permite que nuestra web sea una web dinámica y adaptable a diferentes proyectos domóticos. Con ella se permite que sea el instalador o el propio usuario, el que configure su portal, dando de alta los aparatos de su instalación que quiera controlar vía **HTTP**. Es decir, esta opción permite que la web pueda ser usada en cualquier circuito **EIB** eliminando la barrera que nos limitaba al circuito usado en el proyecto usando enlaces estáticos. Con la creación de una base de datos modificable por el propio usuario, los enlaces **html** que eran necesarios para el control de eibcontrol se pueden generar de forma dinámica con la información de incluida en la base de datos.

Para el diseño de la base de datos se ha hecho uso de **MYSQL** por ser un software de uso libre, además de combinarse perfectamente con nuestro servidor **Apache**. Por otra parte, ya se tenían conocimientos de su uso lo cual ha facilitado el trabajo.

### 4.1. Tablas

Dentro de la base de datos ha surgido la necesidad de incluir un total de 6 tablas: edificio, planta, estancia, aparato, grupo y grupo\_aparato. A continuación describimos la finalidad con la que se ha creado cada una.

#### 4.1.1. Tabla Edificio

En el caso de encontrarnos en un proyecto domótico grande, pueden existir varios edificios en nuestro hogar o empresa. A sí que se ha querido separar correctamente, en caso necesario, las instalaciones por edificios para facilitar así su localización en la web. Del edificio tan solo guardaremos un nombre, descripción y una clave que los identificara que se genera automáticamente.

#### 4.1.2. Tabla Planta

Nuestro edificios, como en una situación real están compuestos con una o más plantas. La creación de estas tiene la misma finalidad que los edificios, facilitar el

usuario encontrar lo que necesita. Cada una de estas plantas vendrá identificada con una clave única que se genera automáticamente llamada "id". En ella se escribirá el nombre que desee el usuario y una descripción si fuera necesaria.

#### **4.1.3. Tabla Estancia**

Dentro de cada uno de nuestros edificios, dividiremos las diferentes zonas de nuestro hogar en estancias, estas podrán ser de tipo dormitorio, cocina, salón, baño, entrada, terraza, recibidor, porche o garaje. Cada una de estas estancias vendrá identificada con una clave única que se genera automáticamente llamada "id". En ella se escribirá el nombre que desee el usuario y una descripción si fuera necesaria.

#### **4.1.4. Tabla Aparato**

Dentro de cada una de las estancias de nuestro hogar encontraremos los aparatos (luces, calefacción, vigilancia...). En esta tabla identificaremos con una clave automática los aparatos dentro de cada estancia. Para darlos de alta el usuario o instalador, como información relevante, tan sólo deberá conocer la dirección de grupo de cada aparato para su correcta instalación. Esta dirección de grupo será la que use la aplicación web para consultar el estado del mismo o para conmutarlo. Como información adicional, se deberá de insertar un nombre y descripción del aparato en cuestión. Además se puede, si se conoce, añadir la dirección física del aparato para posibles ampliaciones del proyecto en las que se pudiera programar el aparato.

#### **4.1.5. Tabla grupo**

En un primer momento esta tabla no estaba incluida en el proyecto ya que se consideraba oportuno que cada aparato almacenara la dirección del grupo al que pertenece. En el caso de que participara en más de un grupo, se debía de dar de alta nuevamente el aparato tantas veces como direcciones de grupo tuviera, indicando su uso en la descripción del mismo. El problema surge a la hora de mostrar el estado de la casa, en el que un mismo aparato aparecería en diversas ocasiones en los listados (tantas como direcciones de grupo) y peor todavía, podía tener estados distintos (encendido/apagado) dependiendo del estado de la dirección de grupo. Debido a que se buscaba una solución más profesional, se incluyó esta tabla en la que se registran una única vez las direcciones de grupo que hay en nuestro sistema, y mediante un código complejo en la capa lógica de nuestra aplicación web obtener adecuadamente el estado del grupo y el estado de cada aparato independientemente del estado de todas sus direcciones.

#### **4.1.5. Tabla grupo\_aparato**

Ya que la cardinalidad entre la tabla grupo y la tabla aparato es de muchos a muchos (un aparato TIENE muchas direcciones de grupo y una dirección de grupo TIENE muchos aparatos), se ha necesitado incluir esta tabla para que se cumpla esta. En ella relacionamos un grupo con un aparato.

## 4.2. Diagrama Entidad-Relación

A continuación, observamos el diagrama entidad-relación que forma nuestra base de datos. Las cardinalidades de las relaciones son todas 1:N:

Un edificio tiene n plantas.

Una planta tiene n estancias.

Una estancia tiene n aparatos.

Un aparato tiene n grupos (grupo\_aparato)

Un grupo tiene n aparatos (grupo\_aparato)

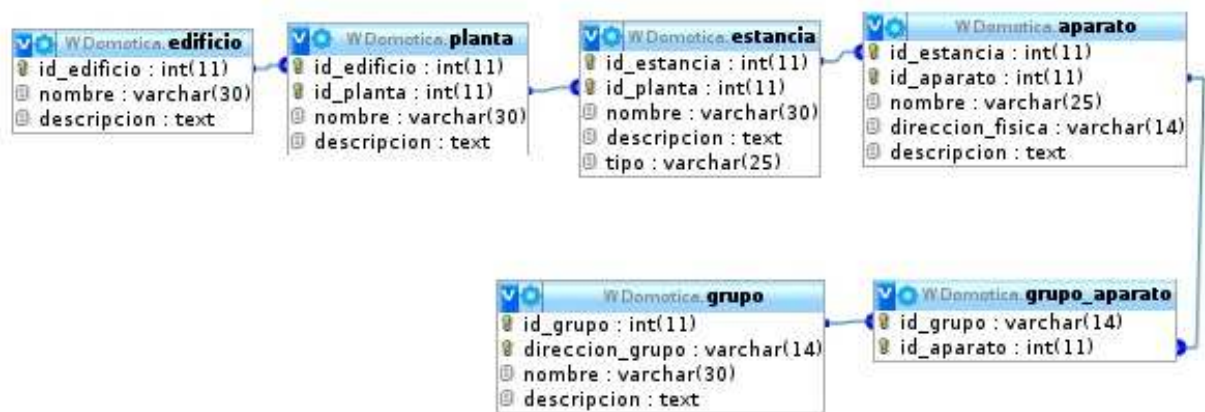


Figura 1. Estructura de la base de datos de la aplicación

# Capítulo 5

## Estructura e implementación de la aplicación Web

En el siguiente capítulo se va a describir que estructura sigue nuestra aplicación web y a continuación describiremos los detalles de su implementación e instrucciones de funcionamiento.

El objetivo a alcanzar con la aplicación es crear una página que permita ver el estado de los aparatos de nuestro hogar/empresa y permitir su encendido/apagado. Además como se ha comentado en el capítulo anterior se ha dotado de una base de datos con la que es posible que este proyecto sea usado en diferentes instalaciones EIB. Para ello, como veremos más adelante, tan solo se deberán de dar de alta en la web los distintos aparatos y las direcciones de grupo.

Nuestro sitio web está escrito en *html* con *php* embebido y operar sobre una base de datos *MySQL*.

### 5.1. Arquitectura

La arquitectura elegida para el desarrollo del proyecto ha sido la división en tres capas (Presentación, Lógica y Persistencia). Se ha decidido implementar la división en capas puras (cada capa se comunica únicamente con la capa subyacente) y cerradas (cada capa se comunica con la capa inferior a través de una clase interfaz de capa), para mejorar la modificabilidad del sistema y la independencia inter-capa y de datos, mejorando así la calidad del SW desarrollado.



```

        <TD>Datos obligatorios</TD>
    </TR>
    <TR valign="top">
        <TD>Nombre:</TD>
        <TD><input class="obligatorio" type="text" name="nombre" title="Nombre que aparecerá para
identificar el edificio"></TD>
    </TR>
    <TR>
    <TR>
        <TD>Descripción:</TD>
        <TD><input class="obligatorio" type="text" name="descripcion" title="Descripción del
edificio"></TD>
    </TR>

    <TR>
    <TD></TD>
        <TD align="right"><INPUT type="submit" VALUE="Crear" title="Enviar los datos para dar de alta un nuevo
edificio" /></TD>
    </TR>
</TABLE>
</form>

```

Listado 1. Formulario alta de un nuevo edificio

Como podemos ver en la segunda línea, al pulsar sobre el botón Crear, los datos se enviarán a `index` y este se encargará, tras comprobar la variable “contenido”, de redirigir hacia la página **php** correspondiente como vemos en el siguiente fragmento de `contenidoCentral.php` que forma parte de `index.php`.

```

switch($contenido){
    case " operationNuevoEdificio ":
        $nombreContenidoCentral= "";
        $contenidoZonaInformacion= " operationNuevoEdificio.php";
        break;

```

Listado 2. Fragmento código, redirección a la página de operación

Vemos ahora como ahora `operationNuevoEdificio.php` pasa los datos a nuestra capa de lógica.

```

<?
require_once "preambulo.php";

if(!empty($_POST["nombre"])){
    $resultado=$Sistema->nuevoEdificio($_POST["nombre"], $_POST["descripcion"]);
    if($resultado!=0){
        echo "Operación realizada con éxito.";
    }
    else
        echo "Error, vuelva a intentarlo";

```



```
}  
?>
```

Listado 3. Código operationNuevoEdificio.php pasando datos a la capa lógica.

Existen fragmentos y páginas de la capa de lógica en las que se muestra información obtenida de la base de datos al usuario. En este caso se hace uso de igual modo de funciones de la capa lógica, pero estas ahora nos retornan un array de datos con los que trabajar. Vemos aquí un fragmento de código en las que se solicita las plantas existentes en nuestra instalación para que el usuario pueda seleccionar una en un desplegable.

```
<?php  
    $plantas=$Sistema->obtenerPlantas();  
    if(!empty($plantas)){  
        $nPlantas=count($plantas);  
        for($i=0;$i<$nPlantas;$i++){  
            $planta= $plantas[$i];  
            $idPlanta=$planta["id_planta"];  
            $nombrePlanta=$planta["nombre"];  
            echo '<OPTION value="'. $idPlanta. "'>'. $nombrePlanta.'</OPTION>';  
        }  
    }  
?>
```

Listado 4. Fragmento formulario de formNuevaEstancia.php, solicitando datos al sistema

### 5.1.2. Capa lógica

La capa lógica de nuestro sitio web está escrita en su totalidad en el archivo *sistema.php*. Dentro encontramos la clase sistema, la cual incluye todas las funciones que necesitamos para recuperación e inserción de datos en la base de datos. Podemos ver a continuación un fragmento de código de la clase con o un método de recuperación de datos y otro de inserción de datos en la base de datos.

```
class Sistema{  
    private $persistencia;  
  
    function __construct(){  
        $this->persistencia=new comunicacion_persistencia();  
    }  
  
    function nuevaEstancia($idPlanta,$nombre,$descripcion,$tipo){  
        return $this->persistencia->addEstancia($idPlanta, $nombre, $descripcion, $tipo);  
    }  
  
    function obtenerEstancias(){  
        return $this->persistencia->obtenerEstancias();  
    }  
}
```

Listado 5. Fragmento de código de sistema.php

Como podemos ver en el código, la clase tiene una instancia de la clase persistencia, para mandar los datos a una función de comunicación\_persistencia.php que se encarga de su inserción o recuperación en la base de datos.

Para poder obtener el estado de un aparato (encendido/apagado) se ha tenido que escribir una función algo más compleja. Esto es debido a que un aparato puede estar presente en diferentes direcciones de grupo que lo conmuten y cada una de estas direcciones puede estar en un estado diferente, dejando dependiendo de orden en que se conmuten el aparato en un estado diferente. Tras analizar la problemática se concluyó que únicamente se tenía que tener en cuenta la dirección de grupo que operó última sobre el aparato, pues esta es la que imperará sobre su estado. Ya que en nuestra base de datos no podíamos guardar cual es la última en ejecutarse, debido a que ocurren cambios ajenos a nuestra web por ejemplo cuando se pulsa un interruptor, debíamos de consultar cual era preguntándole a nuestro servidor **EIB eibcontrol**. **eibControl** registra en una variable de su archivo **xml** el estado de una dirección de grupo y cuando esta fue modificada en una variable llamada **Timestamp**.

En el código pues debíamos de seguir los siguientes pasos:

1. Obtener todas las direcciones de grupo en la que aparece el aparato.
2. Por cada dirección obtener su *timestamp* (momento de última modificación).
3. Comprobar que dirección de grupo fue conmutada mas tarde.
4. Leemos el estado de la dirección de grupo que se ha ejecutado más tarde.

Este es el código resultante de la función:

```
function obtenerEstadoAparato($idAparato){
// 1. Obtener todas las direcciones de grupo en la que aparece el aparato
    $grupos=$this->persistencia->obtenerDireccionesAparato($idAparato);
// 2. Por cada dirección obtener su timestamp (momento de ultima modificación)
    $nDirecciones=count($grupos);
    $tiempos[0]=0;
    for($i=0;$i<$nDirecciones;$i++){
        $grupo=$grupos[$i];
        $dirGrupo=$grupo["direccion_grupo"];
        $url='http://localhost:8080/eibhomeserver?instruction=<eibtype="read"
path="/eib/groups/'.$dirGrupo.'/timestamp"/>';
        $file = @fopen($url, 'r');
        $contenido="";
        if($file){
            while(!feof($file)) {
                $contenido .= @fgets($file, 4096);
            }
            fclose ($file);
            $pos = strpos($contenido,"data=");
            $tiempos[$i] = substr($contenido,$pos+6,10);
        }
    }
// 3. Comprobar que dirección de grupo fue conmutada mas tarde (está es la que predomina sobre el
    estado del aparato)
    $nTiempos=count($tiempos);
    $mayor=-1;
```

```

$DirGrupo='-1'; // Posición del grupo que ha actuado mas tarde
for($i=0;$i<$nTiempos;$i++){
    if($tiempos[$i]>$mayor){
        $mayor=$tiempos[$i];
        $DirGrupo=$i;
    }
}
$estado='-1';
if($DirGrupo!=-1){
    // 4. Leemos el estado de la dir de grupo que se ha ejecutado mas tarde
    $grupo=$grupos[$DirGrupo];
    $dirGrupo=$grupo["direccion_grupo"];
    $estado=$this->obtenerEstadoGrupo($dirGrupo);
}
return $estado;
}

```

Listado 6. Función obtenerEstadoAparato de la clase sistema

En la página *formConmutarGrupo.php* se permite al usuario poder cambiar el estado de las direcciones de grupo existentes en el sistema entre encendido y apagado. Para que esto sea posible debemos de ejecutar la orden correspondiente de ***eibcontrol*** en el navegador. Para ello creamos los enlaces dependiendo del estado en el que se encuentre actualmente el navegador como vemos en el siguiente fragmento de código.

```

if($estado==1){
    echo "Encendido->";
    $url='operationConmutar.php?dirGrupo='.$DirGrupo.'&operacion=0';
    echo "<a href='$url'>Apagar</a>";
}
else{
    echo "Apagado->";
    $url='operationConmutar.php?dirGrupo='.$DirGrupo.'&operacion=1';
    echo "<a href='$url'>Encender</a>";
}

```

Listado 7. Fragmento de *formConmutarGrupo.php*

Los enlaces anteriores mandan los datos hacia *operationConmutarGrupo.php*, el cual ejecuta la función en *sistema.php* que permitirá la conmutación. Vemos aquí la función conmutar de la clase sistema, que forma la url necesaria a partir de la dirección de grupo y el tipo de operación que se quiera realizar (0 apagar, 1 encender).

```

function conmutar($dirGrupo,$operacion){
    $url='http://localhost:8080/eibhomeserver?instruction=<eib                                type="write"
path="/eib/groups/'.$dirGrupo.'/curvalue" data="'.$operacion.'"/>';
    $contenido = ";
    $file = @fopen($url, 'r');
    fclose ($file);
}

```

Listado 8. Función conmutar de la clase sistema

### **5.1.3. Capa persistencia**

En la capa persistencia encontramos dos archivos: accesoBD\_MySQL.php y comunicaci3n\_persistencia.php. En la clase AccesoBD encontramos todo lo necesario para realizar la conexi3n con MySQL y realizar las consultas, modificaciones e inserciones en la base de datos. La capa de l3gica se comunica con la clase comunicacion\_persistencia, la cual ejecuta las operaciones b3sicas de lectura, modificaci3n o inserci3n sobre la clase AccesoBD.

# Capítulo 6

## Programación de los componentes

Para comenzar describiremos los componentes sobre los que se ha trabajado y la permitido la realización de pruebas, los tipos utilizados han determinado la amplitud del proyecto ya que al no disponer de más tipos de receptores y actuadores el proyecto no es todo lo generalizado como se desearía.

### 6.1. Componentes del circuito<sup>9</sup>

#### 6.1.1. Actuador

El actuador de 6 canales de la Serie **eibSOLO** está especialmente pensado para la conmutación de cargas C con capacitancia o con corrientes de conexión (carga inductiva).

La potencia de ruptura de cada contacto asciende a 16A en 250V AC, de manera que cada canal puede estar alimentado por su propia fase / magnetotérmico.

Cada contacto se lleva a un terminal doble sin tornillos de conexión rápida. Gracias a ello, se hace posible el empalme de la fase, de acuerdo con las normativas, así como al introducción de dos cables en el terminal de salida de la forma más sencilla.

La posibilidad de manejo manual de cada canal sirve también como indicador del estado de la conexión.



Figura 3. Actuador A6F16H Serie eibSOLO

<sup>9</sup> <http://www.hidrair.com/articulo.asp?f=PS3&s=INSTABUS%20EIB>

### 6.1.2. Fuente de alimentación

Las fuentes de alimentación generan la alimentación necesaria para la línea **KNX/EIB**. Llevan integrada una bobina de Bus.

La **NT 640-4** de **LINGG & JANKE** posee, además, una salida 29 V DC no filtrada para alimentar otra línea en combinación con una bobina adicional (conexión mediante terminal).

El número de componentes Bus a los que puede alimentar dependerá de la potencia de los mismos. Es importante calcular esto en la fase de diseño del proyecto.

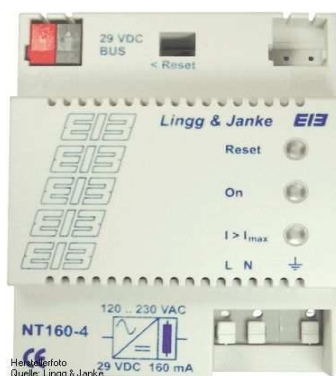


Figura 4. Fuente alimentación NT160-4

### 6.1.3. Detector de movimiento

Detector de movimiento UP 255 Delta perfil blanco titán. Interruptor de proximidad para detección de personas. El detector infrarrojo integrado reacciona ante cambios de temperatura dentro de su área de vigilancia. Ángulo de detección horizontal: 180°. Montaje a 2,20 m. Alcance frontal: 12 m aprox. Alcance lateral: 6 m aprox. Sensor de luminosidad integrado. Tipo 5WG1 255-2AB12 de **Siemens**.



Figura 5. Detector de movimiento (5WG1 255-2AB12)

### 6.1.4. Pulsador

Pulsador doble UP 243, Delta Profil blanco titán. Consta de dos teclas con posibilidad de pulsación en la parte superior y en la parte inferior. Dispone de dos LED's para indicación de estado y un LED de orientación. Montaje sobre el acoplador al bus UP 110. Tipo 5WG1 243-2AB11 de **Siemens**.



Figura 6. Pulsador (5WG1 243-2AB11)

### 6.1.5. Sensor de luminosidad.

Sensor de luminosidad GE 254 para iluminación indirecta. Se compone de un convertidor y un receptor (célula fotoeléctrica) con un conductor de conexión de 2 m. de longitud (no puede ser prolongado). El convertidor recibe del receptor el valor actual de luminosidad y regula o controla la iluminación a través del instabus EIB. Rango de ajuste: de 150 a 1.950 Lux. Montaje dentro de los equipos o en falsos techos. Tipo 5WG1 254-4AB01 de **Siemens**.



Figura 7. Sensor de luminosidad (5WG1 254-4AB01)

### 6.1.6. Acoplador de bus

Acoplador de bus UP 110. Aparato de empotrar para montaje en cajas de mecanismos. Permite la conexión a la línea de bus de los siguientes aparatos finales de bus: pulsadores, detectores de movimiento, reguladores de temperatura, etc. Tipo 5WG1 110-2AB03 de Siemens.



Figura 8. Acoplador de bus (5WG1 110-2AB03)

### 6.1.7. Interface

Interface RS 232, N 148/04 para protocolos estándar y FT1.2. Permite la conexión al puerto serie COM1 o COM2 de un ordenador personal para el direccionamiento, parametrización, visualización, protocolo y diagnóstico de los componentes bus. Mediante el interface RS 232, N 148/04 es posible manejar todos los componentes de una red EIB completa. Montaje en carril DIN. Dimensiones: 3 módulos. Tipo 5WG1 148-1AB04 de **Siemens**.



Figura 9. Interface (5WG1 148-1AB04)

### 6.1.8. Enchufe estándar

El enchufe hembra, tomacorriente o toma de corriente generalmente se sitúa en la pared, ya sea colocado de forma superficial (enchufe de superficie) o empotrado en la pared montado en una caja (enchufe de cajillo o tomacorriente empotrado), siendo éste el más común. Constan, como mínimo, de dos piezas metálicas que reciben a sus homólogas macho para permitir la circulación de la corriente eléctrica. Estas piezas metálicas quedan fijadas a la red eléctrica por tornillos o, actualmente con mayor frecuencia, por medio de unas pletinas plásticas que, al ser empujadas, permiten la entrada del hilo conductor y al dejar de ejercer presión sobre ellas, unas chapas apresan el hilo, impidiendo su salida.



Figura 10. Enchufe Estándar

### 6.1.9. Interruptores diferenciales

La instalación cuenta con dos diferenciales.

#### 6.1.9.1. Interruptor diferencial CF16, tipo CA<sup>10</sup>

Un interruptor diferencial exponencial, también llamado disyuntor por corriente diferencial o residual, es un dispositivo electromecánico que se coloca en las instalaciones eléctricas con el fin de proteger a las personas de las derivaciones causadas por faltas de aislamiento entre los conductores activos y tierra o masa de los aparatos.



Figura 11. Interruptor diferencial CF16

Especificaciones:

Tipo de disyuntor	diferencial
Corriente de choque	250 A
Capacidad de cortocircuito	6 kA
Característica	AC
Desconexión	instantáneo
Durabilidad mecánica	4000 ciclos
Montaje	DIN

<sup>10</sup> [http://es.wikipedia.org/wiki/Interruptor\\_diferencial](http://es.wikipedia.org/wiki/Interruptor_diferencial)



Sección de conductos de conexión	1,5...35 mm <sup>2</sup>
Grueso de carriles de conexión	0,8...2 mm
Clase de impermeabilidad	IP40

### 6.1.9.2. Interruptor diferencial DG62 C1 <sup>11</sup>

Interruptor diferencial en caja moldeada serie M2



Figura 12. Interruptor diferencial DG62

Especificaciones:

Intensidad nominal I <sub>n</sub>	6-40A
Tensión nominal en c.a. U <sub>n</sub>	230/400V
Curva de disparo	C
Capacidad de cortocircuito según EN/IEC 60898-1	6kA
Clase de selectividad	3
Capacidad de terminales	25 ó 2 x 10 mm <sup>2</sup>
Polos	1,1 + N; 2
Peso	115 g/pol

## 6.2. Programación de los aparatos

Antes de poder usar un producto basado sobre el protocolo **EIB**, este debe ser programado. La programación, básicamente se basa en ajustar algunos parámetros de configuración sobre el uso que le vamos a dar al aparato (normalmente los que traen por defecto son validos) y en asignar una dirección o varias de grupo a un aparato, estas direcciones serán las que serán usadas para enviar/recibir los telegramas necesarios para la comunicación.

La programación de los componentes del circuito se ha realizado gracias a una versión de evaluación de **eib tool software 3(ETS3)**. **ETS** facilita la configuración de instalaciones domóticas, en esta memoria se van a describir los pasos generales que se han seguido sin entrar en detalles específicos del programa, ya que no es el propósito del proyecto. Para más información sobre el uso general del programa podemos usar el manual de **ETS3**.

<sup>11</sup>

[http://www.gepowercontrols.com/es/resources/literature\\_library/catalogs/downloads/Instal\\_Cat\\_ES\\_final\\_LR.pdf](http://www.gepowercontrols.com/es/resources/literature_library/catalogs/downloads/Instal_Cat_ES_final_LR.pdf)

### 6.2.1. Preparación de ETS3

Tras la instalación de **ETS3**, el primer paso que hemos tenido que realizar es añadir las bases de datos de los fabricantes de los aparatos que vamos a usar. Estas bases de datos son aportadas por los propios fabricantes y se pueden bajar desde su propia web. Urls de descarga:

- Base de datos para EIB de Siemens: [https://www.swe.siemens.com/spain/web/es/industry/automatizacion/Formacion/sce/Documents/PDB\\_J\\_ES.VD2](https://www.swe.siemens.com/spain/web/es/industry/automatizacion/Formacion/sce/Documents/PDB_J_ES.VD2)
- Base de datos para EIB de Lingg-Janke : [http://www.lingg-janke.de/uploads/tx\\_sbdowloader/LJ0506\\_01.vd2](http://www.lingg-janke.de/uploads/tx_sbdowloader/LJ0506_01.vd2)

### 6.2.2. Configuración del edificio

Una vez actualizadas nuestras bases de datos comenzaremos a crear un nuevo proyecto **ETS3** con la estructura de nuestra casa “ficticia” (ya que nosotros tenemos todos los componentes juntos). En nuestro caso hemos configurado un edificio de una planta y una única habitación en la que se encuentran todos nuestros componentes descritos anteriormente.

Hemos llamado a nuestro proyecto domoticaW. A continuación en la ventana de Edificios vamos a dar de alta un nuevo edificio. Vamos a darle como nombre “EdificioPrincipal”.

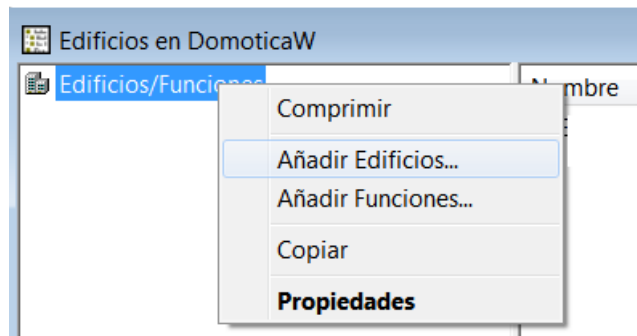


Figura 13. Detalle creación edificio en ets3

Ahora dentro del edificio daremos de alta una parte del edificio y la llamaremos “PlantaPrincipal” y dentro de esta una habitación que llamaremos HabitaciónPrincipal. Como resultado debe de quedar la estructura de la imagen.

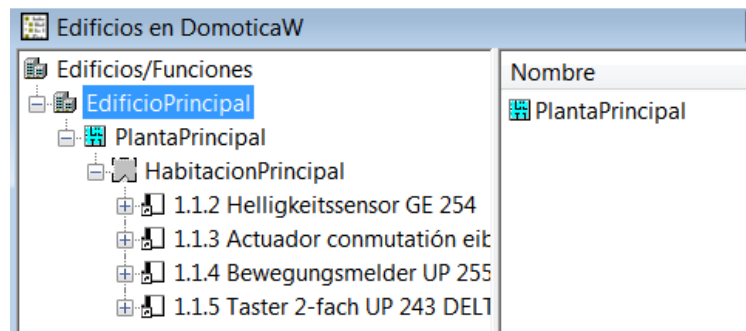


Figura 14. Estructura del edificio en ets3

### 6.2.3. Inserción de aparatos

Una vez creada la estructura de nuestro edificio, vamos a añadir los aparatos que componen nuestra instalación, para ello pulsamos sobre la habitación con el botón derecho y en el menú contextual seleccionamos Añadir Aparatos.

Debemos de seleccionar el fabricante sobre el que queremos hacer la búsqueda y escribir su número de pedido. A veces su búsqueda no es trivial y debemos de escribir parte del numero de pedido y buscarlo manualmente, por ejemplo el pulsador costo de encontrar bastante porque se escribía su número de pedido completo (5WG1 243-2AB11) y en la base de datos está guardado como 5WG1 243-2AB\_1 debido a que la programación y configuración es igual para los modelos 5WG1 243-2AB11 y 5WG1 243-2AB21. Después de encontrarlo debemos insertarlo con un doble clic, debemos tener en cuenta que el orden de inserción determinara la dirección física de cada aparato. Aunque esta dirección puede ser modificada a posteriori.

Llegados a este punto en el que se ha citado la dirección física es conveniente que aclaremos que en ets3 existen dos tipos de direcciones: las físicas y las lógicas<sup>12</sup>:

- **Las direcciones físicas**, son asignadas automáticamente a cada aparato en el momento de su inserción en el proyecto, y serán usadas por ets3 para identificar a cada aparato para su comunicación con el programa, una vez que termines las labores de programación y configuración de los aparatos podemos olvidarnos de estas direcciones pues no volverán a hacernos falta. Dicha dirección está definida conforme a la posición del dispositivo en la matriz de conexión y cada componente se identifica por su zona, línea y número de componente, de manera que la estructura topológica es la siguiente:

Zona + Línea + Dispositivo  
(4 bits) (4 bits) (8 bits)

- **Las direcciones lógicas**, llamadas en **ETS** direcciones de grupo. Estas direcciones son con la que trabajará realmente la instalación mientras está en funcionamiento y no tiene por qué ser única, es decir, durante la realización del proyecto se pueden determinar hasta catorce direcciones de grupo, en función de las diferentes partes de la instalación, de forma que varios dispositivos pueden compartir una misma dirección lógica. Esto sirve para definir grupos de dispositivos desde el punto de vista de la función técnica que van a desempeñar (del tipo "luces del piso de arriba", "interruptores", "sensores de alarma", etc.).

Cada grupo principal contiene, según el criterio del usuario, hasta 2.048 subgrupos. Como se ha dicho antes, las direcciones de grupo de los componentes se ordenan independientemente de las direcciones físicas. De este modo, cada componente puede comunicarse con cualquier otro. Así pues, para el funcionamiento práctico del sistema se requiere una dirección, llamada dirección de grupo, destinada al tráfico de telegramas. En cada telegrama se incluye la dirección de grupo del receptor, de manera que los componentes del bus leen esta dirección, lo que determina si deben aceptar o no la información. Los componentes del bus pueden responder a más de una dirección de grupo. La

---

<sup>12</sup> <http://www.ingecasa.com/Sistema%20EIB.pdf>

recepción del telegrama será efectiva sólo si la dirección de grupo coincide con alguna de las que tiene el componente y, además, la transmisión es correcta.

Una vez introducidos todos los aparatos, pasaremos a conectar el circuito a nuestro PC mediante el puerto COM para la programación de la dirección física de los mismos. Debemos aclarar en este momento que **ETS** funciona sobre protocolo estándar no sobre **ft1.2**, así que habrá que conmutar nuestra interfaz a protocolo estándar. Ahora bastará con pulsar una a uno con el botón derecho y pulsar en programar, en la nueva ventana seleccionaremos programar dirección física.

#### **6.2.4. Configurando las direcciones de grupo**

Pasamos ahora a configurar las direcciones de grupo, que serán las que nos permitirán dar la funcionalidad que necesitamos a nuestra instalación, por ejemplo si en una misma dirección de grupo añadimos la función de conmutar de la tecla izquierda de nuestro pulsador y añadimos la entrada A de conmutación de nuestro actuador, al pulsar el pulsador izquierdo, será la entrada A y solo la A la que sea conmutada. Digamos que hemos sustituido el cable físico que iba desde el interruptor al enchufe en las instalaciones estándar por un enlace lógico que se puede cambiar a nuestro antojo durante la programación de los aparato.

Hemos creado una serie de direcciones de grupo en las que interviene los mismos aparatos en combinaciones diferentes para dar una mayor cantidad de ejemplos y acercarnos más a una instalación real en la que una luz por ejemplo se pueda encender por medio de un pulsador y además de conmutarse por el encendido de un sensor de presencia.

En nuestro caso hemos añadido dos grupos principales en la ventana de direcciones de grupo, uno para las funciones en las que los sensores (presencia y luz) y nuestro pulsador conmutaran los enchufes enlazados a nuestro actuador llamado grupo "Funciones" y otro para la conmutación individual del actuador sobre los enchufes llamado "enchufes".

Dentro de este grupo se han creado 1 grupo intermedio llamado actuador, debido a que las funciones que añadamos operaran sobre este. Dentro de este grupo intermedio hemos añadido 4 direcciones de grupo, una por cada una de los enchufes que están conectados al conmutador.

La primera dirección de grupo conmutara la entrada A del actuador cuando sea pulsada la tecla izquierda de nuestro pulsador, la hemos llamado por ello "pulsador izquierdo". Para que la programación de la dirección de grupo funcione correctamente deberemos arrastrar a la ventana de la dirección de grupo las operaciones de apagado y encendido de la tecla izquierda de nuestro pulsador y seguidamente la entrada A del conmutador. Podemos encontrar ambos en la ventana de Edificios o de topología.

Las demás direcciones se configuran siguiendo la misma idea, uniendo las funciones de los aparatos sobre entradas de nuestro actuador.

Objeto	Aparato	Envío	C	R	W	T	U	Producto	Programa
0: Conectar tecla izquierda - Enc...	1.1.5 Pulsador doble UP 2...	S	C	-	W	T	U	Pulsador doble UP 243 DE...	12 52 On-off-Conmut/Reg/Pers...
1: Conectar tecla izquierda - Ap...	1.1.5 Pulsador doble UP 2...	S	C	-	W	T	U	Pulsador doble UP 243 DE...	12 52 On-off-Conmut/Reg/Pers...
0: Entrada A Conmutar - Canal ...	1.1.3 Actuador conmutati...	S	C	-	W	-	-	Actuador conmutación eib...	6x conmutar, tiempo, fun. lóg.,...

Figura 15. Direcciones de grupo en ets3

El segundo grupo principal funcionara de un modo un poco distinto, ya que no existirá un actuador que opere sobre nuestro actuador. En la dirección de grupo, tan solo añadiremos las entradas del actuador que queremos que queden afectadas. El motivo es porque estas direcciones serán conmutadas desde nuestro servidor web, por tanto no es necesario que haya ningún pulsador ni similar que conmute la entrada.

Una vez creadas todas las direcciones de grupo tan solo nos queda programar los aparatos. Para ello, en la ventana de topología o edificios, debemos pulsar uno a uno sobre los aparatos con el botón derecho y seleccionar programar, y en la siguiente ventana pulsamos sobre Programar Aplicación.



Figura 16. Ventana programación en ets3

Ahora ya está listo para funcionar, y podemos realizar las primeras pruebas.



# Capítulo 7

## Configuración y puesta en marcha del servidor

Para poder disponer de una web que acceda a nuestra instalación **EIB**, necesitamos de un pc-servidor que esté conectado a esta. Pasamos ahora describir los pasos a seguir para poner en marcha nuestro servidor sobre una maquina Linux.

### 7.1 Servidor Web

Este pc deberá ser configurado como servidor para que pueda atender las peticiones de los clientes. Dado que nuestro proyecto necesita de un servidor que soporte **php** y **mysql** se ha pensado como solución instalar **XAMPP**. Este software permite instalar de forma fácil un servidor apache que contiene todo lo que necesitamos.

Descargamos pues el paquete de instalación disponible en: <http://www.apachefriends.org/es/xampp.html>. Una vez descargado, descomprimos el paquete de instalación y lo movemos a la carpeta /opt y listo. Ahora para la puesta en marcha del servidor tan solo debemos de iniciar xampp con esta instrucción (con permisos de administrador):

```
/opt/lampp/./lampp start
```

Si queremos entrar a nuestra interfaz para la configuración de algunos servicios de **xampp**, lo podemos hacer a través del navegador con la siguiente url: <http://localhost/>. Si lo que queremos es tener acceso a nuestro servidor de bases de datos **Mysql** para la modificación e inserción de tablas, debemos hacerlo mediante la siguiente url: <http://localhost/phpmyadmin/index.php>

### 7.2. eibcontrol

Pasamos ahora a comentar los pasos a seguir para instalar, configurar y poner en marcha el software.

#### 7.2.1. Instalación

Lo primero que deberemos hacer es descargarnos el paquete desde <http://sourceforge.net/projects/eibcontrol/> concretamente el enlace de la versión utilizada en nuestro proyecto es este <http://sourceforge.net/projects/eibcontrol/files/eibcontrol/0.9.1/eibcontrol-0.9.1.tgz/download?test=goal>. Una vez descargado, descomprimiremos el paquete en el directorio raíz.

```
cp eibcontrol-0.9.1.tgz /
```

Llegado a este momento es importante recordar que debemos tener instalado en nuestra maquina Linux **g++** para poder compilar los archivos. Para instalar **g++** bastara con ejecutar la siguiente orden:

```
apt-get install g++
```

Ahora ya estamos preparados para compilar el programa.

```
cd /eibcontrol  
make
```

Ya tenemos nuestro programa instalado, faltará ahora configurarlo para nuestra instalación concreta.

### 7.2.2. Configuración

Eibcontrol necesita saber la estructura lógica de nuestra instalación .El fichero de ejemplo **XML** *eibcontrol/homeserver/eibdb.xml* hay que adaptarlo a la instalación EIB con la que vamos a trabajar. El **XML** tiene que contener entradas para todas las direcciones de grupo que se desean controlar.

En el fichero xml se debe almacenar las diferentes direcciones de grupo, por cada dirección de grupo debemos de indicar obligatoriamente su *eistype*, *curvalue* y *timestamp*. La función de cada variable es:

- *eistype*, indicar el tipo de componente. En nuestro caso son todo de tipo 1.
- *curvalue*, esta variable es usada por el programa para almacenar el estado de la dirección de grupo (0 apagado, 1 encendido).
- *timestamp*, aquí se guarda el momento en el que sucedió el ultimo evento en esta dirección de grupo.

Vemos a continuación un fragmento del fichero usado en nuestra instalación, en el podemos observar 4 direcciones de grupo: 1/0/0, 1/0/1, 1/0/2 y 1/0/3.

```
<node name="eib">  
<node name="groups">  
<node level="1" name="1">  
<node level="2" name="0">  
<node level="3" name="0">  
<node name="eistype">1</node>  
<node name="curvalue">0</node>  
<node name="timestamp">1317233294</node>  
</node>  
<node level="3" name="1">  
<node name="eistype">1</node>  
<node name="curvalue">0</node>  
<node name="timestamp">1316772386</node>  
</node>  
<node level="3" name="2">  
<node name="eistype">1</node>  
<node name="curvalue">1</node>  
<node name="timestamp">1317235153</node>
```



```
</node>
<node level="3" name="3">
<node name="eistype">1</node>
<node name="curvalue">1</node>
<node name="timestamp">1316777872</node>
</node>
</node>
</node>
```

Listado 9. Fragmento de eibdb.xml

### 7.2.3. Encendido

Antes de poner en marcha nuestro servidor **EIB**, el pc, como hemos comentado antes, debe estar conectado con el bus **EIB** mediante un acoplador de bus compatible con el protocolo **FT1.2** (una **BCU2**), por ello en este momento debemos de conmutar nuestra interfaz a ft1.2. Una vez conectado, para su puesta en marcha tan solo deberemos de escribir en el terminal la siguiente instrucción

```
/eibcontrol/./eibcontrol start
```

### 7.2.4. Funcionamiento

Si todo ha ido bien, se pueden controlar los dispositivos registrados en el fichero XML vía EIB Server. Se pueden manejar los dispositivos desde un navegador web como Mozilla o IE. Existen dos tipos de instrucciones una para la lectura de un valor y otra para la escritura vemos aquí unos ejemplos:

- Lectura: [http://localhost:8080/eibhomeserver?instruction=<eib type="read" path="/eib/groups/2/0/1/curvalue"/>](http://localhost:8080/eibhomeserver?instruction=<eib type=);
- Escritura: [http://localhost:8080/eibhomeserver?instruction=<eib type="write" path="/eib/groups/2/0/1/curvalue" data="1"/>](http://localhost:8080/eibhomeserver?instruction=<eib type=);

En la variable *type* escribiremos el tipo de operación, en el *path* debemos de sustituir la dirección del ejemplo por la dirección de grupo que queramos leer y en el caso de querer escribir un valor lo tengo que indicar en la variable data (0 apagar, 1 encender).



# Conclusiones

Previo al comienzo de la realización del proyecto se realizó una gran tarea de investigación para el conocimiento de la domótica, sus estándares y sus usos. La sorpresa vino en la gran dificultad que se tuvo para encontrar un software libre para el control de dispositivos EIB, aún más para encontrar un software para la programación de los mismos, pues no se encontró y se debió de usar una versión de evaluación del único software que se encontró (**ETS**). Podemos concluir que la domótica es un mundo aún por explotar, en la que no existe una gran cantidad de oferta software para el control y programación de los componentes, y por tanto hay oportunidad de negocio. Esto es debido a la lenta inserción de la domótica en nuestra sociedad, producida tal vez por unos precios aún algo excesivos para el servicio que hoy en día se les da. Esta lenta evolución de la domótica, viene agravada aún más por la “crisis del ladrillo” en la que no se producen apenas nuevas viviendas. Puedo concluir por tanto que EIB es una buena apuesta, pero de futuro, en la que se reactive la economía y se comience de nuevo a invertir en nuevos inmuebles, aunque también es cierto que en el momento que esto ocurra la cantidad de oferta software domótica crecerá exponencialmente.

La realización de este proyecto ha sido una experiencia enriquecedora, con la introducción a un mundo bastante desconocido y con muchas posibilidades de futuro.

## Trabajo Futuro

El presente proyecto podría ser ampliado añadiendo nuevas funcionalidades dependiendo del tipo de dispositivo que se esté usando, por ejemplo la regulación por porcentaje de la iluminación, la cual no se ha implementado por no disponer de un dispositivo sobre el que probarlo.

Si esta aplicación tuviera como fin un uso real, debería de ampliarse con la codificación de la seguridad, que permitiera un control de que los datos introducidos por el usuario son válidos y la autenticación del mismo para evitar el uso indeseado.

A demás sería muy interesante ampliar este proyecto para que una vez configuradas las direcciones de grupo en la web, pudieran poderse programar los aparatos, sin necesidad de necesitar otro software, en este caso el **ETS3**. Esta labor necesitará un gran conocimiento de las instrucciones el estándar EIB, además de necesitar un software menos limitado como es eibcontrol.



# Referencias Bibliográficas

- [1] <http://www.cedom.es/que-es-domotica.php>
- [2] <http://es.wikipedia.org/wiki/Domotica>
- [3] <http://www.domotica-maiher.com/ventajas/>
- [4] <http://www.buscaclima.es/descargas/domotica/standares%20de%20domotica.pdf>
- [5] <http://es.wikipedia.org/wiki/X10>
- [6] <http://www.domotica.es/x10>
- [7] [http://www.ingenieria.eu/html/pagina.php?par\\_idioma=es&par\\_opcion=2&va\\_tipo=30](http://www.ingenieria.eu/html/pagina.php?par_idioma=es&par_opcion=2&va_tipo=30)
- [8] <http://www.mundomotica.es/web1/knx1.htm>
- [9] <http://www.hidrair.com/articulo.asp?f=PS3&s=INSTABUS%20EIB>
- [10] [http://es.wikipedia.org/wiki/Interruptor\\_diferencial](http://es.wikipedia.org/wiki/Interruptor_diferencial)
- [11] [http://www.gepowercontrols.com/es/resources/literature\\_library/catalogs/downloads/Instal\\_Cat\\_ES\\_final\\_LR.pdf](http://www.gepowercontrols.com/es/resources/literature_library/catalogs/downloads/Instal_Cat_ES_final_LR.pdf)
- [12] <http://www.ingecasa.com/Sistema%20EIB.pdf>