

# ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



## PROYECTO FIN DE CARRERA

### ENTORNO 3D INTERACTIVO PARA CONTROL DE REDES DOMÓTICAS

**Titulación:** Ingeniero de Telecomunicación

**Autor:** Dña. María Ferragut Fiol

**Tutores:** D. José Alberto Rabadán Borges

**D. Julio Francisco Rufo Torres**

**Fecha:** Febrero 2012



# ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



## PROYECTO FIN DE CARRERA ENTORNO 3D INTERACTIVO PARA CONTROL DE REDES DOMÓTICAS

### HOJA DE FIRMAS

**Alumno/a**

*Fdo.: María Ferragut Fiol*

**Tutor/a**

**Tutor/a**

*Fdo.: D. José Alberto Rabadán Borges Fdo.: D. Julio Francisco Rufo Torres*

*Fecha: Febrero 2012*



# ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



## PROYECTO FIN DE CARRERA

### ENTORNO 3D INTERACTIVO PARA CONTROL DE REDES DOMÓTICAS

#### HOJA DE EVALUACIÓN

Calificación: \_\_\_\_\_

**Presidente**

*Fdo.:*

**Vocal**

**Secretario/a**

*Fdo.:*

*Fdo.:*

*Fecha: Febrero 2012*



# Agradecimientos

*Señor, dame fuerzas para lo que me  
pides y pide lo que quieras  
San Agustín 354-430.  
Obispo, filósofo y Padre de la Iglesia Latina*

Mi más sincero agradecimiento a mis profesores D. José Antonio Rabadán Borges y D. Julio Francisco Rufo Torres por su apoyo y dedicación a lo largo no sólo de este proyecto sino de toda mi carrera. Por su paciencia y su ayuda prestada.

A mis padres por enseñarme la importancia de la responsabilidad, la dedicación y la coherencia en mis decisiones no sólo en el ámbito laboral sino también en el diario.

A mi hermana Carmen, por mostrarme que no importa las veces que uno caiga, ya que si eres capaz de levantarte y luchar siempre alcanzarás tus objetivos.

A mi hermana Begoña por enseñarme a valorar los pequeños detalles y a que por gritar más no se adquiere mayor razón, sino todo lo contrario.

A mi hermana Mercedes, por demostrarme con su ejemplo que la felicidad no se logra con títulos, sino sacrificándonos por hacer felices a los que nos rodean cada día.

A mis sobrinos por hacerme ver cada día desde que nacieron, que Dios siempre está a mi lado sonriéndome.

Héctor Salas Betancor y Diego Chanca Palazón, gracias por devolverme la sonrisa cada vez que la perdía, esto no habría sido posible sin vuestro apoyo y amistad.

A mis amigos y compañeros de fatiga, Juan Miguel Cañizález Díaz, Pablo Rodríguez de Armas, Elena Vitores Quintana, José Maestre Díaz y Nuria Trujillo Quijada, porque cada uno de vosotros fue un rayo de alegría tanto en la sala de estudio como celebrando cada aprobado, muchas gracias por vuestra ayuda, apoyo y alegría.

A la Universidad de Las Palmas de Gran Canaria, por facilitarme los medios y materiales necesarios para la elaboración de éste proyecto.

# Glosario

<b>AA</b>	Acopladores de Área
<b>AB/BCU</b>	Unidad de Acoplamiento al Bus
<b>ADSL</b>	Línea de abonado digital asimétrica
<b>AL</b>	Acopladores de Línea
<b>AML</b>	Amplificador repetidor de Línea
<b>API</b>	Application Programming Interface
<b>BCC</b>	Controlador de Acoplamiento al Bus
<b>BIM</b>	Bus Interface Module
<b>CAN</b>	Controller Area Network
<b>CSMA-CA</b>	Carrier Sense Multiple Access With Collision Avoidance
<b>CX</b>	Cable Coaxial
<b>EEPROM</b>	Electrically Erasable Programmable Read-Only Memory
<b>EHS</b>	European Home System
<b>EIB</b>	European Installation Bus
<b>EIBA</b>	European International Business Academy
<b>EIS</b>	Eib Interworking Standard
<b>FDM</b>	Frequency Division Multiple Access
<b>FM</b>	Fecucy Modulation
<b>FO</b>	Fibra Óptica
<b>FPS</b>	First Person Shooter
<b>HOME PNA</b>	Home Phonenumber Networking Alliance
<b>IA</b>	Inteligencia Artificial
<b>IFE</b>	Interfaz Físico Externo
<b>IR</b>	Infla rojos
<b>LONWORKS</b>	Local Operation Network
<b>MA</b>	Módulo de Aplicación
<b>MT</b>	Módulo de Transmisión
<b>OPC</b>	Ole for Process Control
<b>OpenGL</b>	Open Graphics Library
<b>OSI</b>	Open Systems Interconnection
<b>PA</b>	Programa de Aplicación
<b>PL</b>	Power Line
<b>PPP</b>	Point to Point Protocol
<b>RDSI</b>	Red Digital de Servicios Integrados
<b>REP</b>	Repetidores
<b>RF</b>	Radio Frequency

<b>ROM</b>	Read Only Memory
<b>SOA</b>	Zona de Operación Segura
<b>SOAP</b>	Simple Object Access Protocol
<b>TP</b>	Par Trenzado
<b>TPS</b>	Third Person Shooter
<b>VDSL</b>	Very High Bit-Rate Digital Subscriber Line
<b>XDSL</b>	Línea de Abonado Digital

# Índice Temático

<b>AGRADECIMIENTOS .....</b>	<b>I</b>
<b>GLOSARIO.....</b>	<b>III</b>
<b>Capítulo 1. Introducción .....</b>	<b>3</b>
1.1. Objetivos del proyecto .....	5
1.2. Organización de la memoria .....	7
<b>Capítulo 2. Sistemas domóticos .....</b>	<b>11</b>
<b>2.1. Composición de un sistema domótico .....</b>	<b>16</b>
2.1.1. Elementos de un sistema domótico .....	17
2.1.2. Arquitectura del sistema domótico.....	18
2.1.3. Medio de transmisión .....	22
2.1.4. Tecnologías existentes .....	26
<b>2.2. Estándares. Sistema KNX .....</b>	<b>27</b>
2.2.1. X10 .....	29
2.2.2. CEBus .....	31
2.2.3. HomePNA .....	33
2.2.4. LonWorks.....	34
2.2.5. BatiBus .....	37
2.2.6. EHS .....	38
2.2.7. European Installation Bus (EIB) .....	40
2.2.8. KONNEX/KNX .....	42
<b>Capítulo 3. Sistema KNX .....</b>	<b>47</b>
<b>3.1. Topología del sistema.....</b>	<b>48</b>
3.1.1. Línea y Área .....	49
3.1.2. Direccionamiento .....	51
3.1.3. Dirección física .....	51
3.1.4. Dirección de grupo .....	54
3.1.5. Técnica de transmisión.....	57
<b>3.2. Acoplador .....</b>	<b>60</b>

<b>3.3. Componentes .....</b>	<b>62</b>
<b>3.4. Comunicación.....</b>	<b>65</b>
3.4.1. Acceso al bus.....	65
3.4.2. Esquema de un telegrama.....	66
<b>Capítulo 4. Entornos 3D de visualización.....</b>	<b>75</b>
<b>4.1. Historia de los videojuegos.....</b>	<b>76</b>
<b>4.2. Géneros de los videojuegos.....</b>	<b>77</b>
4.2.1. Simulación.....	78
4.2.2. Aventura .....	80
4.2.3. Acción .....	81
<b>4.3. Nuestro entorno virtual.....</b>	<b>84</b>
<b>4.4. Motores de juego. GameStudio.....</b>	<b>86</b>
4.4.1. Conocimientos previos.....	86
4.4.1.1. La programación.....	86
4.4.1.2. La física .....	87
4.4.1.3. Las matemáticas .....	88
4.4.1.4. La inteligencia artificial (IA).....	88
4.4.1.5. La comunicación .....	89
4.4.1.6. Application Programming Interface (API).....	89
4.4.2. Game Engine .....	90
4.4.2.1. Game Engine cerrados.....	92
4.4.2.1.1UDK.....	92
4.4.2.1.2Unity .....	92
4.4.2.1.3Scaleform .....	93
4.4.2.2. Game Engine abiertos.....	94
4.4.2.2.1Torque.....	94
4.4.2.2.2Ogre 3D .....	94
4.4.2.2.3Nebula.....	94
4.4.2.3. Nuestro Game Engine: Game Studio .....	95
4.4.2.3.1Entidades.....	96
4.4.2.3.2Editor WED .....	97
4.4.2.3.3Editor SED.....	98
4.4.2.3.4Editor MED .....	100
<b>Capítulo 5. Diseño e implementación de la plataforma virtual.....</b>	<b>105</b>
<b>5.1Diseño de la vivienda. Level Editor.....</b>	<b>106</b>
5.1.1. El intefaz y la barra de menus .....	107
5.1.2. Creación del escenario .....	111
5.1.3. Texturas.....	115
5.1.4. Creación de un modelo.....	118
5.1.5. Efectos exteriores .....	119
<b>5.2. Programación de las comunicaciones externas. Script Editor .....</b>	<b>122</b>

5.2.1. Creación de la entidad .....	123
5.2.2. Programación de la acción .....	125
<b>Capítulo 6. Diseño e implementación del sistema domótico .....</b>	<b>131</b>
<b>6.1 Integración de internet en la domótica .....</b>	<b>132</b>
6.1.1. TCP/IP no está optimizado para la Domótica .....	132
<b>6.2. Pasarelas IP .....</b>	<b>134</b>
6.2.1. Combridge IPAS .....	134
6.2.2. Falcon .....	135
6.2.3. Calimero .....	135
6.2.4. Pasarelas IP comercializadas por KNX6.....	136
6.2.4.1. Central IP de Jung .....	137
6.2.4.2. NK2 de Lingg&Janke.....	137
<b>6.3. Implementación del sistema domótico .....</b>	<b>138</b>
6.3.1. Componentes.....	140
6.3.1.1. Dimmer.....	140
6.3.1.2. Shutter.....	141
6.3.1.3. Entrada/Salida binaria .....	142
6.3.1.4. EIB Solo Netzwerk-Koppler .....	142
6.3.1.5. Pulsadores.....	143
<b>6.4. Programación del sistema mediante el software ETS3 .....</b>	<b>143</b>
6.4.1. Direcciones de grupo.....	144
6.4.1.1. Iluminación.....	145
6.4.1.1.1. Dimmer ON/OFF.....	145
6.4.1.1.2. Valor Dimmer .....	146
6.4.1.1.3. Entrada Bin .....	147
6.4.1.2. Persianas .....	148
6.4.1.2.1. Persianas SUBIR/BAJAR.....	148
6.4.1.2.2. Persianas STOP/CORTO.....	149
6.4.2. Programación de los componentes.....	150
6.4.2.1. EIBSolo. Network .....	150
6.4.2.2. Dimmer.....	152
6.4.2.3. Shutter.....	154
6.4.2.4. Entrada binaria.....	156
6.4.2.5. Salida binaria .....	157
6.4.2.6. Pulsadores.....	158
<b>6.5. Diseño de la página web .....</b>	<b>160</b>
<b>6.6. Producto final .....</b>	<b>162</b>
<b>Capítulo 7. Conclusiones.....</b>	<b>177</b>
<b>Bibliografía.....</b>	<b>183</b>
<b>Pliego de Condiciones.....</b>	<b>187</b>

P.C.1 Elementos Hardware .....	187
P.C.2 Elementos Software .....	189
<b>Presupuesto .....</b>	<b>191</b>
<b>P.1. Introducción.....</b>	<b>191</b>
<b>P.2 Costes de Recursos Humanos.....</b>	<b>192</b>
<b>P.3.Coste de Recursos Materiales.....</b>	<b>194</b>
P.3.1.Costes de Recursos Hardware.....	195
P.3.2.Costes de Recursos Software .....	196
P.3.3.Material fungible .....	197
<b>P.4.Beneficio industrial .....</b>	<b>199</b>
<b>P.5.Redacción del Proyecto Fin de Carrera .....</b>	<b>199</b>
<b>P.6.Visado del COIT .....</b>	<b>200</b>
<b>P.7.Gastos de envío .....</b>	<b>201</b>
<b>P.8.Aplicación de impuestos.....</b>	<b>201</b>
<b>P.9.Coste total del Proyecto Fin de Carrera.....</b>	<b>201</b>
<b>Anexo .....</b>	<b>205</b>

# Índice de Tablas

Tabla 2.1. Ventajas e inconvenientes de los sistemas centralizado y distribuido .....	21
Tabla 2.2. Medios de transmisión y áreas de aplicación preferidas para estándar EIB.....	41
Tabla 3.1. Limitaciones de las longitudes del cableado .....	49
Tabla 3.2. Tipos EIS .....	71
Tabla 5.1. Menú desplegable del level editor .....	110
Tabla P1. Coeficiente de corrección.....	193
Tabla P.2. Costes asociados a los Recursos Humanos .....	194
Tabla P.3. Costes asociados a los Recursos Hardware.....	196
Tabla P.4. Costes asociados a los Recursos Software .....	197
Tabla P.5. Costes asociados a material fungible .....	198
Tabla P.6. Costes total del proyecto .....	202



# Índice de Figuras

Figura 1.1. Esquema del funcionamiento del sistema total .....	7
Figura 2.1. Esquema de Arquitectura de Sistema Domótica Centralizada.....	19
Figura 2.2. Esquema de Arquitectura de Sistema Domótica Descentralizada .....	20
Figura 2.3. Esquema de Arquitectura de Sistema Domótica Híbrida/Mixta.....	21
Figura 2.4. Tecnologías presentes en las redes de comunicación .....	27
Figura 2.5. Codificación X10 en la red eléctrica de 60 Hz .....	30
Figura 2.6. Red doméstica multimedia IEEE 1394 .....	32
Figura 2.7. Red doméstica HomePNA .....	34
Figura 2.8. Estructura del Neuron Chip y Dispositivo Lonworks de comunicación.....	36
Figura 2.9. Capas del modelo OSI.....	39
Figura 2.10. Ejemplo básico de red EIB.....	42
Figura 3.1. Configuración de un área .....	50
Figura 3.2. Interconexión de áreas.....	51
Figura 3.3. Ejemplo de asignación de direcciones .....	52
Figura 3.4. Ejemplo de direccionamiento físico.....	53
Figura 3.5. Niveles en las direcciones de grupo .....	54
Figura 3.6. Asignación de direcciones de grupo .....	56
Figura 3.7. Desacoplo de alimentación /datos .....	57
Figura 3.8. Conexión de alimentación y dispositivos al bus .....	58
Figura 3.9. Detalle de transmisión simétrica y cancelación del ruido en la transmisión...	59

Figura 3.10. Generación de corriente portadora sobre tensión de alimentación .....	60
Figura 3.11. Asignación de direcciones físicas .....	61
Figura 3.12. Resolución de colisiones CSMA/CA en EIB.....	66
Figura 3.13. Secuencia de envío de un telegrama ante la activación de un evento.....	67
Figura 3.14. Formato de transmisión de un byte .....	67
Figura 3.15. Formato de un telegrama.....	68
Figura 3.16. Campo control.....	68
Figura 3.17. Campo de dirección destino .....	69
Figura 3.18. Formato del campo de datos .....	70
Figura 3.19. Ejemplo de trama de datos de conmutación.....	72
Figura 3.20. Obtención del campo de comprobación de la trama .....	73
Figura 4.1. Editor WED con el diseño de la vivienda .....	98
Figura 4.2 Editor SED mostrando parte del código del proyecto.....	99
Figura 4.3. Editor MED mostrando el personaje utilizado.....	101
Figura 5.1. Vistas del level editor.....	107
Figura 5.2. Barra de menú.....	108
Figura 5.3. Level editor con un bloque añadido .....	111
Figura 5.4. Posición del bloque para la creación de un hueco.....	113
Figura 5.5. Creación de un hueco de puerta .....	114
Figura 5.6. Elemento prefabricado Steps01.wmp.....	115
Figura 5.7. Texturas empleadas en suelo y paredes .....	116
Figura 5.8. Texturas de la casa .....	117
Figura 5.9. Personaje añadido a la vivienda .....	118
Figura 5.10. Propiedades del personaje .....	119
Figura 5.11. Cubo de cielo y terrenos de la casa .....	120

Figura 5.12. Visión desde el balcón con el programa en ejecución .....	121
Figura 5.13. Propiedades del sol de este proyecto.....	122
Figura 5.14. Propiedades del sprite .....	124
Figura 5.15. Asociación de la acción al sprite .....	125
Figura 5.16. Función principal del código y lista de elementos .....	126
Figura 5.17. Función abrir y acción luz.....	127
Figura 6.1. Arquitectura waist-line.....	136
Figura 6.2. Esquema del sistema domótico implementado .....	139
Figura 6.3. Direcciones de grupo.....	144
Figura 6.4. Componentes que conforma el grupo Dimmer ON/OFF.....	145
Figura 6.5. Componentes que conforma el grupo Valor dimmer .....	147
Figura 6.6. Configuración de la salida binaria .....	148
Figura 6.7. Conexión del grupo persiana SUBIR/BAJAR.....	149
Figura 6.8. Conexión del grupo persiana STOP/CORTO .....	150
Figura 6.9. Programación de los datos del eibSOLO Netzwerk-Koppler .....	151
Figura 6.10. Configuración de los componentes del dimmer.....	152
Figura 6.11. Programación del dimmer .....	153
Figura 6.12. Tamaño de datos utilizado en el dimmer para su posible conexión.....	154
Figura 6.13. Programación del uso de las persianas Louvre .....	155
Figura 6.14. Programación de las pulsaciones a utilizar .....	155
Figura 6.15. Programación de la entrada binaria.....	156
Figura 6.16. Programación de la entrada binaria.....	157
Figura 6.17. Programación de la salida binaria .....	158
Figura 6.18. Programación del pulsador controlando el dimmer .....	159
Figura 6.19. Programación del pulsador controlando el shutter.....	159

Figura 6.20. Editor web NK2 con la página de este proyecto .....	160
Figura 6.21. Parámetros de configuración del ETS.....	163
Figura 6.22. Acceso a propiedades del proyecto del Nk2 .....	164
Figura 6.23. Parámetros de configuración del Nk2 .....	164
Figura 6.24. Implementación del sistema final .....	173
Figura 6.25. Entorno virtual con los botones de control .....	174

# **Bloque I**

## **Introducción**



# Capítulo 1

## Introducción

*‘¿Podría usted decirme, por favor, que camino debo tomar ahora?’*

*‘Eso depende en gran medida de dónde quieras llegar’, dijo el Gato.*

*‘No me importa donde llegar’, dijo Alicia*

*‘Entonces da igual el camino que tomes’*

*Lewis Carol, Alicia en el país de las maravillas*

Las nuevas funciones y necesidades de los edificios/viviendas y de sus usuarios, nos han conducido a desarrollar nuevos productos capaces de satisfacerlas. Y todo ello ha supuesto el nacimiento de diferentes sistemas con muy diversas cualidades, capaces de realizar dichas funciones y de comunicarse por distintos medios de transmisión, es lo que se ha dado en llamar domótica/inmótica o más recientemente hogar digital.[1]

Podemos ver los orígenes de la domótica en España a principios de los años noventa, pero no será hasta los años 2002-2003 cuando pasa a ser un concepto conocido por la sociedad. En la actualidad, el número de viviendas automatizadas es todavía relativamente bajo respecto al total de viviendas, pero el interés en su adopción está creciendo progresivamente. [2]

Cada día vemos más y más como la automatización de viviendas se va incorporando a nuestras vidas, cosas que eran impensables o únicamente accesibles para un sector reducido las encontramos poco a poco con más frecuencia en viviendas, restaurantes, etc. Lo que se consideraba un lujo comienza a encontrar un hueco en el usuario medio.

Los principales motivos que podemos encontrar para la incursión de esta nueva tecnología en nuestras vidas pueden ser, fundamentalmente, la confortabilidad y la seguridad. La domótica nos proporciona no sólo la comodidad de ajustar los diversos elementos de nuestro entorno con un simple movimiento de mano, sino también la tranquilidad de vernos seguros, tanto mientras estamos en el interior del entorno domotizado como cuando abandonamos el lugar durante un intervalo considerable de tiempo. La posibilidad, no sólo de tener una seguridad por alarmas, como encontrábamos hasta ahora, sino también la de emular un ambiente habitable, consiguiendo burlar con mayor facilidad a los intrusos.

Sin embargo una posibilidad más atractiva que está, es la de realizar nosotros mismos este control a las horas que consideremos y en el sitio que nos encontremos. Llegados a este punto, pasaremos a tratar el interfaz con el usuario, como el elemento fundamental que nos permite realizar estas acciones remotas.

Los entornos inteligentes enfatizan en interfaces más amigables para el usuario, servicios más eficientes, más control por parte del usuario y soporte para interacciones humanas. Las investigaciones actuales sobre entornos inteligentes tratan con interacción social, natural y multimodal [7,5]. La mayoría de las investigaciones acerca de interfaces sociales están enfocadas al diseño de los llamados agentes corpóreos conversacionales (ECAs, Embodied Conversational Agents) [3]. Estos agentes se hacen visibles en el interfaz como actores animados que representan seres humanos. La utilidad de estas interfaces de tipo "humanoide" y sus ventajas frente a otros tipos de interfaz sigue discutiéndose [6], pero parece probado [4] que cuando un humano se comunica con un agente de este tipo revela más información de tipo personal, usa un lenguaje más cuidadoso y acepta mejor las sugerencias y recomendaciones. No obstante, por el momento este tipo de interfaces no están proliferando y hay que contar con otro tipo de relaciones hombre máquina como la realidad virtual o los videojuegos. Este tipo de interfaces no sólo logran una comunicación con el usuario, sino que además consiguen que éste se encuentre en un ambiente relajado y

---

[4]

cómodo, puesto que la mayor parte de la población existente controla y disfruta de los diferentes entornos que presentan los distintos juegos que inundan el mercado.

Así pues, basándonos en este hecho, podemos afirmar que existe un entorno que permite la creación de una interfaz que permitirá realizar una conexión con el mundo exterior a través de un entorno que genera confianza y comodidad al usuario, que serán lo que denominaremos entornos 3D de visualización.

El objetivo principal de este trabajo es la creación de un interfaz 3D que desarrolle una visión virtual de la vivienda real, de tal modo que podamos movernos por la pantalla como si nos encontráramos en nuestra propia vivienda. Esta interfaz proporciona al usuario la impresión de encontrarse en el lugar que está representando y le crea una identificación con el entorno mayor que cualquier otro tipo de interfaz, puesto que el 3D es una reproducción natural y fácil de reconocer ya que el ser humano percibe, recuerda y se orienta en tres dimensiones, siendo por tanto, más eficiente visualizar información compleja en tres dimensiones que con otras técnicas. Además, el entorno diseñado permite la realización de acciones sobre los elementos de la vivienda de una manera similar a como se realiza en las situaciones reales (pulsando botones o conmutadores, subiendo y bajando controles, etc, de manera virtual evidentemente), lo cual hace aún más natural y familiar la experiencia para el usuario.

Por otra parte, el uso de plataformas software estándar, que pueden ejecutarse desde dispositivos convencionales como PCs, PDAs, Tablet, smartphones, etc, permiten disfrutar de esta experiencia a personas discapacitadas, mediante el uso de los mecanismos específicos que utilizan normalmente para interactuar con sus dispositivos informáticos.

## **1.1 Objetivos del proyecto**

En este documento se describen los trabajos realizados en el Proyecto Fin de Carrera (PFC), que consiste en la realización de un interfaz mediante un entorno en 3D interactivo para así facilitar al usuario la utilización de los sistemas domóticos de forma remota. Éste sistema resultará útil a infinidad de usuarios puesto que permitirá el manejo de los diferentes sistemas de la casa mientras se encuentra en cualquier otra ubicación.

Únicamente usando nuestro dispositivo sería posible a través del interfaz ver un modelo de la casa en cuestión, caminar por ella y accionar las diferentes posibilidades tales como, subida y bajada de persianas, encender y apagar las luces, etc; acciones que se realizarán por parte de la red domótica de la vivienda real.

La utilización de este interfaz facilitará el uso de todo el sistema, ya que no será necesario conectarse a internet para buscar una página, sino que, sencillamente usando este sistema con sólo caminar por él como en un videojuego realizaremos, de una manera más visual, todas las acciones deseadas en nuestro hogar, olvidándonos de todo ya que, el interfaz se encargará de realizar la conexión con la red domótica e indicarle las diferentes acciones que el usuario quiera realizar.

Como se ha comentado anteriormente, el objetivo principal de este proyecto será la creación del explicado sistema mediante las herramientas GameStudio, para la creación de la interfaz gráfica, y del lenguaje de programación C tanto para la realización de las transmisiones al domicilio en cuestión para la utilización deseada del sistema domótico, como para la creación de los diversos scripts necesarios en nuestra interfaz. Como resultado, el susodicho entorno se probará con un entrenador domótico que reúna diferentes componentes reales de automatización de viviendas y que se comporte a todos los efectos como una vivienda domotizada real. Demostrando de esta manera su viabilidad y aplicabilidad práctica.



Figura 1.1. Esquema del funcionamiento del sistema total

## 1.2 Organización de la memoria

Ésta memoria se va a dividir en cuatro bloques principales, cada uno de ellos compuesto por los capítulos necesarios para su correcta explicación. A continuación pasamos a explicar brevemente la composición de cada uno de estos bloques con una breve explicación de los capítulos que los componen:

Bloque I. Introducción: Se muestra una presentación del proyecto.

Capítulo 1. Introducción: En éste capítulo realizamos una presentación inicial y mostramos los objetivos para que así el lector pueda obtener una idea global del mismo. En la última parte del capítulo mostramos una estructuración de la memoria.

Bloque II. Estudio previo: Se presenta el estudio previo a la realización del proyecto.

Capítulo 2. Sistemas domóticos: A lo largo de éste capítulo veremos un estudio sobre las instalaciones domóticas observando así su diseño y funcionamiento.

Capítulo 3. Sistema KNX: Aquí estudiaremos el funcionamiento en profundidad del sistema KNX, que será el empleado en la realización de este PFC.

Capítulo 4. Entornos 3D de visualización: Aquí estudiaremos los diferentes tipos de videojuegos existentes en el mercado, así como la reacción de los usuarios con cada uno de ellos, escogiendo, finalmente, el tipo más apropiado a desarrollar para lograr los objetivos deseados. A continuación se explicarán las diferentes herramientas necesarias para el desarrollo de la interfaz así como la justificación de la herramienta utilizada destacando las características empleadas en la realización de nuestro sistema.

Bloque III. Desarrollo: Se muestra los pasos realizados para la creación de éste proyecto así como los resultados finales obtenidos.

Capítulo 5. Diseño e implementación del entorno virtual: En este capítulo explicaremos cada uno de los pasos realizados durante el desarrollo del entorno virtual para lograr el funcionamiento deseado del sistema. Por último se mostrará la programación realizada la cual permitirá la unión de la interfaz creada con el sistema domótico desarrollado.

Capítulo 6. Diseño e implementación: En esta parte veremos la programación realizada para la domotización de la vivienda y cómo se ha logrado el correcto funcionamiento de la misma, así como la creación de una página online para el control del sistema domótico en cualquier punto y la unión de los diferentes trabajos realizados para la creación del producto final.

Bloque IV. Conclusiones: Exponemos las conclusiones extraídas del trabajo.

Capítulo 7. Conclusiones: Durante este capítulo veremos de una forma clara y concisa las conclusiones del trabajo realizado

# **Bloque II**

## **Estudio previo**



## Capítulo 2

### Sistemas domóticos

*“Los conceptos fundamentales que emergieron hace 30 años no han cambiado significativamente; aunque la tecnología evoluciona, los conceptos permanecen”*

*Bertrand Meyer, Software Engineering in the Academy, 2001*

El término Domótica proviene de la palabra latina domus (que significa casa) e informática. Se entiende por domótica al conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda.

Las diferentes automatizaciones que se llevan a cabo en la vivienda consisten en un gran abanico de servicios, de los cuales podemos destacar como más relevantes los siguientes [28]:

- **Ahorro energético**

Cada día existe un mayor interés en reducir el gasto energético sin reducir por ello las comodidades que nos facilita, muchas veces esto se consigue mediante una gestión eficiente de los aparatos sin necesidad de sustituir estos por otros con un menor consumo. Para ello tan sólo debemos plantearnos el control de algunos detalles, como por ejemplo:

El sistema debe identificar la presencia humana en la habitación para así poder regular la temperatura ambiente o mantener las luces apagadas hasta la entrada de una persona y dependiendo de la hora solar en la que se encuentre, es decir, si son las 12 del mediodía no es necesario encender la luz del salón mientras que si entramos en esta misma habitación a las 10 de la noche necesitaremos la luz. Gracias a cada uno de estos detalles se puede reducir el consumo eléctrico sin la pérdida de las comodidades que nos aportan cada uno de estos elementos.

- **Confort**

Este apartado abarca todas aquellas mejoras que se puedan introducir en la vivienda y que mejoren la comodidad de sus habitantes. Esto nos presenta un amplio abanico de posibilidades, tales como la regulación de la luz según el nivel de luminosidad ambiente, integración de portero al teléfono o del videoportero al televisor, programación del apagado y encendido de diferentes sistemas como pueden ser las luces, los aspersores o la subida y bajada de persianas.

Sin embargo dentro de este gran abanico sin duda destaca principalmente las instalaciones de climatización, ventilación y calefacción en donde resaltan los siguientes sistemas o productos:

- ✓ Programadores para calefacción, climatización o encendido y apagado de cualquier aparato eléctrico.
- ✓ Sistemas de comunicación de unidades de climatización mediante el uso de un ordenador y con la posibilidad de ser configuradas por el propio usuario.
- ✓ Sistemas para visualización y gobierno de caudales de aire, de manera que se permita su control para gobernar los motores de impulsión y las compuertas de los conductos de ventilación.

- **Seguridad**

En este apartado nos encontramos infinidad de sistemas para proteger nuestra vivienda de posibles hurtos, y comprende desde alarmas de seguridad, hasta simuladores de presencia. Algunos de los diferentes sistemas que podemos encontrar en tanto a seguridad son:

- ✓ Sistemas clásicos de protección contra incendios que, utilizando los tradicionales detectores (velocimétricos o iónicos) convenientemente interconectados mediante un cableado estructurado, son capaces de detectar rápidamente el foco inicial del fuego y conectar con los cuerpos encargados de controlar este tipo de situaciones mediante sistemas de aviso centralizados e instalaciones automáticas de seguridad. Además, son de gran utilidad operaciones automatizadas que incluyan el control de las infraestructuras de aire acondicionado o el cierre automático de las puertas cortafuegos de las dependencias afectadas por el posible incendio.
- ✓ Sistemas de vigilancia contra intrusiones tanto exteriores, como interiores: Si el sistema domótico instalado cuenta con sensores destinados a encender las luces en las zonas de paso de forma automática, pueden configurarse para la detección de intrusos en cuanto se active el sistema de alarma. En caso contrario se pueden instalar diferentes tipos de sensores, como por ejemplo los conocidos volumétricos, o diferentes contactos y sensores de apertura en puertas y ventanas. En caso de alarma se puede programar el sistema para que realice toda una serie de llamadas telefónicas para verificar el estado, o simplemente la alarma puede enviarse a la Central de Alarmas, que será la encargada de llamar al propietario o avisar a la policía en caso necesario.
- ✓ Sistemas de detección de fuego, fugas de gas, intrusión, incidentes eléctricos, humos de diferentes naturalezas o explosiones utilizando técnicas de vídeo multifunción. Estos mecanismos permiten que las cámaras de seguridad puedan detectar focos de luz muy pequeños a varias decenas de metros o una intrusión humana a una distancia que supere la centena.

- ✓ Botones de pánico: En caso de que escuchemos ruidos extraños, o que notemos la presencia de intrusos en casa, los llamados “botones de pánico” pueden iluminar completamente la vivienda y el exterior, al tiempo que lanzan una señal de alarma a la Central de Alarmas, o incluso directamente a la policía.
  
- ✓ Métodos de control de acceso gracias a la utilización de fichas perforadas, tarjetas con banda magnética, dispositivos que incluyan mecanismos de detección de proximidad, sistemas de lectura de datos y registro de imágenes del documento de identidad, de la huella digital o del iris del ojo o de terminales capaces de identificar la voz.
  
- ✓ Sistemas de simulación de presencia, consiste en que el sistema actúe sobre luces, persianas, o incluso toldos, de forma que desde fuera se tenga la impresión de que la casa se encuentra habitada aunque sus propietarios estén de vacaciones. Esta actuación podrá realizarse de forma automática, o controlada de forma remota vía teléfono o internet.

- **Accesibilidad**

Aquí se incluirán las aplicaciones o instalaciones de control remoto del entorno que favorecen la autonomía de personas con limitaciones funcionales, o discapacidad. El objetivo de estas tecnologías es favorecer la autonomía personal. 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. Algunas de las diferentes ventajas que puede ofrecer la domótica a éstas personas son:

- ✓ Gestión digital del domicilio y comunicación entre los diferentes dispositivos.
  
- ✓ Interacción remota con el hogar.

- ✓ Recepción en el domicilio de nuevos servicios gracias a la conexión permanente de la vivienda con el exterior a través de una línea ADSL, cable-módem o cualquier otro acceso de banda ancha.
- ✓ Monitorización y control remoto de los equipos electrónicos, puesta en marcha de los mismos de forma conjunta y realización de actividades a distancia, como, por ejemplo, la compra a través de Internet desde el frigorífico.
- ✓ Mejora el confort, ya que el sistema puede adaptarse a cualquier estilo de vida, reduciendo el trabajo doméstico y los desplazamientos y facilitando el manejo de los dispositivos.
- ✓ Decisión acerca de qué acciones se desea temporizar y desde qué punto quiere controlar cada elemento.
- ✓ El sistema permite la conexión de elementos específicos para personas con discapacidad, como puertas con apertura automática o mandos a distancia para el control de la iluminación.
- ✓ La realización de cada maniobra puede ir seguida de un sonido o una luz de confirmación para personas con problemas visuales o auditivos, respectivamente.
- ✓ Activación y desactivación de cualquier equipo del domicilio con un comando de VOZ.
- ✓ Recepción de servicios de utilidad para personas con necesidades especiales, como telemedicina, mediante la conexión de dispositivos en el hogar que permiten la realización de pruebas sencillas (chequeos, controles puntuales atención de

urgencias, tomas de tensión o medición de niveles) y el envío de los datos al hospital a través del PC; o teleasistencia a través de la contratación del equipamiento complementario en el domicilio, como alarmas que el usuario acciona en caso de peligro o teléfonos manos libres que le permiten hablar aún cuando se encuentre lejos del aparato.

- ✓ Colocación de cámaras en el interior de la vivienda que permiten informar visualmente del estado del paciente.
  
- ✓ Teleconsulta, televisita y detección de comportamientos anómalos

- **Comunicaciones**

En este apartado encuentran su lugar todos los sistemas o infraestructuras de comunicación que posee nuestra vivienda, tales como la tele asistencia, Tele mantenimiento, control tanto externo como interno, es decir, el control a través de Internet mediante PC, PdAs o mandos inalámbricos.

Como podemos ver es un sector que abarca un gran número de productos que hemos incluido en apartados anteriores ya que están creados específicamente para facilitar el confort, la seguridad, etc.

## **2.1 Composición de un sistema domótico**

Podemos explicar un sistema domótico diciendo que son diferentes equipos interconectados a una red para obtener información del entorno que les rodea y poder realizar acciones sobre dicho entorno. Para ello vamos a ir viendo las diferentes partes de las que se componen éstos sistemas [29].

### 2.1.1 Elementos de un sistema domótico

Un sistema domótico se compone de diferentes elementos, los cuales podemos clasificar principalmente en:

- **Unidad de control:** Este dispositivo es el encargado de gestionar toda la información que transcurre a través de la red y a dar las instrucciones oportunas a los demás elementos de la misma para resolver los problemas o circunstancias surgidas o trasladar las órdenes correspondientes.
- **Sensores o Transductores:** Son los elementos encargados de medir un fenómeno físico y convertir en una magnitud eléctrica (Intensidad o tensión), es decir, que son capaces de captar los cambios físicos en el recinto y transmitir la información recopilada a la unidad de control o a los actuadores para que realicen las operaciones oportunas. Existen diferentes tipos, como pueden ser de temperatura, humedad, gas, luminosidad, incendio, humo, intrusión, consumo, etc.
- **Actuadores:** Son los aparatos cuya función consiste en recibir la información proveniente de la unidad de control o de los propios sensores e interpretarla de forma que se transforme en instrucciones o acciones físicas para, por ejemplo, subir una persiana o activar el aire acondicionado.
- **Medio de transmisión:** Se trata del soporte de operación de la red domótica, a través del cual circula toda la información tratada con la intención de que llegue a los dispositivos finales para que éstos actúen de forma deseada.

## 2.1.2 Arquitectura del sistema domótico

Cuando hablamos de arquitectura en un sistema domótico nos referimos a la estructura que conforma su red. La clasificación se realiza según donde resida la “inteligencia” del sistema. Las principales arquitecturas que podemos encontrar son:

- **Arquitectura Centralizada:** En un sistema de arquitectura centralizada, un controlador centralizado, que normalmente será un PC o un dispositivo de características similares, envía la información a los actuadores e interfaces según el programa, la configuración y la información que recibe de los sensores, sistemas interconectados y usuarios. Es decir, el funcionamiento global del sistema depende de la programación introducida en la central domótica. Este sistema de control es el corazón o núcleo de dicho recinto, de tal manera que cuando no se encuentra disponible la totalidad del sistema deja de operar, y su instalación no es compatible con la eléctrica convencional en cuanto que en la fase de construcción hay que elegir esta topología de cableado.

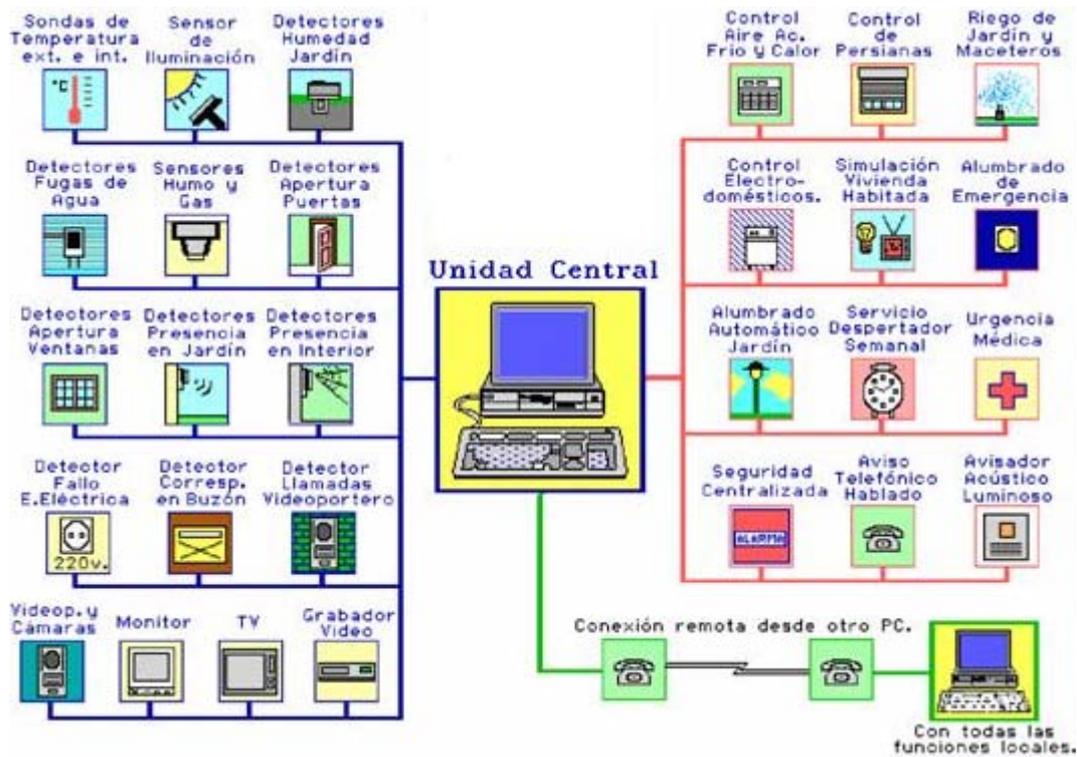


Figura 2.1. Esquema de Arquitectura de Sistema Domótica Centralizada

- Arquitectura Distribuida o Descentralizada:** No es necesario tener una central inteligente conectada para funcionar y tomar decisiones sobre las acciones a desarrollar. Solo hace falta un PC para programar las unidades, y como cada una de estas posee un microprocesador son completamente autónomas. Esta característica proporciona al instalador domótico una libertad de diseño que le posibilita adaptarse a las características físicas de cada lugar en particular. Hay sistemas que son de arquitectura distribuida en cuanto a la capacidad de proceso, pero no lo son en cuanto a la ubicación física de los diferentes elementos de control, y de manera inversa, sistemas que son de arquitectura distribuida en cuanto a su capacidad para ubicar elementos de control físicamente distribuidos, pero no en cuanto a los propios procesos de control, que son ejecutados en uno o varios equipos.

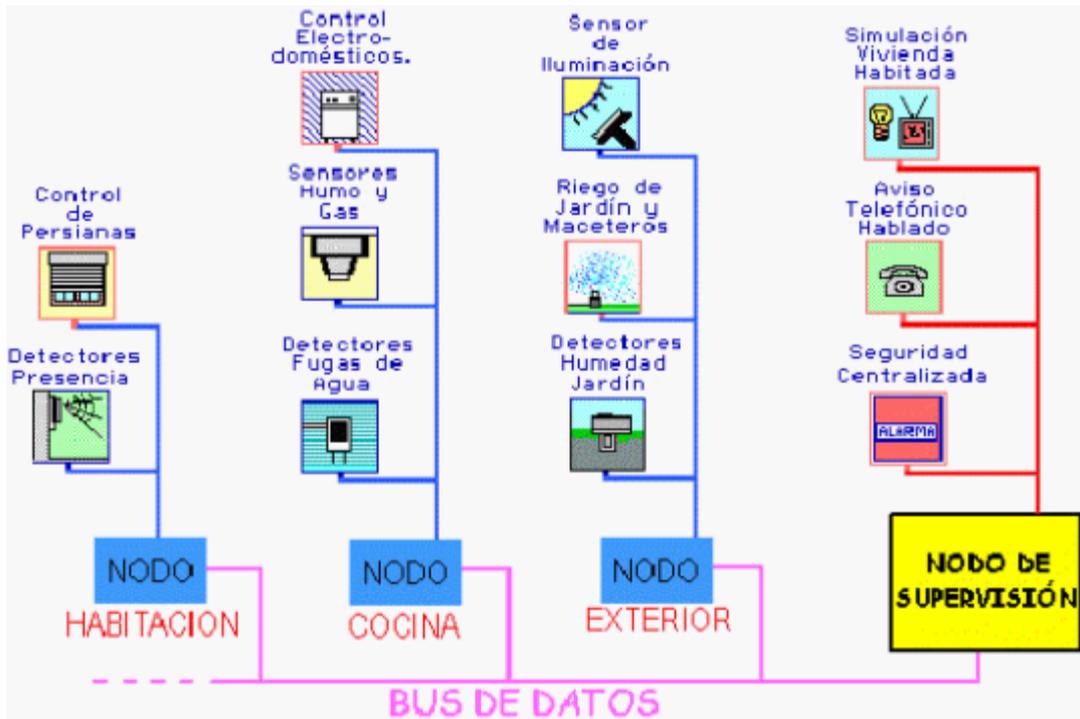
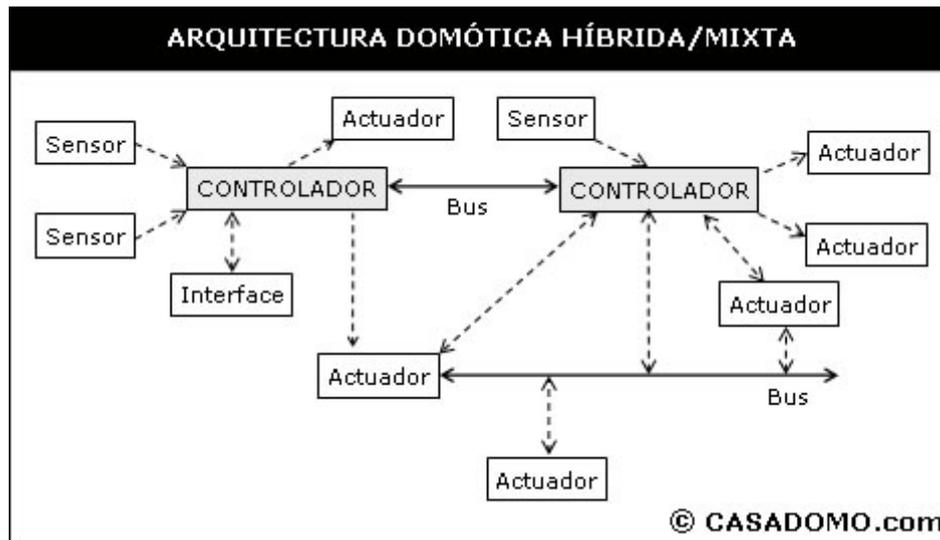


Figura 2.2. Esquema de Arquitectura de Sistema Domótica Descentralizada

- **Arquitectura Híbrida / Mixta:** En un sistema de domótica de arquitectura híbrida (también denominado arquitectura mixta) se combinan las arquitecturas de los sistemas centralizadas y distribuidas o descentralizadas. En este tipo tenemos sistemas con arquitectura descentralizada en cuanto a que disponen de varios pequeños dispositivos capaces de adquirir y procesar la información de múltiples sensores y transmitirlos a un grupo de dispositivos distribuidos por la vivienda; bajo el gobierno de un controlador central



*Figura 2.3. Esquema de Arquitectura de Sistema Domótica Híbrida/Mixta*

Hoy en día hay buenos sistemas centralizados y distribuidos, todos ellos con elevadas prestaciones. Ambas arquitecturas tienen sus ventajas y sus inconvenientes, lo cual a priori no ayuda a decidir cuál es la mejor solución para una vivienda. Podemos ver de forma resumida las ventajas y desventajas que presenta cada uno en la siguiente tabla:

	<i>Ventajas</i>	<i>Desventajas</i>
<b><i>Sistema centralizado</i></b>	<ul style="list-style-type: none"> <li>· fácil instalación</li> <li>· más económico</li> </ul>	<ul style="list-style-type: none"> <li>· poca flexibilidad</li> <li>· dependencia del núcleo</li> </ul>
<b><i>Sistema distribuido</i></b>	<ul style="list-style-type: none"> <li>· elevada flexibilidad</li> <li>· elevada integración</li> </ul>	<ul style="list-style-type: none"> <li>· coste elevado</li> <li>· diseño más complejo</li> </ul>

*Tabla 2.1. Ventajas e inconvenientes de los sistemas centraliza y distribuido*

### 2.1.3 Medio de transmisión

En todo sistema domótico los diferentes elementos de control deben intercambiar información entre ellos a través de uno de los soportes físicos que se expone a continuación:

- **Líneas de distribución de energía eléctrica (corrientes portadoras):**

Si bien no es el medio más adecuado para la transmisión de datos, sí es una alternativa a tener en cuenta para las comunicaciones domésticas dado el bajo coste que implica su uso al tratarse de una instalación existente y de muy fácil conexionado. Para aquellos casos en los que las necesidades del sistema no impongan requerimientos muy exigentes en cuanto a la velocidad de transmisión ni sean críticos los niveles de seguridad y fiabilidad de la misma, la línea de distribución de energía eléctrica puede ser suficiente como soporte de la misma.

- **Soportes metálicos:** la infraestructura de las redes de comunicación actuales, tanto públicas como privadas, tiene en un porcentaje muy elevado cables metálicos de cobre como soporte de transmisión de las señales eléctricas que procesa. En general se puede distinguir dos tipos de cables metálicos:

***Par metálico:*** Los cables formados por varios conductores de cobre pueden dar soporte a un amplio rango de aplicaciones en el entorno domótico. Este tipo de cables es capaz de transportar voz, datos y alimentación de corriente continua. Los denominados cables de pares están formados por cualquier combinación de los tipos de conductores que a continuación se detalla:

- ✓ **Cables:** Formados por un solo conductor con un aislamiento exterior plástico, como los utilizados para la transmisión de las señales telefónicas.

- ✓ **Par de cables:** Cada uno de los cuales formado por un arrollamiento helicoidal de varios hilos de cobre. Los utilizados para la distribución de señales de audio constituyen un buen ejemplo para este caso.
  
- ✓ **Par apantallado:** Formado por dos hilos recubiertos por un trenzado conductor en forma de malla cuya misión consiste en aislar las señales que circulan por los cables de las interferencias electromagnéticas exteriores. Un ejemplo podría ser los utilizados para la distribución de sonido de alta fidelidad.
  
- ✓ **Par trenzado:** Formado por dos hilos de cobre recubiertos cada uno por un trenzado en forma de malla. El trenzado es un medio para hacer frente a las interferencias electromagnéticas, y una de sus aplicaciones es la interconexión de ordenadores. Por otro lado, y aunque los sistemas domóticos de operación del Proyecto puedan implementarse junto con la red eléctrica, cabe reseñar en este punto que el soporte físico escogido para la transmisión de datos previa y posterior al enlace óptico a implementar es el par trenzado.

**Coaxial:** Un cable coaxial es un circuito físico asimétrico constituido por un conductor filiforme que ocupa el eje longitudinal del otro conductor en forma de tubo, manteniéndose la separación entre ambos mediante un dieléctrico apropiado. Este tipo de cables permite el transporte de señales de vídeo y datos a alta velocidad, y dentro del ámbito de la vivienda puede ser utilizado como soporte de transmisión para:

- ✓ Señales de teledifusión que provienen de las antenas (TV y FM).

- ✓ Señales procedentes de las redes de TV por cable.
  
- ✓ Señales de control y datos a media y baja velocidad.
  
- **Fibra óptica:** Constituye el resultado de combinar dos disciplinas no relacionadas como son la tecnología de los semiconductores, que proporciona los materiales necesarios para las fuentes y los detectores de luz, y la del guiado de ondas ópticas, la cual proporciona el medio de transmisión, el cable de fibra. La fibra óptica está constituida por un material dieléctrico transparente, conductor de luz, compuesto por un núcleo con un índice de refracción menor que el del revestimiento, que envuelve dicho núcleo.

Este medio de transmisión aporta gran fiabilidad a la transferencia de datos e inmunidad frente a interferencias electromagnéticas y de radiofrecuencia, además de permitir el envío de gran cantidad de información; sin embargo, la distancia entre los puntos de la instalación es limitada (aunque en el entorno doméstico estos problemas no existen) y su coste tiende a ser elevado.

- **Comunicación no guiada:** La señal de datos se transmite a través de un canal de comunicación atmosférico, como puede ser el aire, sin la necesidad de utilización de cableado.

**Radiofrecuencia:** La introducción de la radiofrecuencia como soporte de transmisión en la domótica ha venido precedida de la proliferación de los teléfonos inalámbricos y sencillos telemandos. Este medio de transmisión puede parecer, en principio, idóneo para el control a distancia de los sistemas domóticos, dada la gran flexibilidad que supone su uso. Sin embargo, resulta particularmente sensible a las perturbaciones electromagnéticas producidas, tanto por los medios de transmisión como por los equipos domésticos.

La fácil interceptación de las comunicaciones, su alta sensibilidad a las interferencias y la dificultad para la integración de las funciones de control en su modalidad analógica son sus principales inconvenientes.

**Infrarrojos:** El uso de mandos a distancia basados en transmisión por infrarrojos está ampliamente extendido en el mercado residencial para supervisar equipos de audio y vídeo o para el control remoto de sistemas de climatización y aparatos de aire acondicionado. La comunicación se realiza entre un diodo emisor, el cual emite una luz en la banda IR sobre la que se superpone una señal convenientemente modulada con la información de control, y un fotodiodo receptor cuya misión consiste en extraer de la señal recibida la información de control.

Los controladores de equipos domésticos basados en la transmisión de ondas en la banda de los infrarrojos presentan gran comodidad y flexibilidad y admiten un gran número de aplicaciones.

Al tratarse de un medio de transmisión óptico es inmune a las radiaciones electromagnéticas producidas por los equipos domésticos o por los demás medios de transmisión. Sin embargo, habrá que tomar precauciones en el caso de las interferencias que pueden afectar a los extremos del medio.

La elemental diferencia entre este tipo de utilización de una comunicación óptica no guiada y la que el presente Proyecto propone no es otra que la sustitución de los propios tramos de red de cableado por este modelo de transmisión, dejando a un lado así su aplicación para otro tipo de dispositivos como mandos a distancia.

## 2.1.4 Tecnologías existentes

Dada la enorme heterogeneidad del campo que se está tratando resulta muy complicado establecer una clasificación exacta de las tecnologías aplicables al entorno del hogar y de la domótica en general, así que es viable ofrecer un panorama de éstas organizadas en tres ámbitos [11]:

- **Redes y tecnologías de comunicación:** Se refieren a estándares utilizados para la comunicación entre equipos y que generalmente especifican los niveles inferiores del modelo de referencia OSI (Open Systems Interconnection). De hecho, y como ejemplo, se destaca la similitud entre este modelo y la estructura de capas que compone al estándar EIB, aspecto que será estudiado en profundidad en el siguiente capítulo.
- **Arquitecturas de objetos distribuidos de propósito general:** Estructuras de software empleadas para la construcción de aplicaciones distribuidas como DCOM, CORBA o Jini con potencial aplicación en la domótica.
- **Arquitecturas de objetos y servicios distribuidos:** También para aplicaciones en el entorno que se está analizando, el domótico.

Para finalizar este apartado cabe destacar el hecho de que la domótica no se mantiene ajena al desarrollo de nuevas técnicas y métodos que adecuen el estado del arte de estas tecnologías a la integración de las distintas redes que ya pueden coexistir en el hogar o en otras edificaciones. Se muestra a continuación, en la *figura 2.4.*, un esquema aclarativo al respecto de lo enunciado:

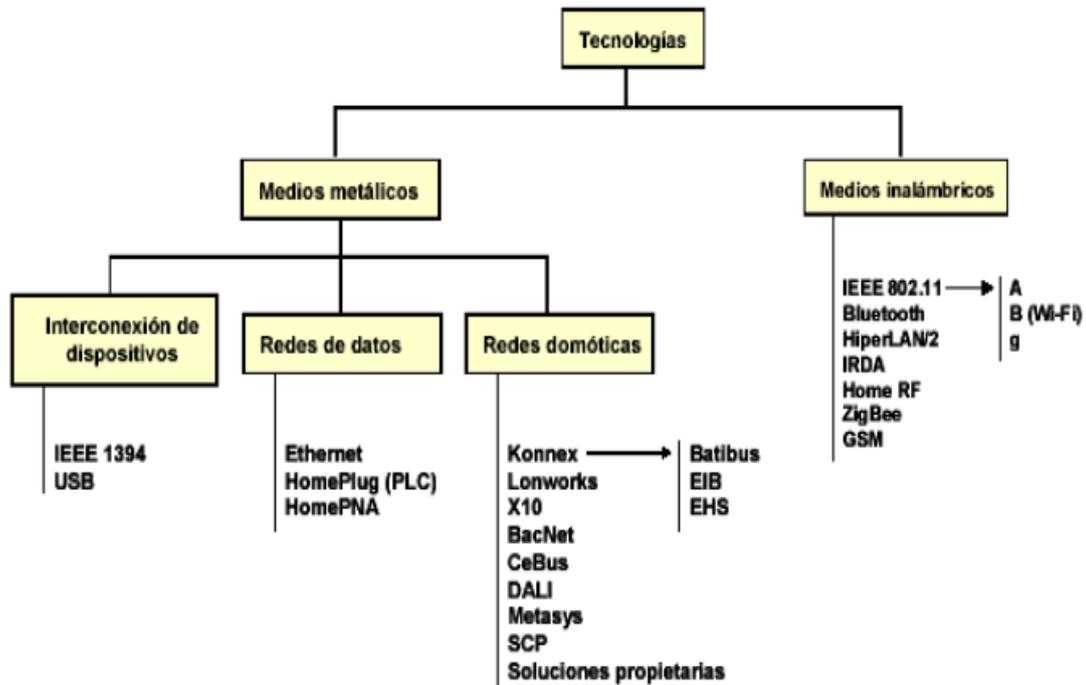


Figura 2.4. Tecnologías presentes en las redes de comunicación

## 2.2 Estándares. Sistema KNX

Tras presentar una idea general de cómo se compone un sistema domótico pasaremos a ver el protocolo de comunicaciones de dicho sistema para finalizar hablando del protocolo KNX en el cual se basa éste trabajo.

Un protocolo de comunicaciones es sencillamente el idioma o formato de los mensajes que los elementos de control utilizan para poder entenderse y cambiar su información de forma coherente. Existen una gran variedad de protocolos, en este documento nos centraremos en los más relevantes [11]:

- **X-10:** El padre de los protocolos, a través de líneas de corriente facilita el control de dispositivos domóticos sin instalación en cualquier casa.

- **C-Bus Clipsal:** Fabricado y diseñado en Australia por Gerard Industry Pty Ltd. con CLIPSAL. Ofrece un completo control de iluminación y de todos los servicios eléctricos, audio y video. Utilizado en edificios, viviendas y edificaciones de todo tipo.
- **HomePNA (Home Phonline Networking Alliance):** Formada por la alianza de varias empresas que trabajan en el desarrollo de una tecnología que permita implementar redes de área local usando la instalación telefónica de una vivienda.
- **Lonworks: Echelon Corp.:** Redes de control comerciales y para el hogar. Una red LonWorks es un grupo de dispositivos trabajando juntos para sensorizar, monitorizar, comunicar, y de algunas maneras controlar. Es muy parecido a lo que puede ser una LAN de PC,s.
- **BatiBUS:** Club International (BCI). Sensores de unión y actuadores para construir sistemas que control en HVAC (Acondicionamiento), seguridad física y personal, acceso.
- **EHS (European Home System):** Una colaboración entre industrias y gobiernos Europeos sobre Domótica. Entre alguna de sus misiones la EHSA tiene el objetivo de la armonización y estandarización en Europa de un BUS común (EHS).
- **InstabusEIB: (European Installation Bus):** Sensores y actuadores para construir sistemas que controlen HVAC (Acondicionamiento), seguridad física y personal y acceso.

### 2.2.1 X10

Estamos ante uno de los protocolos más antiguos usado en aplicaciones domóticas. Se diseñó en Escocia entre los años 1976 y 1978 en Pico Electronics Ltd., Glenrothes, con la intención de transmitir datos por las líneas de baja tensión a muy baja velocidad y costes muy bajos. Dado que éste protocolo usa las líneas de eléctricas de la vivienda, no es necesario tender nuevos cables para conectar dispositivos.

El sistema X-10 ha sido desarrollado para ser flexible y fácil de usar. Podemos empezar con un producto en particular, por ejemplo un mando a distancia, y expandir luego el sistema para incluir la seguridad o el control con el ordenador, siempre que desee, con componentes fáciles de instalar y que no requieren cableados especiales.

Actualmente puede considerarse una de las soluciones domóticas más económicas, rápidas y sencillas, ya que precisa de una instalación mínima debido a la utilización de corrientes portadoras superpuestas a la red eléctrica y una tasa de transferencia bastante reducida. Además, X10 ha fabricado desde sus inicios versiones de su sistema de control para diversas compañías como Leviton Manufacturing Co. o IBM.

Éste estándar permite la conexión de hasta 256 dispositivos que se comunican entre sí con un ratio de 50 bps en Europa y 60 bps en EEUU empleando asimismo portadoras de 120 KHz. Utiliza una modulación muy sencilla, comparado con las de otros protocolos de control por ondas portadoras. El transceiver X-10 está pendiente de los pasos por cero de la onda senoidal de 50 Hz típica de la alimentación eléctrica (60 Hz en EEUU) para insertar un instante después una ráfaga muy corta de señal en una frecuencia fija.

Se puede insertar esta señal en el semiciclo positivo y el negativo de la onda senoidal. La codificación de un bit 1 o de un bit 0, depende de cómo se inyecte esta señal en los dos semiciclos. Un 1 binario es representado por un pulso de 120 KHz durante 1 milisegundo y el 0 binario se representa por la ausencia del mismo. En un sistema trifásico el pulso de 1 milisegundo se transmite tres veces para que coincida con el paso por el cero en las tres fases.

Por lo tanto, el Tiempo de Bit coincide con los 20 ms que dura el ciclo de la señal, de forma que la velocidad binaria de 50 bps viene impuesta por la frecuencia de la red eléctrica que tenemos en Europa. En Estados Unidos la velocidad binaria son 60 bps.

La transmisión completa de un orden X-10 necesita once ciclos de corriente. Esta trama se divide en tres campos de información:

1. dos ciclos representan el Código de Inicio.
2. cuatro ciclos representan el Código de Casa (letras A-P),
3. cinco ciclos representan o bien el Código Numérico (1-16) o bien el Código de Función (Encender, Apagar, Aumento de Intensidad, etc...).

Para aumentar la fiabilidad del sistema, esta trama (Código de Inicio, Código de Casa y Código de Función o Numérico) se transmite siempre dos veces, separándolas por tres ciclos completos de corriente. Hay una excepción, en funciones de regulación de intensidad, se transmiten de forma continuada (por lo menos dos veces) sin separación entre tramas.

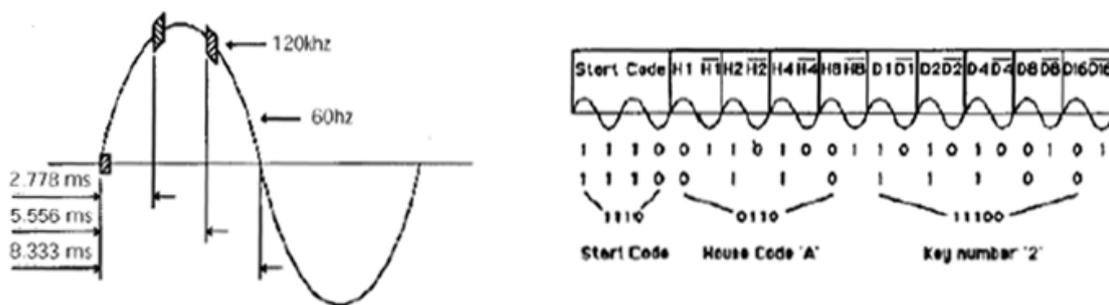


Figura 2.5 Codificación X10 en la red eléctrica de 60 Hz

Al implementarse junto con el tendido eléctrico, X10 puede diseñarse en función de cualquiera de las topologías que conocemos: bus, anillo, estrella, árbol, etc.; por ésta y por más razones se proporciona facilidad de manejo, confort, flexibilidad, modularidad y capacidad de crecimiento entre otras ventajas.

Sin embargo, este estándar también posee algunos inconvenientes. Entre ellos destacan sus escasas posibilidades de extensión y la pequeña capacidad que aporta para integrar dispositivos, ya que sólo dispone de módulos externos interpuestos entre la alimentación y la red.

### 2.2.2 CEBus

CEBus es fabricado y diseñado en Australia por Gerard Industry Pty Ltd. con CLIPSAL. Ofrece un completo control de iluminación y de todos los servicios eléctricos, audio y video. C-Bus controla cualquier tipo de carga eléctrica en forma digital (relays) o variable (analógicas).

La arquitectura del CEBus sigue el modelo de referencia OSI (Open Systems Interconnection), ocupándose cada uno de los niveles de determinadas funciones de la red de comunicación. El CEBus sólo utiliza cuatro de los siete niveles: Físico, Enlace, Red y Aplicación. La interfaz entre los diferentes niveles del nodo CEBus está definido como un conjunto de primitivas de servicio, proporcionando cada nivel servicio al inmediatamente superior.

Para asegurar rapidez y una confiable operación, cada unidad tiene su propio microprocesador que puede ser programado individualmente para integrarse con las otras unidades CEBus, permitiéndole operar independientemente con inteligencia distribuida. Esta inteligencia distribuida permite una alta velocidad de comunicación y asegura que un incorrecto funcionamiento de una unidad no afecte a otra.

El protocolo y el lenguaje son comunes a todos los elementos CEBus, pero existen 6 medios físicos distintos: red eléctrica (PL), par trenzado con telealimentación (TP), cable coaxial (CX), infrarrojo (IR), radiofrecuencia (RF) y fibra óptica (FO). La elección del medio se realiza en función de parámetros como el ahorro energético, comodidad, facilidad de instalación de los productos CEBus, seguridad, coste y sencillez del sistema.

Para la transmisión de datos por corrientes portadoras, el CEBus usa una modulación en espectro expandido; estos se transmite uno o varios bits dentro de una

ráfaga de señal que comienza en 100 kHz y termina en 400 kHz (barrido) de duración 100 microsegundos.

Las tramas definidas en CEBus pueden tener longitud variable en función de la cantidad de datos que se necesitan transmitir. El tamaño mínimo es 8 octetos y el máximo casi 100 octetos. Al igual que los dispositivos EIB, los nodos CEBus tienen grabado una dirección física prefijada en fábrica.

Como parte de la especificación CEBus se ha definido un lenguaje común para el diseño y especificación de la funcionalidad de un nodo, a este lenguaje lo han llamado CAL (Common Application Language) y está orientado a objetos.

La velocidad de transmisión de datos que se consigue es de 10Kbps, y puede ser utilizado tanto en viviendas ya construidas como de nueva construcción.

Se trata de un estándar muy ambicioso, y en él cooperan tanto Europa como Japón, pero no existen muchos productos comercializados, lo que se debe principalmente a su elevado precio

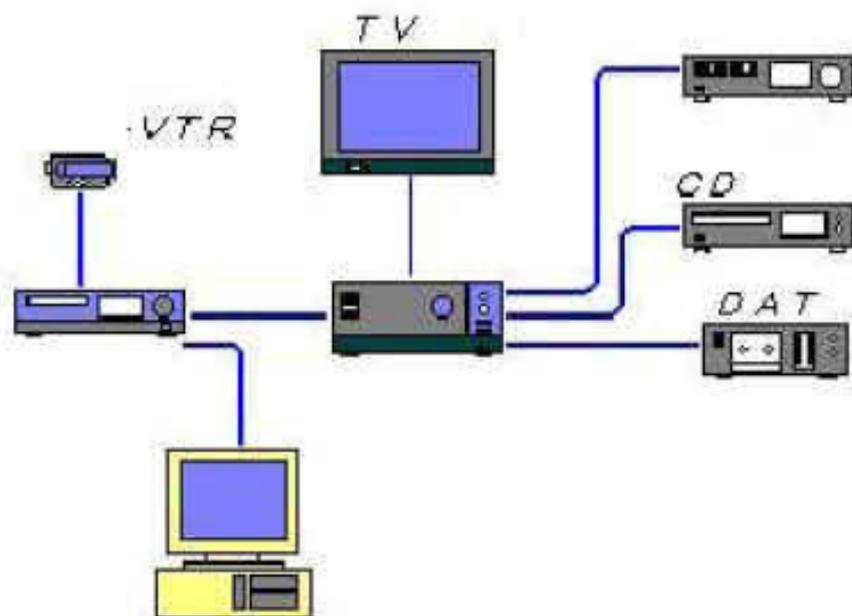


Figura 2.6 Red doméstica multimedia IEEE 1394

### 2.2.3 HomePNA

HomePNA surgió como una alianza de varias empresas que trabajaron en el desarrollo de una tecnología que permitiera la implementación de redes de área local utilizando la instalación telefónica de una vivienda. El objetivo primario era ofrecer entre 2 y 3 Mbps de ancho de banda, aunque posteriormente fue mejorado hasta alcanzar los 10 Mbps, cifra similar a la que proporciona Ethernet.

Del mismo modo que las tecnologías de bucle de acceso xDSL, HomePNA utiliza el ancho de banda libre del cableado telefónico con el fin de inyectar su señal modulada por encima de los 2 MHz. Dicha modulación se lleva a cabo a través de la técnica FDM, consistente en la ocupación por parte de un gran número de señales portadoras de un amplio ancho de banda.

Esta iniciativa nació en Norteamérica, donde la mayoría de las viviendas disponen de una toma de teléfono por habitación (*Figura 2.7*). En Europa, al existir en general dos o tres por hogar solamente, no se prevé una difusión mayoritaria, aparte de los problemas de incompatibilidad surgidos con la tecnología VDSL.

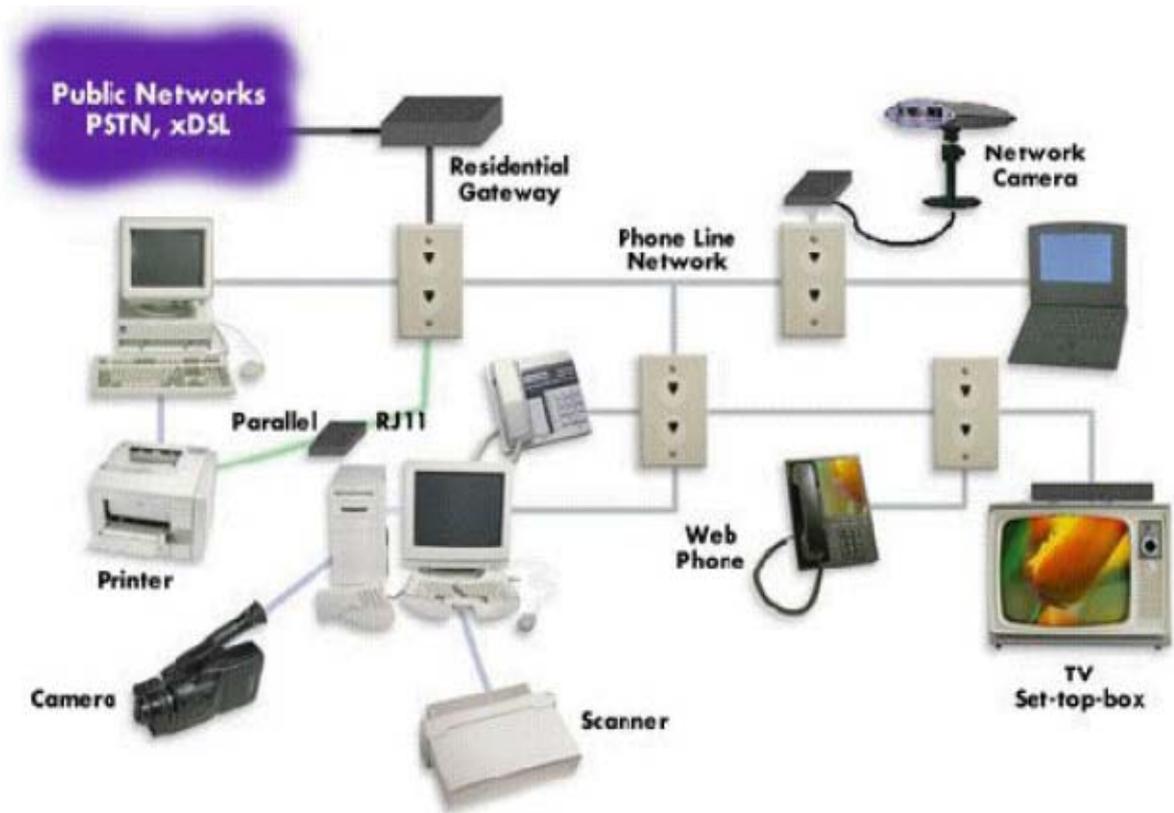


Figura 2.7.Red doméstica HomePNA

El objetivo de construir una red de área local de forma inmediata, con la ausencia de nuevas obras y sin ser necesaria la instalación de más cableado, surgió con la intención de unir ordenadores, impresoras y otros recursos como hubs, routers ADSL o pasarelas residenciales.

## 2.2.4 LonWorks

El protocolo de comunicación empleado, LonTalk, es un protocolo de comunicaciones basado en el modelo de referencia OSI de ISO. Este protocolo (LonTalk) es abierto (previo pago de tasas)

Cualquier dispositivo Lonworks, o nodo, está basado en un microcontrolador llamado Neuron Chip que actualmente fabrican Toshiba y Cypress. El diseño inicial del Neuron y el protocolo LonTalk fueron desarrollados por Echelon en el año 1990. Actualmente toda la información para implementar LonWorks en otro chip esta publicada en medios oficiales pero al estar la familia Neuron chips adaptada y dimensionada exclusivamente para este objetivo los fabricantes que eligen otras opciones son muy escasos (chips sobredimensionados encarecerán los equipos)

Del Neuron chip cabe destacar las características mostradas a continuación:

- ✓ Tiene un identificador único que permite direccionar cualquier nodo de forma unívoca dentro de una red. Con 48 bits de ancho, se graba en la memoria EEPROM durante la fabricación del circuito.
  
- ✓ Su modelo de comunicación es independiente del medio físico sobre el que funciona, por lo que los datos de información pueden circular sobre cualquiera de los que ya se ha analizado en apartados previos.
  
- ✓ Incluye un sistema operativo para ejecutar y planificar la aplicación distribuida y manejar las estructuras de datos intercambiadas en los nodos.

En la *figura 2.8* podemos observar la estructura del microcontrolador Neuron Chip, y a su vez podemos observar de manera más gráfica su modo de comunicación con el sistema a través de la red eléctrica:

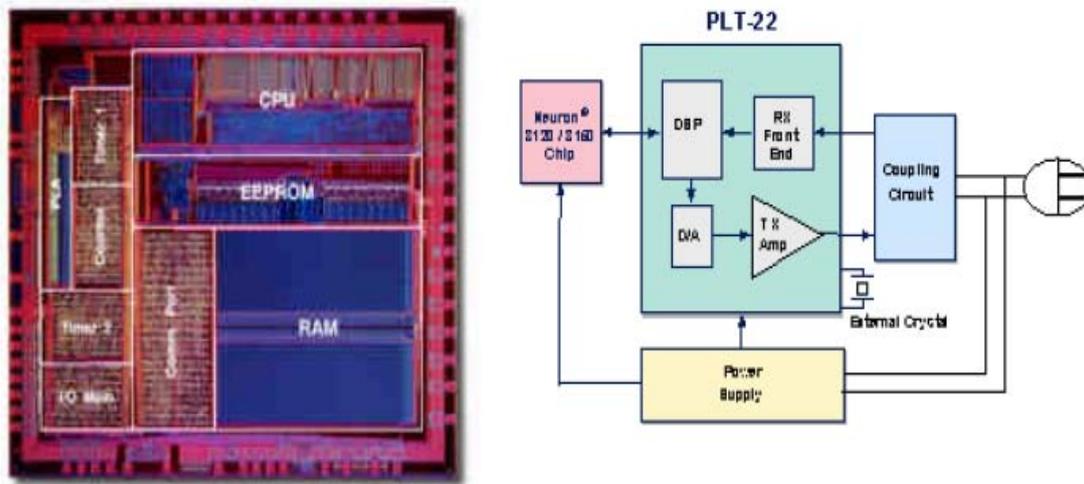


Figura 2.8. Estructura del Neuron Chip y Dispositivo Lonworks de comunicación

Estos circuitos se comunican entre sí enviándose telegramas que contienen la dirección de destino, información para el routing, datos de control así como los datos de la aplicación del usuario y un checksum como código detector de errores. Todos los intercambios de datos se inician en un Neuron Chip y se supervisan en el resto de los circuitos de la red. Un telegrama puede tener hasta 229 octetos de información neta para la aplicación distribuida.

Los datos pueden tener dos formatos, desde un mensaje explícito o una variable de red. Los mensajes explícitos son la forma más sencilla de intercambiar datos entre dos aplicaciones residentes en dos Neuron Chips del mismo segmento Lonworks. Por el contrario, las variables de red proporcionan un modelo estructurado para el intercambio automático de datos distribuidos en un segmento Lonworks. Aunque son menos flexibles que los mensajes explícitos, las variables de red evitan que el programador de la aplicación distribuida esté pendiente de los detalles de las comunicaciones.

El protocolo LonTalk incorpora una serie de características que se considera avanzadas, como servicio de autenticación, transmisiones prioritarias, detección de duplicados, identificación de envíos o diferentes tipos de direccionamiento. Por otro lado, cada subred es una agrupación lógica de un máximo de 127 nodos dentro de un dominio; en éste puede haber además hasta 255 subredes definidas.

## 2.2.5 BatiBus

BatiBUS es un protocolo de domotica totalmente abierto, es decir, que lo puede implementar cualquier empresa. Fue muy utilizado en los antiguos sistemas de control industrial franceses, y debido a sus limitaciones, quedó obsoleto. Desarrollado por Merlin Gerin, AIRELEC, EDF y Landis & Gyr fue el primer bus de control domótico que apareció en el mercado. En 1989, las compañías citadas fundaron el BatiBUS Club International (BCI) con el objetivo de extender y ampliar el uso del BatiBus. En la actualidad BCI está formado por gran cantidad de miembros de múltiples países, entre los cuales se encuentran ciertas compañías líderes en equipamiento de aire acondicionado, calefacción, iluminación, sistemas automáticos, etc.

A nivel de acceso, este protocolo usa la técnica CSMA-CA, (Carrier Sense Multiple Access with Collision Avoidance) similar a Ethernet pero con resolución positiva de las colisiones. Esto es, si dos dispositivos intentan acceder al mismo tiempo al bus ambos detectan que se está produciendo una colisión, pero sólo el que tiene más prioridad continua transmitiendo el otro deja de poner señal en el bus. Esta técnica es muy similar a la usada en el bus europeo EIB y también el en bus del sector del automóvil llamado CAN (Controller Area Network).

La filosofía es que todos los dispositivos BatiBUS escuchen lo que han enviado cualquier otro, todos procesan la información recibida, pero sólo aquellos que hayan sido programados para ello, filtrarán la trama y la subirán a la aplicación empotrada en cada dispositivo.

Es muy sencillo de instalar con una red de suministro de energía a todos los dispositivos y una topología totalmente abierta que es utilizada por todos los dispositivos y aplicaciones. Las principales características de BatiBUS son su facilidad de instalación, bajo coste y capacidad de evolución, ya que el protocolo permite añadir funciones conforme las necesidades lo exijan. La instalación de este cable se puede hacer en diversas topologías: bus, estrella, anillo, árbol o cualquier combinación de estas. Lo único que hay que respetar es no asignar direcciones idénticas a dos dispositivos de la misma instalación.

Todos los dispositivos BatiBUS disponen de unos micro-interruptores circulares o miniteclados que permiten asignar una dirección física y lógica que indentifican

unívocamente a cada dispositivo conectado al bus. Se puede implementar mediante el par trenzado similar al telefónico. Permite direccionar múltiples dispositivos, y soportar diversas aplicaciones utilizando un protocolo muy sencillo.

BatiBUS ha conseguido la certificación como estándar europeo CENELEC. Existen una serie de procedimientos y especificaciones que sirven para homologar cualquier producto que use esta tecnología como compatible con el resto de productos que cumplen este estándar. A su vez, la propia asociación BCI ha creado un conjunto de herramientas para facilitar el desarrollo de productos que cumplan esta especificación.

### **2.2.6 EHS**

El estándar EHS (European Home System) ha sido otro de los intentos que la industria europea (año 1984), auspiciada por la Comisión Europea, de crear una tecnología que permitiera la implantación de la domótica en el mercado residencial de forma masiva. El resultado fue la especificación del EHS en el año 1992. Está basada en una topología de niveles OSI (Open Standard Interconnection), y se especifican los niveles: físico, de enlace de datos, de red y de aplicación.

Tal y como fue pensado, el objetivo de la EHS es cubrir las necesidades de automatización de la mayoría de las viviendas europeas cuyos propietarios que no se pueden permitir el lujo de usar sistemas más potentes pero también más caros (como Lonworks, EIB o Batibus) debido a la mano de obra especializada que exige su instalación.

El EHS viene a cubrir, por prestaciones y objetivos, la parcela que tienen el CEbus norteamericano y el HBS japonés y rebasa las prestaciones del X-10 que tanta difusión ha conseguido en EEUU.

Los medios físicos que se pueden emplear son: red eléctrica (PL), par trenzado de clases 1 y 2 (TP1 y TP2), cable coaxial, radio frecuencia e infrarrojos.

Todos los medios pueden distribuir señales de clase 1 (señales de control), algunos distribuyen además señales de clase 2 (voz/datos baja velocidad) e incluso señales de clase 3 (audio/video/datos alta velocidad).

En EHS se pueden implementar tantas aplicaciones como dispositivos y funcionalidades se encuentren en un hogar. Cada dispositivo está asociado a una determinada área de aplicación, dentro de la cual el elemento es un objeto. Para definir cada objeto se utilizan dos bytes, uno para el área (application area), y otro para el dispositivo (device descriptor).

Existen diversas áreas de aplicación: telecomunicaciones, audio/video, electrodomésticos, calefacción, iluminación, etc.

Cada dispositivo EHS tiene asociada una subdirección única dentro del mismo segmento de red que además de identificar unívocamente a un nodo también lleva asociada información para el enrutado de los telegramas por diferentes segmentos de red EHS.

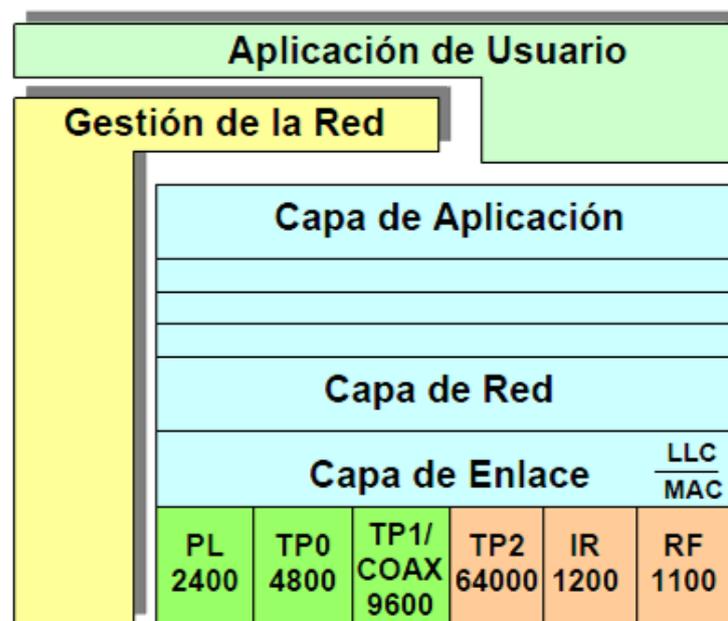


Figura 2.9. Capas del modelo OSI

### 2.2.7 European Installation Bus (EIB)

El Bus de Instalación Europeo EIB, estándar en que se centra este trabajo y que será analizado profundamente más adelante, es un sistema integrado de automatización y control de cualquier tipo de recinto, destinado a la aplicación de soluciones gradualmente compatibles, flexibles y rentables. Debido a su versatilidad funcional, su uso no se reduce a instalaciones simples y limitadas como pueden ser las de viviendas, sino que también proporciona soluciones para instalaciones comunes en el edificio completo.

Este estándar fue promovido alrededor de 1990 por el grupo de fabricantes que engloba la Asociación EIB (EIBA). Con sede en Bruselas, dicha entidad se dedica a la emisión de las marcas registradas relacionadas con el sistema, los modelos de comprobación y calidad de los productos y las actividades de marketing y estandarización. EIB también es distribuido bajo varias denominaciones diferentes, como Instabus Siemens (a la que responde el panel de instalación domótica a utilizar), Tebis o ABB I-Bus.

El EIB (European Installation Bus) es un sistema abierto bajo las mismas premisas que otros sistemas de comunicación. Existen tres posibles medios físicos para la interconexión de dispositivos: cable de par trenzado (con el cual trabajaremos en éste proyecto), red eléctrica de baja tensión y radio-frecuencia. La diferencia entre los dispositivos de los tres tipos radica en la electrónica de acceso al medio, siendo el resto del protocolo de comunicaciones común a todos ellos.

La instalación sobre red eléctrica de baja tensión, que funciona por corrientes portadoras de manera similar a otros sistemas, como X10, se reserva a viviendas o edificios ya construidos, donde la instalación de nuevo cableado sería muy costosa. No obstante, este tipo de medio es muy poco empleado por mayor coste y menor fiabilidad.

Éste es un sistema descentralizado (no requiere de un controlador central de la instalación), en el que todos los dispositivos que se conectan al bus de comunicación de dato tienen su propio microprocesador y electrónica de acceso al medio, goza de una arquitectura distribuida y puede llegar a incorporar a su estructura, en base a ciertas condiciones, más de 65.000 dispositivos repartidos en segmentos de línea, líneas y áreas.

En la *Tabla 2.2*, mostrada a continuación, podemos ver las formas más habituales de transmisión así como las áreas de aplicación de los soportes más frecuentemente utilizados para comunicaciones mediante el estándar EIB.

<i>Medio</i>	<i>Vía de transmisión</i>	<i>Áreas de aplicación</i>
<b>Par trenzado</b>	Bus autónomo de control	Nuevas instalaciones y grandes renovaciones
<b>Red eléctrica</b>	Red existente	Lugares sin necesidad de cableado adicional y con disponibilidad del tendido de 230 V
<b>Radiofrecuencia</b>	Radio	Lugares donde no se puede o desea tender cableado

*Tabla 2.2. Medios de transmisión y áreas de aplicación preferidas para el estándar EIB*

Cabe señalar que en una red EIB, de par trenzado, podemos encontrar tres elementos diferentes: bus, sensores y actuadores.

El bus es el medio físico que une a todos los elementos del sistema, es un cable bifilar del tipo PY-CYM 2 x 2 x 0.8 de sección 0,5 mm<sup>2</sup> y una resistencia de bucle de 72  $\Omega$ /Km (o cualquier otro del tipo de cable de comunicación con la misma sección y que sea trenzado y apantallado). Este cable conductor puede tenderse por el mismo recorrido que la instalación eléctrica, de la misma forma pueden empalmarse y derivarse. Obviamente si la instalación eléctrica es apantallada se garantiza aún más la inmunidad a interferencias debido a la red eléctrica.

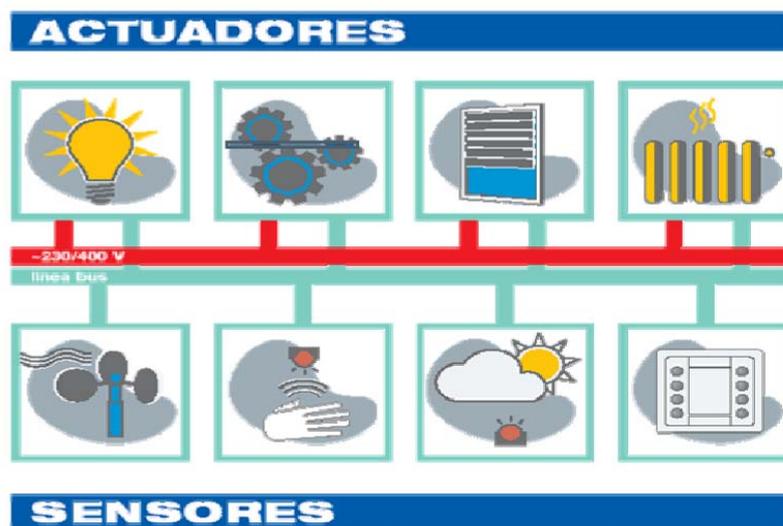
Los sensores son los elementos encargados de pasar parámetros físicos (temperatura, humedad, movimiento, etc.) o actuaciones sobre pulsadores e interruptores

en una señal, llamada en este argot, telegrama, que es interpretada por los elementos a los que va dirigida para producir una actuación.

Los actuadores son los que reciben las órdenes en forma de telegramas de los sensores y las ejecutan, bien abriendo, cerrando, temporizando, regulando otros sistemas (actuando directamente sobre los circuitos de potencia a controlar) o visualizando las órdenes recibidas.

Los dispositivos se fabrican para ser colocados sobre riel DIN estándar de 35x7.5, para montaje superficial y para embutir. El DIN tiene las pistas del bus incorporadas, por lo tanto no hay que cablear desde un dispositivo hasta el bus, ya que este tiene en la parte trasera dos placas que al colocarlo sobre el riel hacen contacto con el bus. La parte que pueda quedar libre del riel DIN se puede cubrir con unas tapas. En las cajas de 10x5 se pueden colocar teclas convencionales y una interface para Instabus. De esta manera no se limita la línea de llaves y tomas que se pueden instalar.

Para finalizar este punto, en la *figura 2.10* se representa de forma esquemática un ejemplo de lo que podría ser una estructura básica de red EIB:



*Figura 2.10. Ejemplo básico de red EIB*

## 2.2.8 KONNEX/KNX

El Konnex es la iniciativa de tres asociaciones europeas EIBA, (European Installation Bus Association), Batibus Club International, EHSA (European Home Systems Association), con el objeto de crear un único estándar europeo para la automatización de las viviendas y oficinas [12].

Los objetivos de esta iniciativa, con el nombre de "Convergencia", son:

- ✓ Crear un único estándar para la domótica e inmótica que cubra todas las necesidades y requisitos de las instalaciones profesionales y residenciales de ámbito europeo.
- ✓ Aumentar la presencia de estos buses domóticos en áreas como la climatización o HVAC.
- ✓ Mejorar las prestaciones de los diversos medios físicos de comunicación sobre todo en la tecnología de radiofrecuencia.
- ✓ Introducir nuevos modos de funcionamiento que permitan aplicar una filosofía Plug&Play a muchos de dispositivos típicos de una vivienda.
- ✓ Contactar con empresas proveedoras de servicios como las telecom y las eléctricas con el objeto de potenciar las instalaciones de telegestión técnica de las viviendas o domótica.

En resumen, se trata de, partiendo de los sistemas EIB, EHS y Batibus, crear un único estándar europeo que sea capaz de competir en calidad, prestaciones y precios con otros sistemas norteamericanos como el Lonworks o CEBus.

Gracias a la flexibilidad de la tecnología KNX, cualquier instalación puede ser fácilmente adaptable a las necesidades cambiantes del usuario. Asimismo, utilizando las

pasarelas correspondientes, es posible la transmisión de telegramas EIB/KNX a otros medios por ejemplo fibra óptica

El estándar KNX permite a cada fabricante una elección libre entre el modo de configuración y el medio de comunicación para el desarrollo de un dispositivo en una aplicación del sistema. KNX incorpora tres modos de configuración distintos:

- El "E-mode" (Modo Fácil  $\rightarrow$  Easy mode)

Este mecanismo de configuración está destinado a instaladores con una formación básica y provee una rápida evolución del aprendizaje pero con funciones limitadas, comparado con el "S- mode". Los componentes "E-mode" están ya preprogramados y cargados con una serie de parámetros. Con un configurador simple, cada componente puede ser en parte reconfigurado, principalmente los parámetros de configuración y los enlaces de comunicación. La AsociaciónKonnex ofrece un configurador independiente del fabricante llamado "ETS 3 Starter" que permite planificar las instalaciones, configurar y unir productos especiales seleccionados certificados KNX en las instalaciones con funcionalidades estándar.

- El "A-mode" (Modo Automático  $\rightarrow$  Automatic mode)

Este mecanismo de configuración está desarrollado especialmente para aplicaciones de usuario final, por ejemplo, dispositivos domésticos o pequeño material de instalación, vendido a través de los canales de venta al gran público. Los componentes "A-mode" disponen de mecanismos de configuración automática, que adaptan sus enlaces de comunicación al resto de componentes "A-mode" en la red. Cada componente contiene un número determinado de parámetros fijados y una librería con instrucciones de cómo comunicar con otros componentes "A-mode". Gracias al ETS 3 Starter algunos productos certificados KNX pueden ser utilizados en más de un modo de configuración. Por ejemplo en el S-mode con el "ETS 3 Professional" y en el "E-mode" con el "ETS 3 Starter".

Futuras versiones de ETS harán posible el enlace de todos los productos certificados KNX instalados en una instalación sin importar su modo de configuración.

- El "S-mode" (Modo Sistema  $\rightarrow$  System mode)

Este mecanismo de configuración está enfocado a instaladores preparados para realizar funciones de control sofisticadas en edificios. Todos los componentes "S-mode" en una instalación serán diseccionados por una herramienta de software común (ETS), basado en una base de datos de productos suministradas de los productos para su planificación, configuración y unión lógica. La configuración "S-mode" ofrece el más alto grado de flexibilidad en funcionalidad así como en links de comunicación.

Éste estándar nos presenta un gran abanico de ventajas, entre las cuales debemos destacar:

- Independencia de cualquier tecnología tanto de hardware como de software

La tecnología KNX se ha convertido a nivel mundial en el primer estándar abierto libre de regalías e independiente de la plataforma hardware para Sistemas de

- Control de Viviendas y Edificios.

Asegura que los productos de distintos fabricantes utilizados en distintas aplicaciones funcionarán y se comunicarán unas con otras. Esto permite un alto grado de flexibilidad en la ampliación y modificación de las instalaciones.

- Calidad del producto

La Asociación Konnex requiere un alto nivel de control de calidad durante todas las etapas de la vida del producto. Por esta razón, todos los miembros de Konnex que desarrollan productos KNX bajo la marca KNX, tienen que cumplir con la ISO 9001 antes que puedan solicitar la certificación de productos KNX.

Adicionalmente a la ISO 9001, los productos deben cumplir con los requerimientos de la norma Europea para Sistemas Electrónicos en Viviendas y Edificios, es decir, EN 50090-2-2. En caso de duda, la Asociación Konnex está autorizada a testear de nuevo los productos para su certificación o bien puede requerir a los fabricantes un declaración de conformidad del hardware.

- Funcionalidades independientes de los fabricantes

El estándar KNX contiene distintos perfiles de aplicación para diversas aplicaciones comunes en Viviendas y Edificios. Bajo la supervisión del Grupo Técnico varios grupos de trabajo de especificación de las aplicaciones realizan propuestas para estandarizar diversas funcionalidades (inputs, outputs, diagnóstico de datos y parámetros) en el dominio específico de aplicación. Para asegurar un alto grado de disciplina cruzada e interoperabilidad multivendedor, el Task Force Interworking reevalúa estas propuestas antes de que se tome la decisión de incorporar un perfil de aplicación en el estándar KNX.

La Asociación Konnex pone a disposición de todos, una herramienta de software para la ingeniería independiente del fabricante para planificar las uniones lógicas y configurar los productos certificados KNX.

## Capítulo 3

### Sistema KNX

*“No se puede desatar un nudo sin saber cómo está hecho”*

*Aristóteles 384 AC-322 AC. Filósofo griego.*

A día de hoy tanto si hablamos de una casa como de un complejo de oficinas, la demanda del mercado respecto al confort y la funcionalidad en la gestión del aire acondicionado, iluminación y sistemas de control de accesos va creciendo de forma considerable, al mismo tiempo que la sociedad va siendo más consciente de la necesidad de un uso eficiente de la energía. Nos encontramos pues en una situación en la que son requeridas mayores comodidades y seguridad con un menor consumo energético, esto sólo puede ser mejorado con un control inteligente y la supervisión de todos los productos involucrados. Cómo podemos imaginar todo esto requiere un mayor cableado, que vaya desde los sensores y actuadores hasta las centrales de supervisión y control. Tales masas de cableado implican un mayor esfuerzo en el diseño y en la instalación, incrementando el riesgo potencial de fuego, así como disparando los precios.

Para transferir datos de control de todos los componentes de gestión de edificios se requiere un sistema que elimine los problemas que presentan los dispositivos aislados,

asegurando que todos los componentes se comuniquen a través de un lenguaje común. En resumen, se necesita un sistema tal que sea independiente del fabricante y de los dominios de aplicación. La solución que nos aporta el mercado actual a este problema es el estándar KNX, basado en diferentes estándares con más de 20 años de experiencia [12].

A través del medio de transmisión (par trenzado, radio frecuencia, línea de fuerza o IP/Ethernet) de KNX, sobre el que se conectan todos los dispositivos, se intercambia la comunicación. Los dispositivos conectados al bus, tanto sensores como actuadores, son utilizados para el control de equipamiento de gestión de edificios en todas las aplicaciones posibles: iluminación, persianas / contraventanas, sistemas de seguridad, gestión energética, calefacción, sistemas de ventilación y aire acondicionado, sistemas de supervisión y señalización, interfaces a servicios y sistemas de control de edificios, control remoto, medición, audio / video, control de bienes de gama blanca, etc. Todas estas funciones pueden ser controladas, supervisadas y señalizadas utilizando un sistema uniforme sin la necesidad de centros de control adicionales.

A lo largo de éste capítulo veremos la estructura interna de éste estándar y su funcionamiento, logrando, de este modo, alcanzar los conocimientos necesarios para la realización de la programación de los diferentes componentes que integran el sistema a tratar en este proyecto.

### **3.1 Topología del sistema**

Debido a que el EIB debe resultar económicamente rentable desde el sistema más pequeño hasta las soluciones más complejas implementadas en edificios funcionales tienen una estructura jerárquica.

La conexión de los componentes al bus se puede realizar en cualquier topología (estrella, árbol o línea), lo que facilita la instalación en edificios y viviendas. La única opción que no está permitida es cerrar la instalación, es decir, no se puede montar una instalación en anillo. Este sistema no necesita resistencias de cierre.

La topología del sistema se realiza en tres niveles de conexión: línea, área y componente.

### 3.1.1 Línea y Área

La línea forma la unidad de instalación más pequeña, consta de un máximo de cuatro segmentos de línea, cada uno de los cuales puede disponer de 64 componentes bus y debe ser alimentado por una fuente adecuada que está separada galvánicamente del resto, lo que supone que si se produce un fallo en la línea el resto seguirá funcionando normalmente.

El número real de componentes conectados depende de la fuente y del consumo de cada aparato.

A la hora de realizar el diseño debemos tener en cuenta una serie de limitaciones referentes a las longitudes de las líneas y las normas de actuación respecto a la topología, estas limitaciones están mostradas en la siguiente tabla:

<b>Longitud total de todos los cables tendidos en una línea</b>	<b>&lt; 1000 metros</b>
<b>Distancia máxima de línea entre dos componentes</b>	<b>&lt; 700 metros</b>
<b>Distancia máxima entre dos fuentes de alimentación y cada componente</b>	<b>&lt; 350 metros</b>
<b>Longitud máxima de línea entre dos fuentes de alimentación</b>	<b>200 metros</b>

*Tabla 3.1. Limitaciones de las longitudes del cableado*

Si se desea emplear más de una línea este sistema nos permite conectar hasta 15 líneas a una línea principal mediante el uso de los llamados Acopladores de Líneas (AL), formando lo que se denomina áreas. También es posible la ubicación de hasta 64 componentes en la línea principal aunque éstos disminuyen con el número de acopladores de línea en uso.

Si fuera necesaria la conexión de más de 64 aparatos en una línea y no fuera posible mantener las longitudes de línea se puede hacer uso de repetidores (REP) en la línea. El repetidor forma un nuevo segmento de línea, que necesitará disponer de una fuente de alimentación propia. Es posible el incremento de hasta un máximo de tres segmentos por línea, estos tres repetidores deben conectarse en paralelo tras el segmento de línea principal sin poder ser colocados en la línea principal de área.

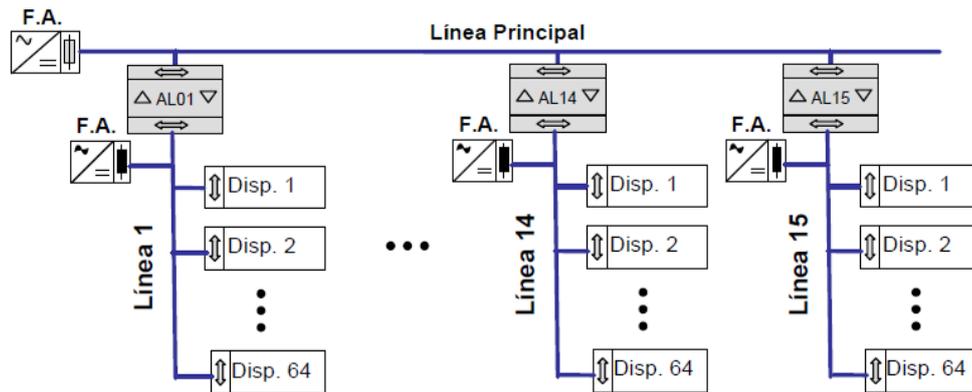


Figura 3.1. Configuración de un área

Mediante la utilización de los llamados Acopladores de Área (AA) es posible conectar hasta 15 áreas por medio de una línea de áreas, además el uso de interfaces adecuados permite conectarse a otros sistemas de automatización de edificios.

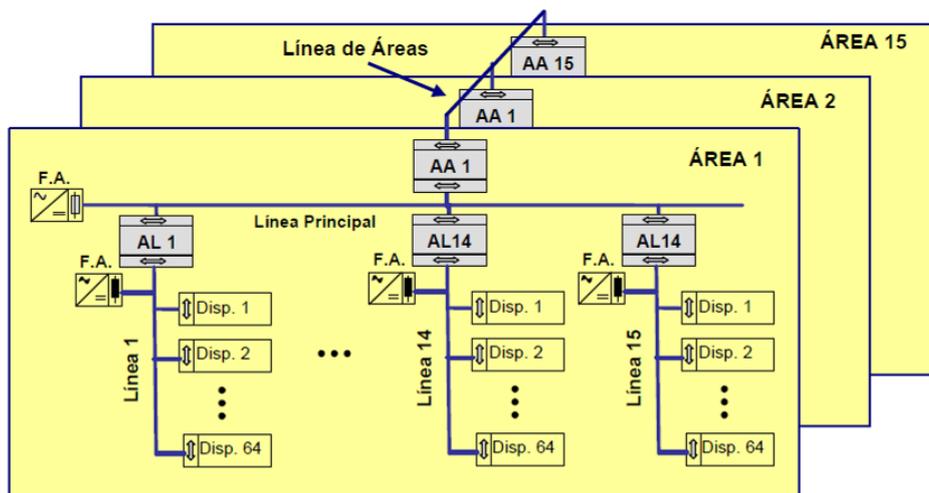


Figura 3.2. Interconexión de áreas

### 3.1.2 Direccionamiento

Los diferentes elementos existentes en una instalación EIB quedan perfectamente identificados gracias al sistema de direccionamiento. Existen dos tipos de direcciones: direcciones físicas y direcciones de grupo.

### 3.1.3 Dirección física

La dirección física sirve para identificar de forma unívoca el componente, describiendo su localización dentro de la topología.

La dirección física consta de tres campos, con el siguiente formato:

- ✓ Área: Identifica una de las 15 zonas (4 bits).

✓ Línea: Identifica una de las 12 líneas (4 bits).

✓ Componente: Identifica uno de los 64 posibles componentes de una línea (8 bits).

Así por ejemplo, la dirección física 1.1.64, indica el componente 64 de la línea 1 y de la zona 1.

La dirección física solo se asigna una vez a cada componente bus. Estos reciben su dirección física a través del software de programación ETS3, cuando se actúa sobre el botón de programación de cada componente en el momento en que este se programa.

La línea de zonas y la línea principal tienen como dirección física la 0 y la dirección de un acoplador de bus desprogramado es 15.15.255

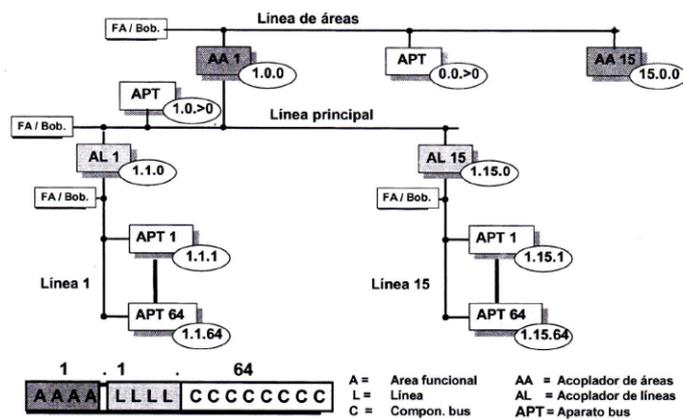


Figura 3.3. Ejemplo de asignación de direcciones

Si observamos la imagen anterior podemos ver que:

A= 1-15 señala las áreas funcionales 1-15

A= 0 señala los componentes en la línea de áreas

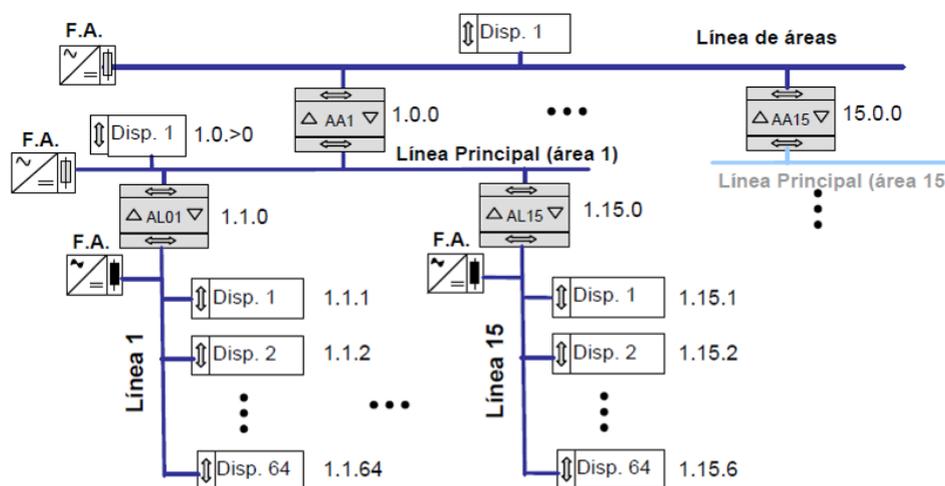
L= 1-15 señala las líneas 1-15 en las áreas definidas por A

L= 0 señala la línea principal

C= 1-255 señala los componentes en la línea definida por L

C= 0 señala el acoplador de línea o de área

En la *figura 3.4* se muestra un ejemplo de direcciones físicas asignadas a los dispositivos de un sistema EIB:



*Figura 3.4. Ejemplo de direccionamiento físico*

En la línea de áreas se conectan hasta 15 acopladores de área (AA), cuyas direcciones irán desde 1.0.0 hasta 15.0.0. Esta línea puede tener conectados dispositivos normales (direcciones 0.0.>0).

Cada área tiene una línea principal, con su fuente de alimentación, a la que se conectan los acopladores de línea (AL), con direcciones 1.1.0 a 15.0.0, y a cada línea secundaria conectada a un acoplador de línea pueden conectarse hasta 64 dispositivos.

Para la interconexión de diferentes líneas y diferentes áreas se emplea la unidad de acoplamiento. Este elemento es el mismo para los diferentes tipos de conexión, y dependiendo de la dirección física que se le asigne actuará como acoplador de línea, acoplador de área, o incluso repetidor dentro de una misma línea.

En el caso del acoplador de línea o de área, la unidad de acoplamiento actúa como encaminador (router), y mantiene una tabla interna de direcciones de las subredes que conecta para aislar el tráfico entre ellas.

### 3.1.4 Dirección de grupo

Las direcciones de grupo se emplean para definir funciones específicas del sistema, y son las que determinan las asociaciones de dispositivos en funcionamiento (y la comunicación entre sus objetos de aplicación).

Las direcciones de grupo asignan la correspondencia entre elementos de entrada al sistema (sensores) y elementos de salida (actuadores). Se pueden utilizar dos tipos de direccionamiento de grupo: de dos y tres niveles, dependiendo de las necesidades en la jerarquización de las funciones del sistema.

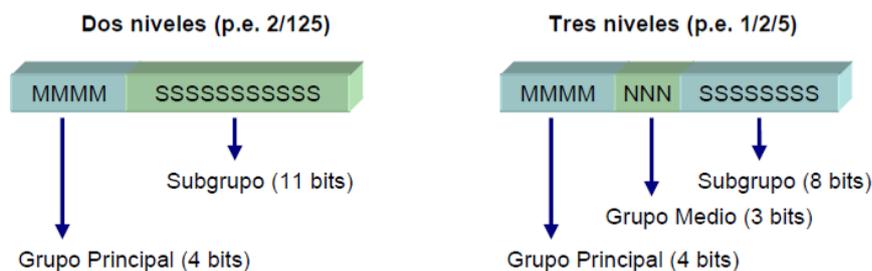


Figura3.5. Niveles en las direcciones de

Habitualmente el campo de grupo principal se utiliza para englobar grupos de funciones (alarmas, iluminación, control de persianas, etc.). Se pueden emplear valores de 1 a 13, los valores 14 y 15 no deben emplearse, ya que no son filtrados por los acopladores y podrían afectar a la dinámica de funcionamiento de todo el sistema. En todos los campos la dirección 0 está reservada para funciones del sistema.

En la configuración de una instalación EIB, la asignación de direcciones de grupo es básica para asegurar su correcto funcionamiento. Las direcciones de grupo, que asocian sensores con actuadores, se pueden asignar a cualquier dispositivo en cualquier línea (son independientes de las direcciones físicas), con las siguientes condiciones:

- Los sensores sólo pueden enviar una dirección de grupo (sólo se les puede asociar una dirección de grupo).
- Varios actuadores pueden tener la misma dirección de grupo, es decir, responden a un mismo mensaje o telegrama.
- Los actuadores pueden responder a más de una dirección de grupo (pueden estar direccionados o asociados a varios sensores simultáneamente).

La *figura 3.6* ilustra un ejemplo sencillo de asociación de elementos en una instalación EIB. En él se dispone de nueve componentes distribuidos en dos salas, y cableados en la misma línea de bus (una sola fuente de alimentación). Los pulsadores P1 y P2 se emplean para encender y apagar simultáneamente todas las luces de sus respectivas salas, y el sensor crepuscular S para apagar las más próximas a las ventanas cuando entra luz del exterior.

Para realizar la asignación de direcciones físicas deberá decidirse en qué área y línea vamos a trabajar. En este caso supondremos que los elementos están en el área 1, línea 1, por lo que las direcciones físicas se asignarán arbitrariamente como 1.1.X (siendo X el número de dispositivo).

*Funcionamiento:* El pulsador P1 activa / desactiva las cargas conectadas a los actuadores L11, L12 y L13. El pulsador P2 actúa sobre L21, L22 y L23. El sensor crepuscular S actúa sobre las bombillas más próximas a las ventanas (L11 y L21) para encender o apagar cuando anochece.

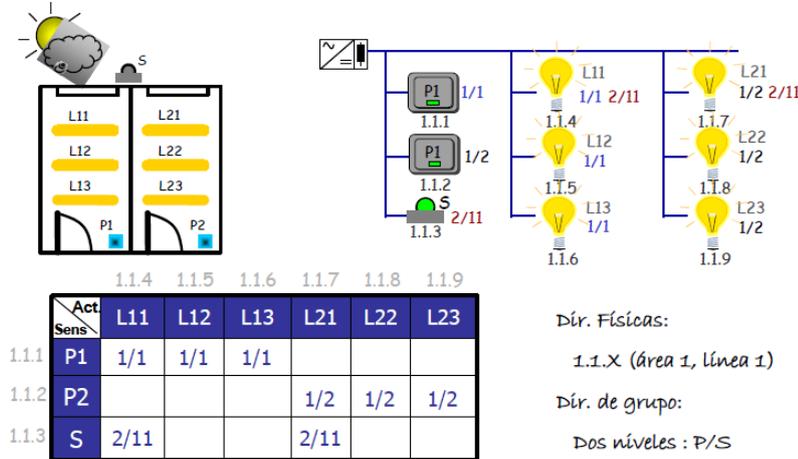


Figura 3.6. Asignación de direcciones de grupo

Para realizar las asociaciones sensores-actuadores, será necesario asignar las direcciones de grupo a los componentes. En este caso emplearemos direcciones de dos niveles con la nomenclatura P/S, siendo P el grupo principal (valores de 1 a 13) y S el grupo secundario (puede tomar valores de 1 a 2047). La asignación, en este caso se realiza también a criterio del diseñador, teniendo en cuenta las restricciones descritas en este capítulo.

De este modo, se comienza asignando una dirección de grupo única a cada sensor: P1 se asocia a 1/1, de manera que cuando el usuario pulse la tecla, se enviará por el bus un telegrama que contendrá, entre otros campos, la dirección de grupo 1/1. Dicha dirección de grupo se asociará también a los actuadores L11, L12 y L13, de forma que cuando escuchen el telegrama con esa dirección, se activarán simultáneamente.

El mismo proceso se sigue para P2, al que enviará la dirección 1/2, que se asocia también a L21, L22 y L23.

Por último, el sensor crepuscular S se programa para enviar la dirección 2/11, a la que responden los actuadores L11 y L21 . [10]

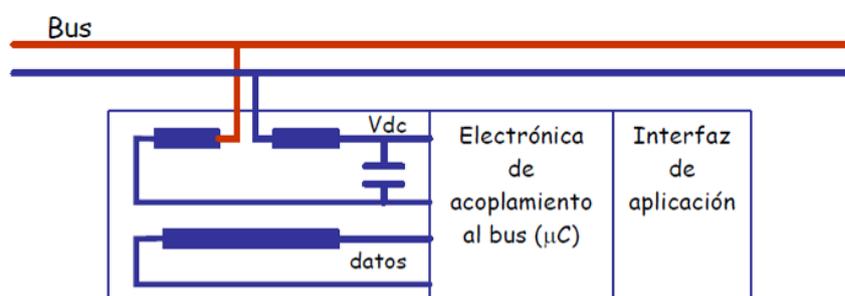
### 3.1.5 Técnica de transmisión

A la hora de realizar la transmisión de los datos se utilizará, de los medios físicos anteriormente nombrados, el de par trenzado.

Los datos se transmiten como una tensión alterna superpuesta sobre la alimentación en corriente continua del bus, empleando para ello únicamente dos hilos. Para ello es necesario, por una parte, aislar la fuente de alimentación de los datos, para que ésta no suponga una carga sobre ellos, y por otra, desacoplar los datos de la componente de alimentación continua en cada dispositivo.

Los dispositivos conectados al bus (*Figura 3.7*) disponen de un transformador para separar la componente continua de alimentación, de la componente alterna que representa los datos.

En el primario, la componente inductiva ( $X_L=2\cdot\pi\cdot f\cdot L$ ) resulta muy baja para la componente continua ( $V_{dc}$ ), y la componente capacitiva  $X_C$  es muy alta, por lo que en  $C$  tenemos la tensión de alimentación continua. Para la tensión alterna de datos ( $V_{ac}$ ), de alta frecuencia, la  $X_L$  presenta un valor importante, y la  $X_C$  es muy baja, por lo que los datos resultan filtrados, eliminándolos de la tensión de alimentación del módulo.

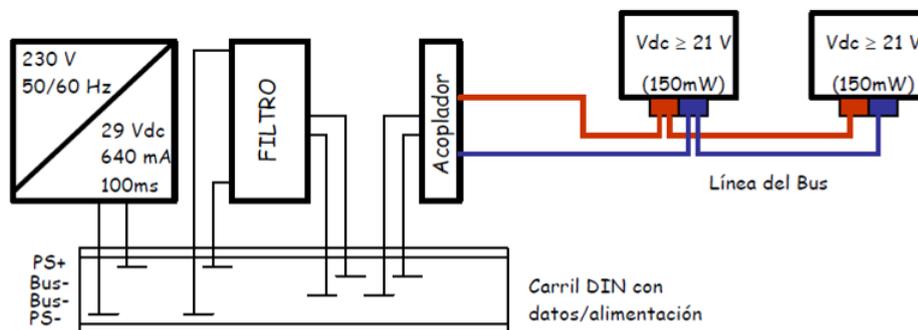


*Figura 3.7. Desacoplo de alimentación /datos*

Cuando un dispositivo envía datos, la electrónica de acoplamiento los emplea para excitar el secundario del transformador, de modo que se inducen al primario y se superponen a la tensión continua de alimentación Vdc.

Para la recepción, los datos representan la corriente alterna en el primario, que se inducen al secundario y son así separados de la tensión continua.

Cada línea tiene su propia fuente de alimentación que suministra la tensión a todos los dispositivos conectados (*Figura 3.8*). La fuente dispone de control integrado de corriente y tensión y salva microcortes de hasta 100  $\mu$ s. La tensión nominal de alimentación es de 29V, y cada dispositivo requiere un mínimo de 21V para mantenerse en zona de operación segura (SOA), y supone una carga típica de 150mW en el bus (en caso de carga adicional, hasta 200mW). De este modo se aseguran unos márgenes de tensión y consumo que garanticen un funcionamiento adecuado incluso utilizando el máximo número de dispositivos posible en la instalación.



*Figura 3.8. Conexión de alimentación y dispositivos al bus*

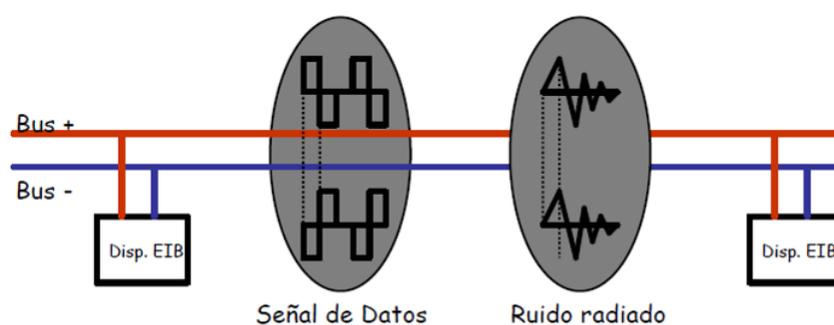
La conexión de la fuente de alimentación al bus se realiza a través de una bobina de filtro, de modo que la etapa de filtrado de alimentación suponga una carga despreciable sobre la componente de datos y no los interfiera (la bobina tiene una  $X_L \sim 0$  en continua, y una  $X_L$  elevada para la componente de alterna que representa los datos).

El medio físico empleado en la red es un cable de par trenzado (simétrico, de sección 0.8 mm<sup>2</sup> e impedancia característica  $Z_0=72\Omega$ ).

Los datos se transmiten en modo simétrico sobre este par de conductores (no se ponen a tierra). El empleo de transmisión diferencial, junto con la simetría de los conductores, garantiza que el ruido afectará por igual a los conductores, de modo que la diferencia de tensiones permanece invariante (*Figura 3.9*). Esta es una técnica empleada en la mayoría de las redes de comunicación de datos.

La inmunidad al ruido mejora por la baja resistencia del enlace de los dispositivos mediante acoplamiento aislado (transformador). La transmisión de datos se realiza en modo asíncrono, a una velocidad de 9600bps. Los datos se codifican en modo simétrico, como se ha descrito, correspondiendo a un 1 lógico la ausencia de impulso, y a un 0 lógico la presencia de un impulso simétrico. Así, los 0's representan un impulso negativo-positivo de -5V a +5V (*Figura 3.10*).

Para conseguir la simetría en la transmisión, cada dispositivo produce tan solo la onda negativa por absorción de corriente del bus, y es la bobina de acoplamiento de la fuente de alimentación conectada a esa línea la que genera una fuerza contraelectromotriz responsable de la generación de la semionda positiva. Por ello la onda real obtenida no es perfectamente simétrica (*Figura 3.10*), aunque sí muy aproximada.



*Figura 3.9. Detalle de transmisión simétrica y cancelación del ruido en la transmisión*

Por esta razón, existen limitaciones en cuanto a la distancia máxima entre un componente y la fuente de alimentación del bus, que interviene de modo pasivo en la codificación de los datos.

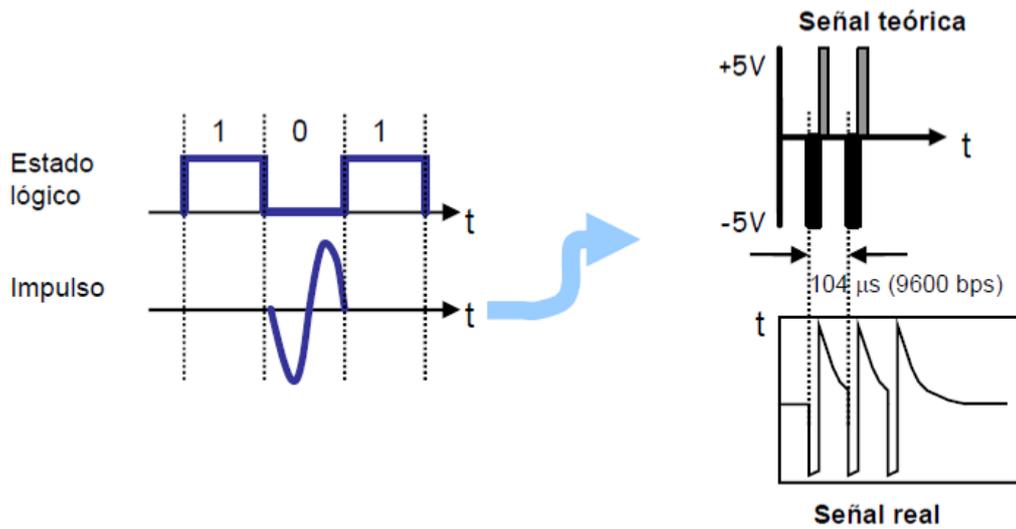


Figura 3.10. Generación de corriente portadora sobre tensión de alimentación

## 3.2 Acoplador

Como ya hemos visto el acoplador es un aparato que nos permite la unión entre líneas y áreas, así pues se puede utilizar de diversos modos, como son:

Acoplador de áreas AA: Conecta la línea de área a la línea principal

Acoplador de líneas AL: Conecta la línea principal a una línea secundaria

Amplificador (Repetidor) de línea AML: Permite ampliar una línea con un segmento más de un máximo de 64 componentes y una longitud de cable adicional de 1000 metros.

Los acopladores de línea y de área sólo dejan pasar telegramas de cruce de líneas, mientras que un amplificador de línea deja pasar todos los telegramas.

La dirección física asignada es la que define si un acoplador es un AA, AL o AML, por ejemplo, la dirección 1.1.0 definiría al acoplador como AL para la línea 1 en el área 1. En la siguiente figura podemos ver esto de una forma más clara.

Dirección física			El acoplador se utiliza	
F	L	A	como	para
> 0	= 0	= 0	Acoplador de áreas	Línea de áreas/Línea principal
> 0	> 0	= 0	Acoplador de líneas	Línea principal/Línea secundaria
> 0	> 0	> 0	Amplificador de línea	Expansión de una línea

Figura 3.11. Asignación de direcciones físicas

Cuando se asignan los parámetros, se le proporciona al acoplador una tabla de filtros, de tal manera que los telegramas que reciba serán filtrados permitiendo el paso únicamente a aquellos que se encuentren en su tabla. Con esto conseguimos que cada línea funcione de forma independiente.

En el caso de un repetidor dejaría pasar todos los telegramas puesto que no posee tabla de filtros.

### 3.3 Componentes

Al margen de los elementos auxiliares para posibilitar el funcionamiento de un sistema EIB, como son la fuente de alimentación, filtros y cables, los elementos más importantes en la instalación son los dispositivos dotados de una cierta 'inteligencia'.

Al tratarse de un sistema distribuido, las funciones a realizar se encuentran programadas en forma de objetos de aplicación en los sensores y actuadores que intercambian información, posibilitando así la realización de las acciones de control. Estos dispositivos constan de tres partes básicas:

- Unidad de Acoplamiento al bus (AB o BCU)
- Módulo de Aplicación (MA)
- Programa de Aplicación (PA)

El acoplador al bus (AB o BCU) es un aparato universal, que contiene la electrónica necesaria para gestionar el enlace: envío y recepción de telegramas, ejecución de los objetos de aplicación, filtrado de direcciones físicas y de grupo para reconocer los telegramas destinados al dispositivo, comprobación de errores, envío de reconocimientos, etc. El acoplador examina cíclicamente la interfaz de aplicación para detectar cambios de señal. Esta unidad de acoplamiento consta de dos partes [10]:

- Un módulo de transmisión (MT), que realiza las siguientes funciones:

- ✓ Desacoplo de la alimentación y datos (acoplamiento por transformador y filtro capacitivo).
  - ✓ Protección contra inversión de polaridad.
  - ✓ Generación de la tensión de alimentación estabilizada a 24Vdc.
  - ✓ Inicialización del volcado de seguridad de la memoria RAM si la tensión del bus cae por debajo de 18 V.
  - ✓ Generación del reset del microprocesador si la tensión del bus cae por debajo de 5 V.
  - ✓ Amplificación y funciones lógicas para la recepción – transmisión desde el bus.
  - ✓ Vigilancia de la temperatura de la unidad.
- El controlador de acoplamiento al bus(BCC) que incluye:
    - ✓ Memoria ROM permanente, que contiene el software del sistema (el ‘sistema operativo’ de la BCU).
    - ✓ Memoria RAM volátil, que contiene datos durante la operación normal del dispositivo.
    - ✓ Memoria no volátil borrable eléctricamente (EEPROM), donde se almacenan el programa de aplicación, la dirección física y la tabla de direcciones de grupo.

Los programas de aplicación se encuentran en una base de datos que proporciona cada fabricante, y pueden ser descargados a las BCU a través del bus utilizando el software adecuado.

El módulo de aplicación y el acoplador al bus se encuentran en el mercado, bien juntos, en una sola unidad, o separados, dependiendo del producto concreto. En cualquier caso, ambos deben ser del mismo fabricante.

Si se adquieren por separado, el acoplador al bus y el módulo de aplicación se unen mediante un conector estandarizado para el KNX denominado IFE (Interfaz Físico Externo). Estos IFE, con 10 o 12 pines proporcionan:

- Un interface para intercambiar información entre ambas partes
- La alimentación del módulo de aplicación

Cuando la BCU es una parte separable del resto del dispositivo bus, podemos encontrárnosla para montaje empotrado, sobre pared, sobre falso techo o en carril DIN

Si la unidad de acoplamiento al bus está integrada junto al resto del aparato viene montada en el mismo mediante el módulo BIM (Bus interface module) o a través de un chipset. Un módulo BIN es una BCU sin conexión posterior y con una serie de componentes adicionales un chipset consiste en el núcleo de un BIM

Todos los componentes del bus disponen, gracias al BCU integrado de su propia “inteligencia”, por ello el sistema KNX funciona de forma descentralizada y no necesita ninguna unidad central de control, sin embargo, funciones centrales pueden necesitar realizarse mediante software instalado en un PC.

Los componentes pueden dividirse fundamentalmente en tres grupos:

- **Sensores:** La unidad de aplicación proporciona información a la BCU, que la codifica y envía inmediatamente, para esto, el acoplador al bus realiza comprobaciones a intervalos regulares sobre el estado de la unidad.

- **Actuadores:** El acoplador al bus recibe telegramas del bus de instalación, los descodifica y manda esta información al módulo de aplicación.
- **Controladores:** Afectan al intercambio de datos entre sensores y actuadores.

## 3.4 Comunicación

La información que circula por el bus es intercambiada entre los componentes individuales en forma de telegramas. En términos de velocidad de transmisión, generación y recepción de pulsos, la tecnología de transmisión EIB no requiere resistencias de terminación. La información se transmite de forma simétrica en el bus, es decir, como una diferencia de potencial entre los dos hilos y no referida a tierra, de este modo, las interferencias o ruido no influyen en la transmisión de la información.

### 3.4.1 Acceso al bus

En este sistema en acceso al bus se organiza mediante el envío de paquetes en serie, lo que implica que en el bus sólo puede haber información proveniente de un único dispositivo en cada momento. El sistema es descentralizado, de éste modo cada componente decide cuando accede al bus. Para evitar conflictos en el caso de que dos de estos componentes quisieran acceder al mismo tiempo el método de acceso al medio empleado es de tipo CSMA/CA (Carrier sense multiple access / Collision avoidance. Acceso múltiple por detección de portadora, evitando colisiones). La codificación se realiza de modo que el estado lógico '0' es dominante (impulso simétrico) sobre el '1', que se denomina recesivo (no hay impulso). El mecanismo de resolución de colisiones es el siguiente [10]:

- El dispositivo comprueba el bus, y si está libre comienza la transmisión.
- Durante el envío cada dispositivo escucha los datos presentes en el bus, comparándolos en todo momento con los que ha transmitido.
- Si no se producen colisiones, el envío se completa sin contratiempos.
- Si, por el contrario, se produce una colisión con los datos enviados por otro equipo, el arbitraje se resuelve por prioridad de los bits dominantes sobre los recesivos.

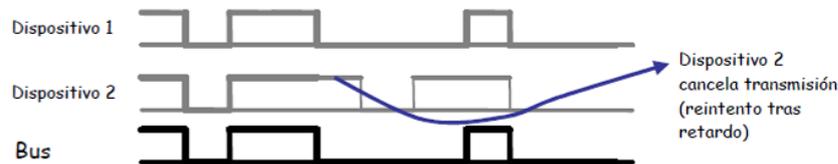


Figura 3.12. Resolución de colisiones CSMA/CA en EIB

Por lo tanto, tendrán mayor prioridad aquellas tramas que presente un mayor número de ceros en su inicio.

### 3.4.2 Esquema de un telegrama

El envío de un mensaje o telegrama en un sistema EIB se realiza cuando se produce un evento, por ejemplo, la activación de un pulsador o la detección de presencia. El dispositivo emisor (sensor) comprueba la disponibilidad del bus durante un tiempo  $t_1$  y envía el telegrama. Si no hay colisiones, a la finalización de la transmisión espera un intervalo de tiempo  $t_2$  la recepción del reconocimiento (Ack). Si la recepción es incorrecta, no se recibe reconocimiento (o bien se recibe no reconocimiento), y la transmisión se reintenta hasta tres veces [10].



Figura 3.13. Secuencia de envío de un telegrama ante la activación de un evento

Todos los dispositivos disecionados envían el reconocimiento simultáneamente. Los telegramas se transmiten en modo asíncrono, a una velocidad de 9600 baudios, donde cada carácter o byte consta de 1 bit de inicio, 8 bits de datos, 1 bit de paridad par, 1 bit de parada y una pausa de 2 bits hasta la siguiente transmisión.

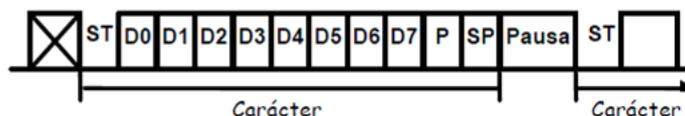


Figura 3.14. Formato de transmisión de un byte

De este modo la transmisión de un byte supone un tiempo de 1,35 ms, y la de un telegrama completo entre 20 y 40 ms (la mayoría de las órdenes son de marcha-paro y suponen un tiempo de envío de 20 ms).

El telegrama que se transmite por el bus, y que contiene la información específica sobre el evento que se ha producido, tiene siete campos, seis de control para conseguir una transmisión fiable y un campo de datos útiles con el comando a ejecutar. A continuación podemos ver un esquema de la composición de un telegrama:

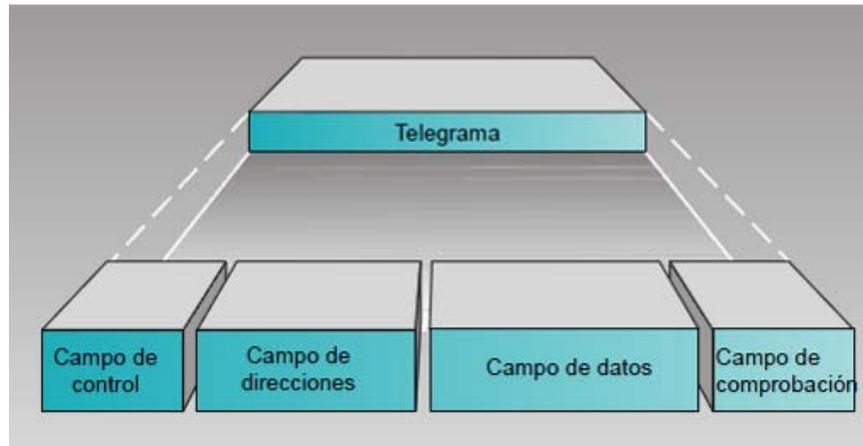


Figura 3.15 Formato de un telegrama

- Control:** Este campo de 8 bits incluye la prioridad que dicho telegrama tiene al ser enviado según el tipo de función (alarma, servicios del sistema o servicios habituales). El bit de repetición se pone a cero en caso de repetirse algún envío a causa del no reconocimiento de alguno de los destinatarios. De este modo se evita que los mecanismos que ya han ejecutado la orden la vuelvan a repetir. En la imagen podemos ver la composición de un telegrama, en donde se nos muestran cada uno de sus campos específicos y, concretamente el campo de control, con sus posibles funcionamientos.

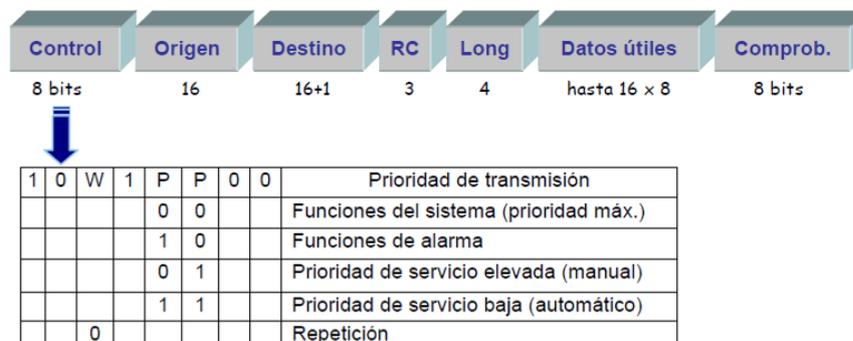


Figura 3.16. Campo control

- **Dirección de origen:** El dispositivo que retransmite la trama envía su dirección física (4 bits con el área, 4 bits de identificador de línea y 8 bits de identificador de dispositivo), de modo que se conozca el emisor del telegrama en las tareas de mantenimiento.
- **Dirección de destino:** La dirección de destino puede ser de dos tipos, en función del valor que tome el bit de mayor peso de este campo (bit 17). Si tiene valor '0', se trata de una dirección física, y el telegrama se dirige únicamente a un dispositivo. Si tiene valor '1', se trata de una dirección de grupo, y el telegrama se dirige a todos los mecanismos que deben escucharlo (los que tengan esa dirección de grupo).

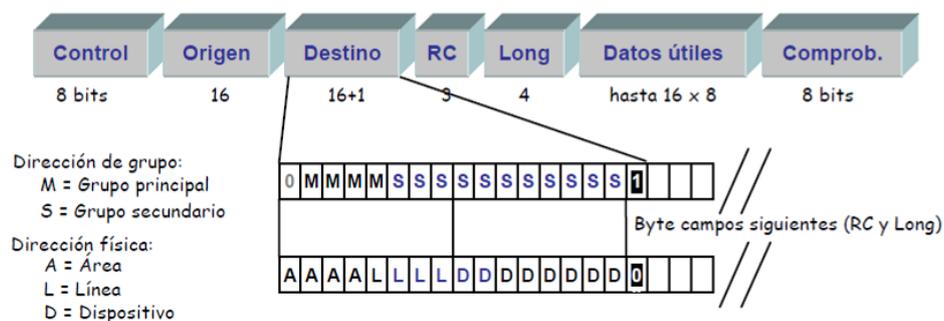


Figura 3.17. Campo de dirección destino

- **Longitud e información útil:** Contiene los datos necesarios para la ejecución de órdenes y transmisión de valores. En los cuatro bits de longitud se indica cuántos bytes contiene el campo de datos (0 = 1 byte, 15 = 16 bytes). El campo de datos útiles contiene el tipo de comando (sólo hay cuatro) y los datos, de acuerdo con el EIB Interworking Standard (EIS). En la tabla 3.2 se muestran los tipo EIS normalizados junto con su tamaño y función.

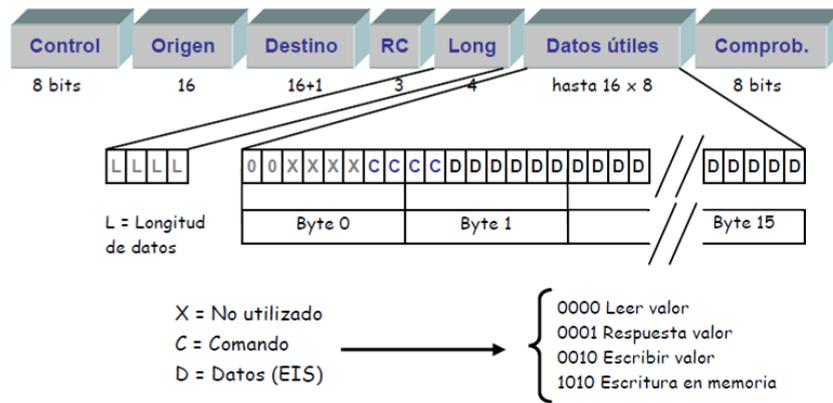


Figura 3.18. Formato del campo de datos

Nº EIS	Función EIB	Nº bytes	Descripción
EIS 1	Conmutación ( <i>switching</i> )	1 bit	Encendido/apagado, habilitar/deshabilitar, alarma/no alarma, verdadero/falso
EIS 2	Regulación ( <i>dimming</i> )	4 bit	Se puede utilizar de 3 formas distintas: como interruptor, como valor relativo y como valor absoluto.
EIS 3	Hora ( <i>time</i> )	3 bytes	Día de la semana, hora, minutos y segundos.
EIS 4	Fecha ( <i>date</i> )	3 bytes	Día/mes/año (el margen es de 1990 a 2089).
EIS 5	Valor ( <i>value</i> )	2 bytes	Para enviar valores físicos con representación S,EEEE,MMMMMMMMMMMM
EIS 6	Escala ( <i>scaling</i> )	8 bit	Se utiliza para transmitir valores relativos con una resolución de 8 bit. P.e. FF = 100 %
EIS 7	Control motores	1 bit	Tiene dos usos: Mover, arriba/abajo o extender/retraer y Paso a Paso.

	<i>(control drive)</i>		
EIS 8	Prioridad <i>(priority)</i>	1 bit	Se utiliza en conjunción con EIS 1 ó EIS 7.
EIS 9	Coma flotante <i>(float value)</i>	4 bytes	Codifica un número en coma flotante según el formato definido por el IEEE 754
EIS 10	Contador 16 bit <i>(16b-counter)</i>	2 bytes	Representa los valores de un contador de 16 bit (tanto con signo como sin signo).
EIS 11	Contador 32 bit <i>(32b-counter)</i>	4 bytes	Representa los valores de un contador de 32 bit (tanto con signo como sin signo).
EIS 12	Acceso <i>(access)</i>	4 bytes	Se usa para conceder accesos a distintas funciones.
EIS 13	Caracter ASCII <i>(Character)</i>	8 bit	Codifica según el formato ASCII
EIS 14	Contador 8 bit <i>(8b-counter)</i>	8 bit	Representa los valores de un contador de 8 bit (tanto con signo como sin signo).
EIS 15	Cadena <i>(Character String)</i>	14 bytes	Transmite un cadena de caracteres ASCII de hasta 14 bytes

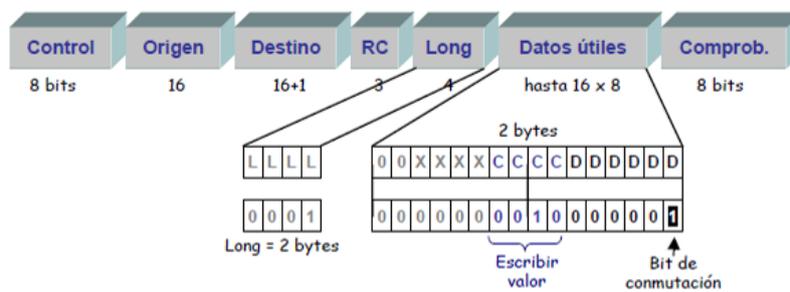
Tabla 3.1 Tipos EIS

El EIS contiene los datos útiles para cada función asignada a los objetos de comunicación. Según este estándar existen siete tipos diferentes, cada uno asignado a un tipo de acción de control (conmutación, regulación de luz, envío de valor absoluto, envío de valor en punto flotante, etc.). De este modo se garantiza la compatibilidad entre dispositivos del mismo tipo de diferentes fabricantes.

Los objetos de comunicación son instancias de clases definidas en el estándar, y se incluyen en los programas almacenados en la memoria de los dispositivos para realizar una determinada acción. Normalmente, el programa de aplicación que se ejecuta en un dispositivo dispone de varios objetos de comunicación, que pueden ser de diferentes tipos EIS. Por ejemplo, un pulsador de dos teclas con un programa de control de iluminación puede tener cuatro objetos: dos de conmutación (uno para cada tecla), tipo EIS 1, que envían las órdenes de encendido-apagado, y otros dos de regulación (uno para cada tecla), tipo EIS 2, para el envío de órdenes de incremento-decremento de luminosidad. Las asociaciones de direcciones de grupo, descritas con anterioridad, se realizan para cada uno de estos objetos de comunicación, de modo que un componente

EIB, con una única dirección física, contiene varios sensores o varios actuadores, cuyo funcionamiento lógico es independiente

En la *figura 3.19* se muestra un ejemplo de trama de datos para el control de un actuador-conmutador de iluminación (EIS tipo 1 de 1 bit).



*Figura 3.19. Ejemplo de trama de datos de conmutación*

- **Campo de comprobación:** Consiste en un byte que se obtiene del cálculo de la paridad longitudinal impar (LRC2) de todos los bytes anteriores incluidos en el telegrama, obteniendo cada uno de sus bits a partir del cálculo de la paridad impar de los bits de igual peso en el resto de campos. En la *figura 3.20* podemos ver el proceso de cálculo. Este campo de comprobación es independiente del bit de paridad par que se obtiene al realizar la transmisión en modo asíncrono de cada

byte del telegrama, y se emplea como una medida adicional para garantizar la fiabilidad en la transmisión.

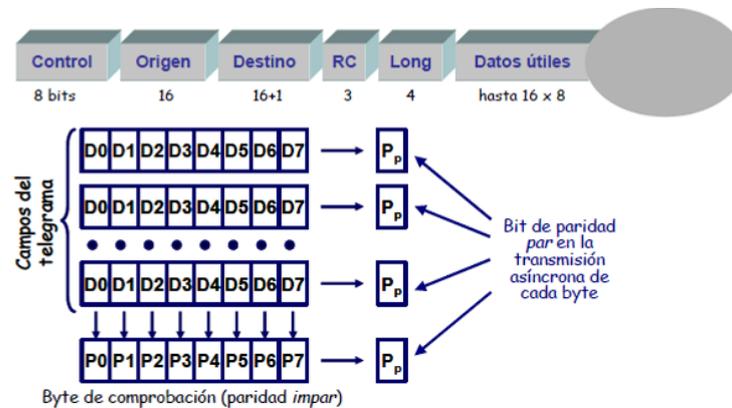


Figura 3.20. Obtención del campo de comprobación de la trama

Cuando un dispositivo recibe el telegrama, comprueba si éste es correcto a partir del byte de comprobación. Si dicha recepción es correcta, se envía un reconocimiento, de lo contrario se envía un no reconocimiento (NAK) para que el emisor repita el envío.

Si el dispositivo está ocupado envía un código Busy para que el emisor reintente la transmisión tras un pequeño retardo.



## Capítulo 4

# Entornos 3D de visualización

*“Los juegos infantiles no son tales juegos,  
sino sus más serias actividades”*

*Michel Eyquem de Montaigne (1533-1592)*

*Escritor y filósofo francés*

Al escuchar las palabras entorno virtual a cada uno de nosotros nos viene una idea de lo que puede significar esto, sin embargo a la hora de definirlo nos encontramos con que esa idea que tenemos no está tan clara como podíamos creer.

Según la Real Academia de la Lengua Española podemos definir estas palabras del siguiente modo:

Entorno: Ambiente, lo que rodea.

Virtual: Que tiene existencia aparente y no real.

Basándonos en éstas definiciones podemos llegar a la conclusión que un entorno virtual vendría a ser una emulación de un ambiente aparentemente real sin serlo. Con ésta idea en la mente si miramos a nuestro alrededor podemos ver que estos entornos nos rodean probablemente más de lo que somos conscientes, así pues un ejemplo real de éste hecho son los videojuegos.

Actualmente los videojuegos forman parte de nuestro ocio, tanto niños como adultos dedican una parte de su tiempo a disfrutar de ellos, con todo esto podemos ver que los videojuegos se han implantado en la sociedad de tal manera que prácticamente todo el mundo en sus casas poseen una o más consolas e infinidad de juegos de diversos tipos, lo cual ha convertido la industria de los videojuegos en un negocio multimillonario que supera a Hollywood y pretende desbancar a la industria discográfica como el negocio de ocio más lucrativo del mundo.

Todo esto ha logrado que la mayor parte de las personas se encuentren cómodas manejando un juego, hecho que todavía no ha alcanzado internet a pesar de llegar a la mayor parte de los hogares.

Aprovechando ésta comodidad de la sociedad, en este proyecto presentaremos un entorno manejado en primera persona por el usuario, puesto que del abanico de opciones existente entre los diferentes tipos de videojuegos éste resulta ser el menos invasivo y el que consigue una mayor interacción con el usuario.

Este juego se situará en un escenario conocido, la propia vivienda del jugador, e interactuará con los distintos paneles de control del sistema domótico localizados en diferentes puntos del hogar emulado, logrando así algo que no es tan común dentro de los videojuegos y el objetivo perseguido de este proyecto la interactividad con el mundo exterior, gracias a la cual no sólo podría ser posible el control de nuestra vivienda con un ordenador sino también con el móvil o la propia consola.

## **4.1 Historia de los videojuegos**

Fue en los años 70 cuando surgió el comienzo de la industria de los videojuegos con la comercialización del juego Pong en 1972, que fue seguido por grandes juegos aún hoy

conocidos y jugados, tales como el Pac-Man y el Space Invaders. Con respecto a las consolas fue también en esta década donde aparecieron las primeras videoconsolas caseras [13].

La década de los 80 se caracterizó por una gran crisis que azotó la industria de los videojuegos debido a un exceso de videojuegos similares y de mala calidad, fue en 1985 cuando comienza a recuperarse y Nintendo lanza el Super Mario Bros, personaje nacido de una creación muy exitosa, el Donkey Kong. Otros grandes éxitos destacaron en esta época, como el Legend of Zelda y el Maniac Mansion.

En 1989 sale a la venta la Game Boy con el mítico juego Tetris, cosechando un gran éxito y convirtiéndose en una superventas en poco tiempo. Este año sería el nacimiento de la consola portátil más famosa de todos los tiempos.

La época de los 90 se puede considerar como la década dorada de los videojuegos, apareciendo consolas como la Super Nintendo teniendo una de máximas rivalidades que ha habido en la historia con la Mega Drive, que con el Sonic lograron hacer frente a Mario llegando hasta los 4 millones de copias vendidas.



Finalmente llegamos a la era moderna, la de las consolas en HD, con disco duro, con grandes gráficos y muchos juegos.

Pero 2005 es el año en el que empieza lo que se denomina “Next-gen” (Siguiendo Generación) con la aparición de la Xbox 360. Sony no se quedaría atrás y al siguiente año presenta en el mercado su nueva PlayStation3 y Nintendo aparecería con Wii.

Esta nueva generación destaca por unos juegos con un realismo extremo consiguiendo así que el jugador se sienta introducido en el juego y facilitando la interacción del usuario con el juego.

## 4.2 Géneros de los videojuegos

Un género de videojuego designa un conjunto de juegos que poseen una serie de elementos comunes. A lo largo de la historia de los videojuegos aquellos elementos que han

compartido varios juegos han servido para clasificar como un género a aquellos que les han seguido de la misma manera que ha pasado con la música o el cine.

Los videojuegos se pueden clasificar como un género u otro dependiendo de su representación gráfica, el tipo de interacción entre el jugador y la máquina, la ambientación y su sistema de juego, siendo este el último criterio más habitual a tener en cuenta.

En este documento se va a proceder a su clasificación en 3 grupos fundamentales, los cuales se componen de diversos subgrupos que abarcan los diferentes tipos de videojuegos que hoy en día podemos encontrar en el mercado [13].

El objetivo a perseguir con esta clasificación es encontrar el género que más se adecue al trabajo a realizar para su posterior creación. Así pues necesitaremos ver el tipo de videojuego que permita una interacción con el exterior y una mayor facilidad de manejo por parte del usuario, siendo por tanto un entorno poco invasivo para el jugador.

### 4.2.1 Simulación

Éste es un género caracterizado por el extremo realismo de un aspecto de la vida real llevada al juego, en el cual el jugador tiene control absoluto de todo lo que sucede. Muchos de ellos se enfocan en inmiscuir de tal manera al jugador que piense que lo que sucede es real, sobre todo con las simulaciones de combate o pilotaje.

En éste tipo de juegos podemos ver diferentes subgéneros, vamos a destacar los más emblemáticos.

- **Simulación musical**

Su desarrollo gira en torno a la música ya sea de tipo karaoke o baile. Los juegos más famosos de éste género son el Singstar para PS2, PS3, UltraStar para



PC y Guitar Hero para Xbox.

- **Simulación de vida**

Éste género se enfoca en controlar un personaje con capacidades y emociones humanas controlando todos los aspectos de su vida, desde donde vivirá, que estudiará y hasta con quien se casará. El realismo de estos juegos y una línea de tiempo que permite al personaje evolucionar, comer, dormir, convivir con otras personas, envejecer e incluso morir, hace que muchas personas tomen estos juegos como desahogo de todo lo que quisieran experimentar en su vida pero sin riesgos. El más popular con absoluta diferencia es el juego The Sims.



- **Arcade**

Género que se caracteriza por su simplicidad de acción rápida, su auge estuvo en la época de los 80. No requiere una gran historia sólo juegos largos y repetitivos, pero adictivos. Los juegos revolucionarios de éste género fueron Pac-Man, Space Invaders, Asteroids.



- **Deportes**

Un videojuego de deportes es aquel que simula las características de un campo de deportes tradicional. Éste tipo de videojuegos han tenido siempre una gran



aceptación, y han ido mejorando, a medida que la tecnología lo permitía, siendo cada vez más realistas. Algunos de estos juegos apuestan por un mayor realismo, por ejemplo el Madden NFL, otros prefieren un enfoque más táctico, mostrando la estrategia detrás del deporte, como podemos ver en el Championship Manager y los hay que apuestan por una interacción mayor con el usuario como puede ser el wii sport, en donde el jugador debe realizar el movimiento como si realmente estuviera jugando.

## 4.2.2 Aventura

Los videojuegos de aventura son un género muy popular donde el protagonista atraviesa grandes niveles, lucha contra enemigos y recoge objetos de valor, suelen ser juegos de larga duración y argumento extenso.

Se caracterizan por la investigación, exploración, la solución de rompecabezas, la interacción con personajes del juego y un enfoque basado en el relato en lugar de desafíos de reflejos. La mayor parte de estos juegos son de computadora y a diferencia de otros videojuegos gracias al enfoque de relato es posible tomar cosas prestadas de la literatura o de las películas. Éste género puede subdividirse en diferentes tipos

- **Aventura conversacional o de texto**

Más común en ordenadores que en videoconsolas, es un género en el que la descripción de la situación en la que se encuentra el jugador proviene principalmente de un texto y a su vez el jugador debe teclear la acción a realizar tales como “coger papel” o “derretir al mago” con lo que nos encontramos que existía una gran dificultad a la hora de escoger el verbo correcto. A veces existen gráficos en estos juegos que ofrecen ayuda complementaria.



- **Aventura gráfica**

Sucesor de las aventuras conversacional el jugador debe ir avanzando y resolviendo los diferentes enigmas que nos plantea la historia. Gracias a la aparición del ratón y el avance de los gráficos se nos permite la interacción con el escenario de una forma más gráfica.



Cabe destacar en este género juegos como el Maniac Mansion, Hollywood Monsters, la saga Monkey Island y el Broken Sword, entre muchos otros

- **Rol**

Es un género de videojuegos que usa elementos de los juegos de rol tradicionales, se caracterizan por la interacción con el personaje, una historia profunda y la evolución del personaje a medida que avanza la historia.



En estos juegos el personaje ha de cumplir una serie de objetivos o misiones en un mundo, generalmente, perteneciente a un tema de fantasía épica, en donde el jugador explora éste mundo conociendo nuevos personajes y recolectando armas, experiencia, aliados e incluso magia. En este género debemos destacar el Final Fantasy, Dungeons & Dragons, Diablo II y el World of Warcraft

### 4.2.3 Acción

Al referirnos a la categoría de acción en un videojuego, hablamos de uno en el que el jugador debe usar su velocidad, destreza y tiempo de reacción. Éste género es el más

amplio y que más juegos abarca del mundo de los videojuegos englobando por tanto muchos subgéneros.

- **Lucha**

Como su propio nombre indica éste tipo de juegos consisten en recrear combates entre personajes, pudiendo ser controlados tanto por la máquina como por el jugador.



Se nos presenta a los combatientes desde una perspectiva lateral y ponen especial énfasis en artes marciales, generalmente imposibles de imitar. En este género podemos destacar el mortal kombat y street fighter como clásicos de la temática y el Assassin's creed II como uno más moderno

- **Beat them up**

Los beat'em up o "juegos de pelea a progresión" son juegos similares a los de lucha, salvo que en este tipo los jugadores deben combatir con un gran número de adversarios mientras avanzan por los diferentes niveles. En este tipo de juegos existe la posibilidad de realizar una partida multijugador de forma cooperativa facilitando así el trabajo. Podemos destacar el street of race o el doble dragón como ejemplos de éste tipo de videojuegos



- **Infiltración**

Los juegos de infiltración son de un género relativamente reciente que no se popularizó hasta la salida del juego Metal Gear Solid en 1998. Éste tipo de videojuegos, también llamados de sigilo o espionaje, se basan en el sigilo, la furtividad y la estrategia, en lugar de la búsqueda de confrontación directa con el enemigo.



- **Plataformas**

Éste es uno de los subgéneros más exitosos de todos los tiempos, en ellos el jugador controla a un personaje que debe avanzar por el escenario evitando obstáculos físicos, bien sea saltando, escalando o agachándose. En un principio los personajes se movían por niveles de forma horizontal, pero en la actualidad con la llegada de los gráficos 3D, este desarrollo se ha ampliado hacia todas las direcciones posibles.

- **Disparos**

Los videojuegos de disparos o shooters son un subgénero de los juegos de Acción que a su vez puede abarcar un amplio número de subgéneros con la característica común de controlar un personaje que, por norma general, dispone de un arma (normalmente de fuego) la cual puede ser disparada a voluntad.

- **Acción en tercera persona**

Los juegos de disparos en tercera persona o también llamados TPS se basan en la alternancia de los



disparos y las peleas o la interacción con el entorno, se juegan con un personaje visto desde atrás y en ocasiones desde una perspectiva isométrica, sacrificando precisión a cambio de una mayor libertad de movimientos. Algunos ejemplos de este género serían Tomb Raider y Resident Evil.

- **Juegos de disparo en primera persona**

Los juegos de disparo en primera persona, también conocidos como FPS, poseen como acciones básicas mover al personaje y manejar un arma, la cual se muestra en pantalla en primer plano y el jugador interactúa con éste. Con ésta perspectiva se pretende dar al jugador la impresión de estar ahí mismo y así poder tener una identificación fuerte con la situación, para lo cual es uso de gráficos en tres dimensiones ayuda a aumentar esta impresión.



Éste tipo de juegos no posee un guión elaborado, salvo algunas excepciones como por ejemplo Halo, sin embargo poseen una gran calidad gráfica llegando a rozar el fotorrealismo en algunos casos.

Permite además partidas en línea, siendo por tanto posible la conexión a internet, facilitando la interacción con el mundo exterior.

Ciertos videojuegos marcaron el género, tales como Doom, Half Life, Unreal, Quake, Far Cry, Counter Strike.

### **4.3 Nuestro entorno virtual**

Como ya mencionamos anteriormente, el manejo de este proyecto por parte del usuario se realizará a través de nuestro propio videojuego.

Tras el estudio presentado en el apartado anterior y dado que es necesario un tipo de videojuego que genere una mayor comodidad para el usuario, en la que se sienta parte de las acciones que realiza y con el cual sea posible la interacción con el mundo exterior para poder realizar así el control de la vivienda, se escogió realizar un videojuego de tipo simulación basado en un modelo de tipo FPS, puesto que de los diferentes géneros estudiados es el que cumple de una forma más exitosa el propósito buscado.

Se podría decir que el videojuego de este proyecto sería del tipo simulación, puesto que emulamos un entorno real, nuestra vivienda, siendo por tanto un escenario conocido para el usuario, se podrá ejecutar tanto en primera persona, es decir, sin ver al personaje, únicamente viendo el entorno que nos rodea cómo si estuviéramos presentes, o en tercera persona, en donde podemos ver un personaje al que controlamos. Con estos dos modos se pretende que el jugador pueda seleccionar la opción más cómoda para él.

Para la creación de un videojuego se utilizan lo que se viene a denominar como motores de juegos, los cuales estudiaremos en mayor profundidad en el siguiente apartado, que consiste en un programa dónde se reúnen diferentes opciones necesarias para la creación de un juego, tales como, el apartado gráfico, la programación, etc. Dependiendo del tipo de juego se utiliza un motor u otro, ya que están enfocados para la creación de un estilo de juego determinado, aventura gráfica, rol, plataformas, FPS, etc.

En el caso de este PFC se ha optado por la creación de un juego de estilo FPS, por lo que se hace necesaria la utilización de un motor de juego especializado para este tipo concreto. La elección del motor de juego de este proyecto fue el 3D GameStudio ya que posee diferentes características que nos facilitarán la creación del apartado gráfico, así como de la interacción con el mundo exterior. En el siguiente capítulo se procederá a detallar las posibilidades ofrecidas por este motor señalando las que fueron de utilidad a la hora de realizar el desarrollo del sistema.

Puesto que el trabajo realizado no es un videojuego propiamente dicho, ya que carece de niveles, enemigos u obstáculos a superar, procederemos a referirnos a él como entorno virtual en el cual, el usuario únicamente realizará las acciones de control de la vivienda, tales como la regulación de la iluminación o el control de las persianas.

## 4.4 Motores de juego. GameStudio.

A la hora de realizar un juego, hay diversos aspectos que debemos tener en cuenta si deseamos que nuestro personaje se mueva en un mundo y posea un aspecto lo más real posible. Éstos no son más que aquellos que podemos percibir a nuestro alrededor, tales como el movimiento del viento, el ruido que nos rodea, los diferentes movimientos corporales que realizamos etc. Cómo podemos ver, identificar estos aspectos no supone una gran dificultad, el auténtico problema reside en trasportarlos a nuestro mundo artificial.

Para ello necesitaremos una serie de conocimientos o herramientas tales como la física, las matemáticas, la comunicación, la inteligencia artificial y, por supuesto, la programación.

En este punto podemos ver como a lo largo de la historia se han producido grandes mejoras, llegando a alcanzar niveles de realismo como el que nos muestra el juego de Assassins Creed, en contraposición a los inicios de los videojuegos con el Super Mario Bros.

### 4.4.1 Conocimientos previos

En este punto trataremos de ver estas herramientas de las que hablábamos anteriormente necesarias para la creación de un juego que permita al usuario sentirse en un escenario real y posible [8, 9].

#### 4.4.1.1 La programación

El lenguaje de programación por excelencia en los videojuegos ha sido el C, aunque actualmente el C++ lo ha desbancado dada sus múltiples ventajas y la posibilidad de una

programación orientada a objetos, por otro lado también tenemos C#, basado en el C y utilizado para las plataformas .NET de Microsoft.

Estos lenguajes de los que hablamos son compilados, es decir, necesitamos una herramienta clave para que funcionen, el compilador. Sin embargo, actualmente tenemos otras opciones en lenguajes de programación que se han hecho un hueco muy importante, son los lenguajes interpretados o lenguajes scripts en donde cada uno de estos contiene variables, datos y código. Por ejemplo si en un juego necesitamos crear interfaces, o mantener posiciones de ciertos objetos, es más cómodo tener un archivo con esta información y que desde el programa principal se lea el archivo y se interprete, esto facilita mucho la modificación del juego, tanto en el proceso de desarrollo como en futuras actualizaciones, y nos permite liberarnos de estar compilando cada vez que hagamos un cambio en el código fuente. Existen lenguajes de scripting que nos facilitan la vida como Python y LUA, pero como ya dijimos es C/C++ quien encabeza la lista de lenguajes de programación, teniendo también su versión para script que será la usada en este trabajo, concretamente usaremos C-Script.

#### 4.4.1.2 La física

Este es un pilar muy importante para muchos videojuegos, si nosotros creásemos nuestro videojuego no podremos olvidar que para lograr un movimiento del personaje necesitaríamos dotarlo de algunas de las fuerzas que dominan la naturaleza: gravedad, densidad, aceleraciones, rozamientos, todo aquello que inunda los libros de física. Por ejemplo en plataformas, como el Super Mario Bross, se aplican ciertas cosas de cinemática básica en la que los personajes, como el movimiento que realiza, cuando salta, caen, etc y por supuesto todo el tema relacionado con las colisiones y respuestas forma parte de la física del juego.

Con todo esto somos capaces de crear personajes y mundos lo más reales posibles, todas estas consideraciones, no son algo actual que se ha tenido en cuenta para mejorar el realismo, como vemos con el ejemplo del Super Mario Bross todo esto ya se estudiaba en los inicios de los juegos, de tal modo que el hecho de que el personaje de Mario lleve una gorra se debe, nada más y nada menos, a cubrir el problema de que al caer su pelo debería

moverse y en aquella época no les era posible conseguir ese efecto, así pues, para subsanar este problema se le introdujo una gorra al personaje.

Con todo esto quiero explicar que la física es algo que nos rodea y que, como es lógico siempre se ha considerado a la hora de generar un mundo virtual, logrando a medida que la tecnología avanza ir aumentando sus efectos y su realismo en este mundo.

#### **4.4.1.3 Las matemáticas**

De entre todas las herramientas necesarias, probablemente una de las más primordiales sean las matemáticas, puesto que no sólo nos las encontraremos al intentar aplicar la física en nuestros juegos, sino que también formar parte de algo mucho más complejo, el motor tridimensional.

Es por esto que a la hora de generar un juego necesitaremos tener un profundo conocimiento del mundo del álgebra línea y vectorial, además no sólo debemos saber que cálculos hacer, también hay que hacerlos de una forma rápida y eficiente.

Todos estos conocimientos matemáticos y físicos nos facilitarán el trabajo a la hora de realizar las diferentes perspectivas, el control de la luminosidad, las reflexiones, la transmisión de la luz, etc.

#### **4.4.1.4 La inteligencia artificial (IA)**

La inteligencia artificial en un videojuego son todos los comportamientos que tienen los personajes que no controlamos, como enemigos, transeúntes, jefes finales, etc [9].

Así pues la IA son grupos de instrucciones y animaciones que el NPC (No Playable Character) tiene que aplicar en determinadas circunstancias, por ejemplo que le ataquen, que esté patrullando, que esté paseando por la calle, que me acerque y le pregunte, etc.

Aplicando conocimientos de IA podemos darle a nuestro juego un toque más real y entretenido. Algunos métodos de la IA aplicadas al desarrollo de juegos son el PathFinding (Implementado a través del Algoritmo A\* para la búsqueda del camino más corto entre dos puntos) aplicado en juegos como Starcraft, cuando por ejemplo queremos mover un personaje a cierto punto del mapa y este es capaz de llegar a ese punto esquivando todos los objetos que encuentre a su paso y usando el camino más corto. Otros métodos aplicados son los algoritmos genéticos, vida artificial, máquinas de estados finitos, árboles de búsqueda, también áreas extensas de la IA como las redes neuronales, etc.

#### **4.4.1.5 La comunicación**

Con este punto nos referimos a todo lo relacionado con la comunicación de datos a través de redes, por ejemplo en todo juego multijugador debemos estar enviando datos de posiciones, puntajes y otras características de personajes, objetos, etc. a otras máquinas para que exista una interacción entre todos los jugadores con el fin de ver reflejado en nuestro PC lo que están haciendo el resto de los jugadores. Esto es usado hoy en día en todos los juegos multijugador, gran ejemplo de esto es Unreal Tournament. En esta área también disponemos de bibliotecas especializadas en el tema, un ejemplo es SDL\_Net.

En nuestro trabajo utilizaremos esta herramienta de los videojuegos para poder comunicarnos con nuestra página web en dónde controlaremos nuestra casa domótizada.

#### **4.4.1.6 Application Programming Interface (API)**

Una API (Application Programming Interface) o Interfaz de Programación de Aplicaciones, es un conjunto de funciones que realizan tareas específicas facilitando la vida al programador. Al hablar de API grafica nos referimos a un set de funciones para inicializar por ejemplo modos gráficos, realizar copiado de datos de la memoria del computador a la tarjeta de video (blitting), etc. Existen APIs específicas para cada tipo de

tarea que queremos realizar. Actualmente las APIs Gráficas más usadas son OpenGL y DirectX.

- **OpenGL** (Open Graphics Library) es una API multiplataforma creada por Silicon Graphics en 1992, que maneja solo el aspecto gráfico de un sistema, dejando afuera el sonido, música, control de teclado, mouse, joysticks, gamepads, etc., los que deben ser controlados con otras APIs especializadas. Existe una biblioteca multiplataforma para el manejo de audio tridimensional llamada OpenAL (Open Audio Library).



- **DirectX** es una API multimedia creada por Microsoft en 1995, que consta básicamente de Direct3D para la parte gráfica, DirectSound y DirectMusic para la parte de audio, y DirecInput para el control de teclados, joysticks, etc. También incluye DirectPlay para comunicación de datos en redes.



El uso de estas dos APIs en un comienzo puede ser un poco complicado y engorroso, especialmente DirectX. Existen algunas alternativas como Allegro o SDL, las que son multiplataforma.

#### 4.4.2 Game Engine

Un game engine (motor de videojuego) es un término que hace referencia a una serie de rutinas de programación y herramientas que permiten el diseño, la creación y la representación de un videojuego.

Los game engine están basados en algunas de las APIs que hemos visto OpenGL o DirectX principalmente, que proveen al programador todas las funcionalidades necesarias para el desarrollo de un juego.

Básicamente un Engine está formado por varios sistemas y subsistemas, por ejemplo un sistema gráfico para manejar objetos 2D o 3D, un sistema de control de entrada (teclado, mouse, etc.), sistema de texto, sistema de red, sistema de scripts, sistema de audio, etc. Todos los conocimientos necesarios que vimos antes ahora están aplicados en un Engine. Las tareas más comunes de un game engine son:

- ✓ Detección de colisiones
- ✓ Render de imágenes (2d y/o 3d)
- ✓ Manejo de Memoria
- ✓ Manejo de Audio

Hoy en día, los engines más sofisticados incluyen la integración de otros engines y software dedicado a tareas específicas también conocidas como Middleware. Las más comunes son:

- ✓ Physics Engine
- ✓ AI Engine
- ✓ World/Level Editor
- ✓ Animation Editor
- ✓ Networking Engine
- ✓ Scripting Editor

Existen por lo general 2 tipos de game engines:

- **Cerrados:** Aquellos engines que no permiten tener acceso al código fuente y por lo mismo no hay forma de poder modificarlos o expandir sus capacidades. Ejem: UDK, Unity Indie, Game Studio.
- **Abiertos:** Aquellos engines que permiten tener acceso al código fuente y así poder adaptarlos a las necesidades del proyecto. Ejem: Torque, Orgre 3D, Nebula.

Cada uno ofrece ventajas y desventajas, en el caso de un engine cerrado, es más amigable para usuarios poco especializados y con conocimientos limitados, mientras que un engine abierto requiere de usuarios más avanzados. La velocidad de desarrollo es también un punto clave, así como las plataformas que soportan.

#### 4.4.2.1 Game Engine cerrados

##### 4.4.2.1.1 UDK



El Unreal Engine es un motor de juego desarrollado por Epic Games, apareció en 1998 para el juego de FPS Unreal. Aunque en principio había sido desarrollado para juegos FPS, se ha utilizado con éxito en una variedad de géneros, entre ellos el sigilo, los MMORPG y juegos de rol. Con su núcleo escrito en C++, el Unreal Engine cuenta con un alto grado de portabilidad y es una herramienta utilizada por los desarrolladores de juegos hoy en día.

La última versión es la UE3, diseñado para Microsoft DirectX 9 (para Windows y Xbox 360), DirectX 10 y 11 (para Windows Vista y posterior), OpenGL para Linux, Mac OS X, PlayStation 3, iOS, Android, y la etapa en 3D de Adobe Flash Player 11 [15].

##### 4.4.2.1.2 Unity

Unity es una herramienta para la creación de juegos 3D o de otros contenidos interactivos tales como visualizaciones arquitectónicas o animaciones 3D en tiempo real. Es compatible con Microsoft Windows y Mac OS X, y sus juegos son ejecutables en Windows, Mac, Xbox 360, PlayStation 3, Wii, iPad, iPhone, así como la plataforma Android.



Unity posee tanto un editor para la programación y un game engine para la ejecución del producto final. Es similar a otros motores como Torque, Blender, Director y Gamestudio, los cuales usan un entorno gráfico integrado como el principal método de desarrollo [14].

#### 4.4.2.1.3 Scaleform

Autodesk Scaleform combina la potencia y el rendimiento de la tecnología moderna de gráficos en 3D con la probada productividad y flujo de trabajo de Adobe Flash y Creative Suite, que genera la creación de contenido de mayor calidad. Sus poderosas características han hecho de la solución Autodesk Scaleform la preferida en más de 800 proyectos, que abarcan desde títulos AAA a los juegos de menor presupuesto y electrónica de consumo. Scaleform proporciona una solución optimizada para crear menús de aceleración de hardware 3D del juego, HUD, texturas animadas, videos en el juego y mini-juegos.

Esta herramienta se ha desarrolla para maximizar el rendimiento en todas las plataformas de videojuegos más importantes como son: Windows, Linux, Mac, PSPVITA, XBOX 360, Playstation 3, Wii, Nintendo 3DS, Ios y Android [16].



## 4.4.2.2 Game Engine abiertos

### 4.4.2.2.1 Torque



Torque Game Engine (TGE) es el motor desarrollado por DINAMIX para su juego Tribes2. Enfocado a la simulación de misiones militares, incluye utilidades para la creación de terrenos, superficies acuáticas, interiores estilo portal y sistemas de partículas. También incluye soporte multiplataforma (Windows, Mac OS y Linux), soporte para red, creación de interfaces de usuario y lenguaje de script estilo C++. Permite importar objetos desde 3D Studio MAX y dispone de librerías matemáticas, de detección de colisiones, de física de vehículos y una Base de Datos Espacial [18].

### 4.4.2.2.2 Ogre 3D

OGRE (Object-Oriented Graphics Rendering Engine) es un motor escrito en C++flexible, orientado a escenas, y diseñado para hacer más simple e intuitiva a los desarrolladores la producción de juegos utilizando hardware de aceleración 3D. La librería de clases permite abstraer los detalles asociados a las librerías de bajo nivel (OpenGL o Direct3D), proporcionando una interfaz basada en objetos [17].



### 4.4.2.2.3 Nebula



"The Nebula Device" es un nuevo motor de juegos de calidad profesional, que puede utilizarse de forma gratuita. Sus creadores, son el

equipo que desarrollo “Urban Assault” (Publicado pro Microsoft en 1998).

Nebula es un motor de arquitectura moderna. Desarrollado en C++ y orientado a objetos, sus clases dlls se cargan de forma independientemente en tiempo de ejecución. Este motor puede ejecutarse en Linux, Win9X, WinNt. Permite intercambiar sin interrupción la visualización con OpenGL y Direct3D. Nebula utiliza como lenguaje de script el estandar tcl/tk. Los principales objetivos de Nebula son los siguientes [19]:

- Independencia de la plataforma
- Gestión de la base de datos del juego: jerarquías 3D, texturas, materiales, luces, sonidos, animaciones, estados y sus relaciones.
- Proporcionar herramientas básicas para el trabajo en equipo para el desarrollo del juego.

#### 4.4.2.3 Nuestro Game Engine: Game Studio



3D GameStudio es un game engine de tipo cerrado. Provee de un motor 3D, un motor 2D, un editor de niveles y modelos, compilador de scripts y librerías de modelos, texturas, etc. Manipula con igual rendimiento escenas de interior y de exterior. Tiene un motor de iluminación que soporta sombras verdaderas y fuentes de luz en movimiento. El principal objetivo que esta aplicación persigue es que el creador del juego no necesite ser un programador experimentado. Ofrecen tres posibilidades para crear un juego [20]:

1. Juegos diseñados a base únicamente de ratón, para usuarios sin conocimientos de programación
2. Juegos o efectos diseñados con algo de programación utilizando C-scripts, para el que quiere algo más

3. Juegos o efectos programados en C++ o Delphi, para programadores con experiencia

#### 4.4.2.3.1 Entidades

Una entidad es un objeto externo que se puede agregar a los niveles. Existen cuatro tipos de entidades: modelos, sprites, entidades de mapa y entidades de terreno. Nosotros hablaremos únicamente de dos tipos de entidades, modelos y sprites.

Un modelo es un objeto animado 3D, almacenado en un archivo con extensión “mdl”. Los modelos se pueden crear con software como Med (incluido en 3DGameStudio). Un modelo consiste en varios triángulos (también conocidos como caras o polígonos) pegados juntos y una piel que los cubre. Siempre que necesitemos un objeto animado 3D (para el jugador, sus enemigos, animales, etc) necesitamos utilizar un modelo. Debemos utilizar modelos para todos los objetos y/o personajes en nuestro juego.

Un Sprite en cambio, no es nada más que gráficos 2D creados con un programa de dibujo. Podemos utilizar los sprites para el diseño de paneles, señales de una calle, cuadros de pinturas, juegos 2D, efectos especiales... podemos incluso utilizarlos como falsos elementos 3D: árboles, arbustos, hierba, etc. Los sprites se pueden acomodar en una de las siguientes formas:

- 1) Sprites orientados, como las pinturas que cuelgan en una pared;

Si roto alrededor de un sprite orientado puedo ver que no es nada más que un objeto 2D, así que utilizaremos un sprite orientado para una señal en la pared, un cuadro, una mancha de sangre en el suelo, etc.

- 2) Sprites que se colocan de frente al jugador en todo momento.

Éste es un sprite de tipo regular, se utiliza con el fin de mostrar un objeto 2D creando la ilusión de que se trata de un objeto 3D para maximizar el funcionamiento del PC. Puedes dar la vuelta alrededor de él tantas veces como quieras ya que siempre se mostrará de frente.

En este proyecto se utilizó una entidad de sprite orientado para emular el panel de control domótico.

#### 4.4.2.3.2 Editor WED

Mediante esta herramienta, que identificamos como Level Editor en la carpeta de programas que genera la instalación del 3D GameStudio, se diseñan los diferentes niveles o escenarios de los que consta el juego. En nuestro caso utilizamos ésta herramienta para la creación de nuestra casa, que será el entorno en donde nos moveremos para poder accionar el sistema domótico.

El editor wed es una herramienta que se basa en la combinación de elementos simples para formar un mundo complejo, es decir, basándose en formas sencillas como cubos, pirámides y esferas podemos crear nuestro propio mundo virtual. Así pues se podría afirmar que la forma en la que GameStudio divide el diseño empleando objetos simples que al agruparse forman objetos más complejos, es un modo de orientación a objetos. Cada elemento que conforma cada nivel de juego es un objeto con una serie de propiedades predeterminadas, pero que pueden escogerse y cambiarse para cada nivel creado. Los elementos agrupados entre sí (grupos de objetos) contendrán a su vez propiedades comunes para todos los elementos de su grupo [8].

El editor considera todo elemento como un objeto, el cual puede ser una entidad un bloque, o un grupo de otros objetos. El agrupar te permite construir estructuras como una escalera, un arco, un puente, o una silla, todo a partir de simples bloques.

La metodología de edición orientada a objetos esconde del usuario la información innecesaria. Con ella, puedes construir un nivel basado en objetos lógicos en lugar de miles de bloques y vértices. La metodología también introduce la modularidad y la portabilidad. El usuario puede construir secciones independientes de un nivel, sin preocuparse del resto

del mapa, siendo así capaz de reunir librerías de objetos complejos reutilizables. Elimina el trabajo extra requerido para volver a crear estructuras similares en mapas diferentes. También abre la posibilidad de usar librerías de objetos y estructuras provistos por terceros [8].

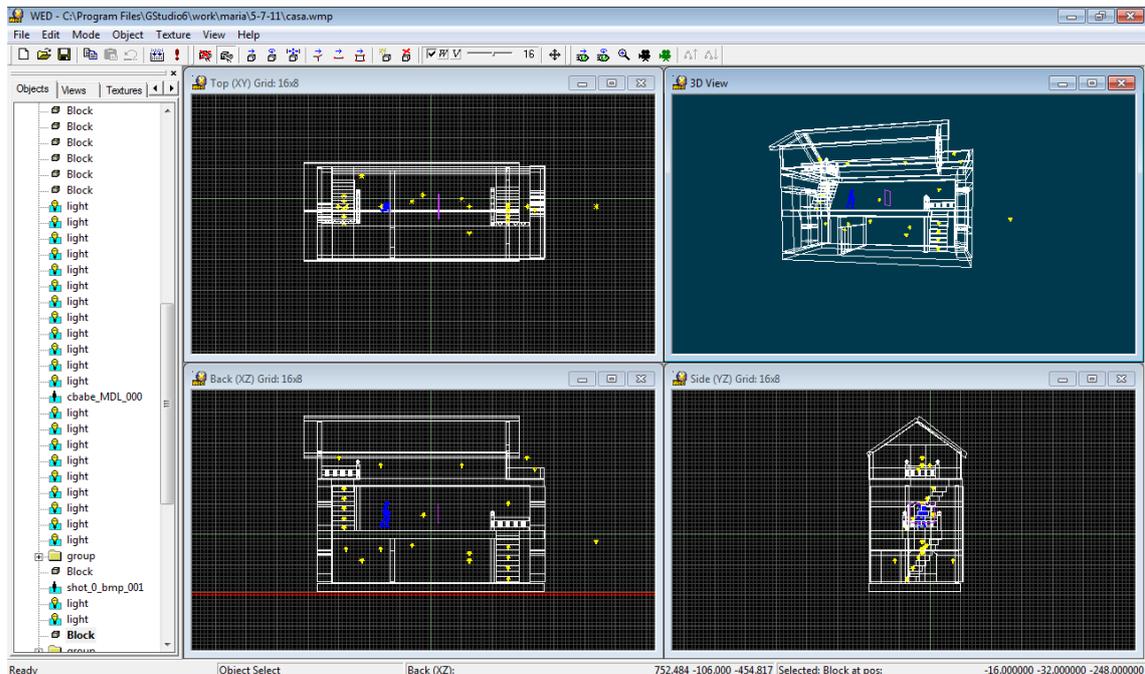


Figura 4.1. Editor WED con el diseño de la vivienda

#### 4.4.2.3.3 Editor SED

Este es el editor de C-Script mediante el cual podremos programar nuestros juegos, tanto como queramos complicarlo. Para integrar la interactividad en los juegos diseñados mediante GameStudio, la herramienta incorpora un lenguaje de programación denominado C-Script, cuya sintaxis y modo de uso son muy similares a los del lenguaje JavaScript.

Como todos sabemos un lenguaje de programación se compone de multitud de conceptos, en esta memoria únicamente hablaremos de uno de ellos, las acciones, puesto que ha sido el necesario para el desarrollo de éste proyecto.

Una acción es una función especial que se puede unir a una entidad: un jugador, un ascensor, un interruptor, un arma, etc. Es posible la unión de acciones a cualquier entidad, todas las acciones creadas aparecen en la lista de acción del editor wed.

A diferencia de una función una acción no necesita ser llamada, necesita ser unida a una entidad, es decir, mientras que a una función se le pasan una serie de parámetros para que realice la operación que deseemos, una acción es asociada a una determinada entidad lo cual le permite realizar diferentes operaciones, por ejemplo, si asocio una acción que permita el control de la iluminación a una determinada entidad, un panel, un botón, etc, cuando mi personaje se acerque a la entidad correspondiente podrá pulsarla y esto hará que la acción comience.

En este proyecto se ha utilizado este editor para la programación de determinadas acciones, es decir, se genero una acción para realizar una llamada a la página web en donde se encuentra el control de nuestro sistema domótico [8].

```

1100  msg_string=txtap_str; //llamamos al string anterior
1101  show_message(); //mostramos el mensaje
1102
1103  splashscreen4.visible = on; //al cliquear vamos a mostrar una imagen
1104  MOUSE_MODE=0; //el puntero no es visible en la pantalla, con un 1 el puntero no sólo sería visible sino que se podría usar para
1105  inkey(entry); //recoge un enter
1106
1107  //wait(13); // wait 3 frames (triple buffering) until splash screen is visible
1108  //level_load(techdemo_vmb);
1109  //load_status(); // restore global skills
1110  //wait(3); // wait until level loaded
1111  //sleep(1); // wait 1 second if it's a commercial application
1112
1113  splashscreen4.visible = off; // quitamos la imagen
1114  bmap_purge(splashaap4); // remove splashscreen from video memory
1115
1116  -}
1117
1118  function desmo()
1119  {
1120      exec("C:\\abrete.bat","");
1121
1122      //exec("C:\\Archivos de Programa\\Internet Explorer\\iexplore.exe","http\\192.168.0.1");//abre el firefox y la pagina pedida
1123  }
1124  //exec("C:\\archivos de programa\\Internet Explorer\\Mozilla Firefox\\iexplore.exe;firefox.exe","minimize","http://desmotivaciones.es")
1125  ACTION luz
1126  {
1127      my.enable_click=on;
1128
1129      //bmap splashmap = <Alien.bmp>
1130      my.event=txtap;
1131
1132      my.event=desmo; // llamamos a la función correspondiente que nos abre el navegador
1133
1134
1135      // return(0);
1136  }
1137
1138  //////////////////////////////////////

```

Figura 4.2. Editor SED mostrando parte del código del proyecto

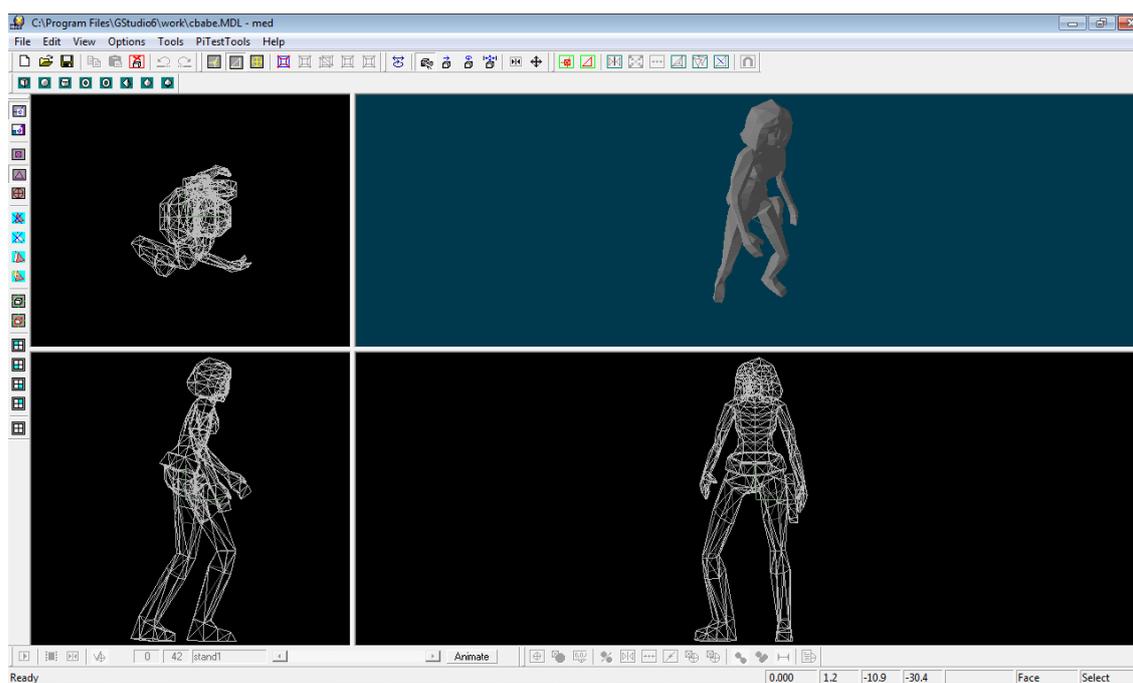
#### 4.4.2.3.4 Editor MED

El editor MED es la herramienta de 3D GameStudio que permite diseñar e importar modelos 3D e integrarlos en nuestro juego.

El editor med nos permite la creación desde botones hasta personajes, siendo el procedimiento muy similar al del editor wed, con la diferencia que en este caso podemos generar movimiento en nuestro diseño, es decir, mientras que con el editor wed podemos crear escenarios el editor med nos permite la creación de personas y objetos, como por ejemplo, un interruptor, un tanque, un helicóptero, etc

Como podemos ver, todos estos modelos que genera el editor necesitan movimiento, es decir, de nada sirve crear a un personaje si no podemos hacer que se mueva, y que se mueva de una forma lógica. Todo esto es lo que nos permite este editor, con el es posible añadir acciones a nuestro modelo, cosa que resulta impensable si hablamos de otro tipo de entidad tales como escenarios.

Al generar un modelo con este editor y añadirlo a nuestro escenario podemos aplicarle acciones, tales como correr, encender las luces, etc. Todo esto permite que nuestro personaje pueda recorrer el escenario e interactuar con el mismo.



*Figura 4.3. Editor MED mostrando el personaje utilizado*



# **Bloque III**

## **Desarrollo**



## Capítulo 5

# Diseño e implementación de la plataforma virtual

*“La programación es una carrera entre ingenieros de software luchando para construir programas cada vez más grandes, mejores y a prueba de idiotas, y el universo intentando producir cada vez más grandes y mejores idiotas.*

*Por ahora, gana el universo”*

*Rich Cook. Escritor y empresario*

Como ya se vio en el capítulo anterior para la creación de una plataforma virtual es necesaria la utilización de lo que se conoce como game engine. Cada game engine es diferente y está focalizado para el tipo de juego que se desea desarrollar, así pues si se piensa desarrollar un tipo de juego como la aventura gráfica, no puede escogerse el mismo game engine que el que sería necesario para la creación de una plataforma para un juego FPS.

Tal y como se explicó anteriormente a lo largo de este documento, en este proyecto se optó por la creación de un entorno virtual de tipo FPS, puesto que cumplía las condiciones necesarias para el desarrollo de este proyecto final de carrera. Así pues para la creación de este entorno virtual se seleccionó el game engine GameStudio, ya que no sólo

cumplía con las necesidades existentes para la creación de la plataforma virtual, sino que además proporcionaba un entorno simple, de fácil manejo y muy completo, que permite modificar todo lo que se desee hasta el grado de complejidad buscado.

A lo largo de este capítulo se mostrarán los diferentes pasos realizados para el desarrollo de nuestro entorno virtual y su correcto funcionamiento. La vivienda representada se basó en un modelo existente, que se corresponde a la maqueta de una casa de muñecas. Puesto que el objetivo del presente proyecto es el desarrollo de esta interfaz y al no existir una vivienda domotizada concreta, se optó por el uso de una maqueta, sin que por ello exista un cambio significativo en lo concerniente a la creación de la plataforma virtual y su funcionamiento.

## **5.1 Diseño de la vivienda. Level Editor**

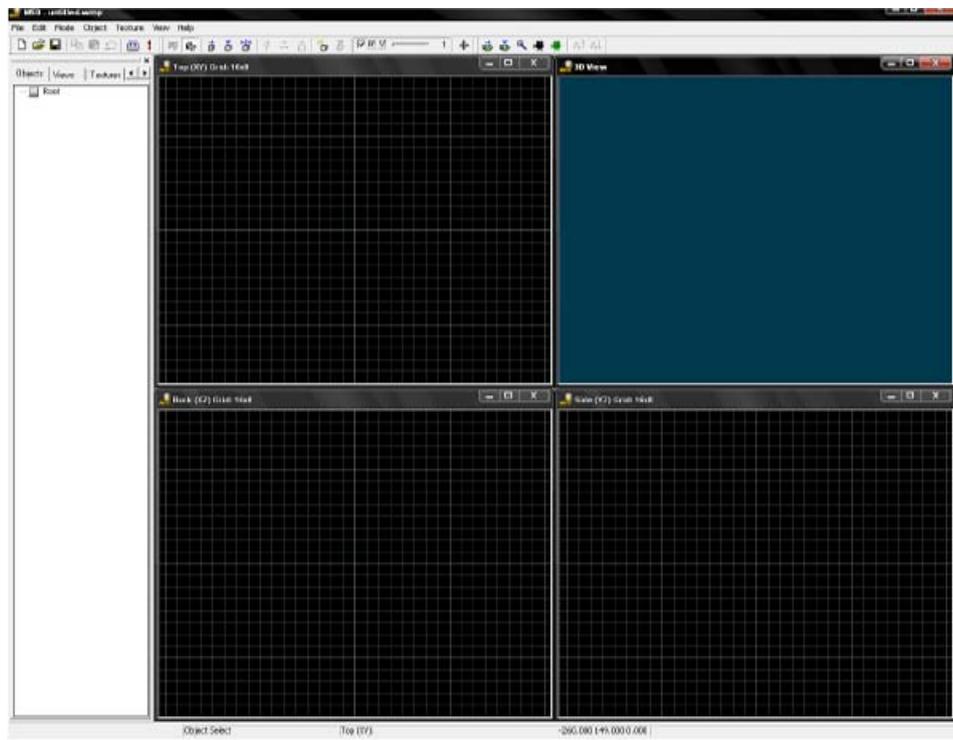
Como se explicó en el capítulo anterior el game engine seleccionado está compuesto de tres tipos de plataformas para el diferente desarrollo de cada una de las partes que conforman el entorno virtual, tales como, los gráficos, la programación, los modelos, etc.

Para realizar la representación de la vivienda en el entorno virtual seleccionado deberemos trabajar en la plataforma Level Editor, puesto que es lo que se denomina un mapa, es decir, el terreno en donde se desenvolverá el personaje, en nuestro caso la vivienda.

Para explicar el trabajo realizado en este editor se comenzará con una explicación de los diferentes paneles que presenta mientras se explicará cómo se creó el susodicho mapa [8].

### 5.1.1 El intefaz y la barra de menus

Al abrir el level editor tenemos una ventana subdivida en cuatro, estas ventanas representarán las cuatro perspectivas del mapa a tratar, es decir, las tres vistas principales (planta, alzado y perfil) y una vista tridimensional. Mediante estas vistas es posible la creación de un mapa ajustando las formas básicas con las que se trabaja en este programa (cubo, pirámide y esfera) para crear los diseños deseados.



*Figura 5.1. Vistas del level editor*

Como se puede ver a continuación en la imagen este editor presenta el siguiente menú, el cual se procederá a explicar.



*Figura 5.2. Barra de menú*

- 1: Nuevo Documento, Cargar y Guardar
- 2: Copiar, Pegar y Deshacer
- 3: Compilar y Ejecutar
- 4: Eliminar Objeto, Seleccionar, mover , girar y agrandar
- 5: Editor de vértices
- 6: Añadir prefabricado y eliminar.
- 7: Control sobre la precisión del ratón en la rejilla.
- 8: Control sobre las vistas (Ir de frente o hacia atrás, girar, ampliar, posición de la cámara, y Walk Trought (Sirve para moverte por el mapa con las teclas de dirección)

De este menú cabe destacar por su importancia el punto cuarto y octavo, ya que se utilizaron repetidamente para la creación del entorno. A continuación en la tabla 5.1 se muestra el menú de este editor explicando sus diferentes opciones, para así facilitar las explicaciones posteriores.

File	Edit	Mode	Object	Te
New			Ctrl+N	
Open...			Ctrl+O	
Close				
Save			Ctrl+S	
Save As...				
Map Properties				
Add Script				
Build WMB...				
Run Level...				
Starter...				
Resource...				
Publish...				
Preferences				
Exit				

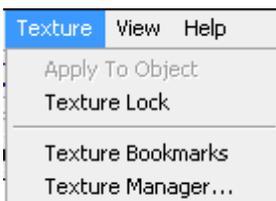
Edit	Mode	Object	Texture	Vie
Undo			Ctrl+Z	
Cut				
Copy			Ctrl+C	
Paste			Shift+Insert	
Select None				
Delete		Delete		
Duplicate		Ctrl+D		
Realign To Grid				
Hollow Block				
CSG Subtract				
Object Mode				▶
Find Object				

## File (Archivo)

- **New:** Empieza un nuevo documento en blanco.
- **Map properties:** Añade script (inteligencia) al juego, el color y posición del sol y la niebla.
- **Add Script:** Añade script al juego.
- **Build Wmp:** Compila el mapa
- **Run:** Ejecuta el juego
- **Starter y Publish:** Publica el juego y se crea un ejecutable del proyecto.
- **Preferences:** Sirve para configurar el programa.

## Edit (Editar)

- **Undo:** Atrás.
- **Cut, Copy y Paste:** Añade Cortar, copiar y pegar
- **Select None:** Deseleccionar.
- **Delete:** Eliminar
- **Duplicate:** Duplicar
- **Hollow Block:** Crea un bloque con dimensión interna.
- **CSG Subtract:** Sirve para hacer agujeros (Huecos de las puertas...)



## Object (Objetos)

- **Group:** Sirve para agrupar objetos, entidades o cualquier objeto, para agrupar se pulsa la G en cada objeto, y cuando terminas el grupo CTRL+G
- **Add Cube:** Añade un cubo. Da a elegir de 3 tamaños.
- **Add Hollow Cube:** Añade un cubo con dimensión interna
- **Add Primitive:** Añade objetos primitivos, como bases, conos..
- **Add Prefab:** Añade Objetos prefabricados.
- **Add Position:** Añade la cámara de inicio.
- **Add Light:** Añade una luz
- **Add Sound:** añade un sonido WAV, que se repite continuamente
- **Add Path:** añade un camino o recorrido.
- **Add Model:** añade un jugador, o cualquier otro modelo (Terreno, sprite, mapa, modelo...)
- **Add Terrain:** añade un terreno
- **Add Sprite:** añade una imagen.
- **Load Entity:** Carga una entidad (modelos, sprites, terrenos)
- **Load Prefab:** Carga un objeto prefabricado
- **Load Sound:** Carga un sonido.

## Texture (Texturas)

- **Apply to Object:** Aplica la textura en el objeto.
- **Textura Manager- Add Wad** (añade una textura en formato .WAD)

## . View ( Vistas )

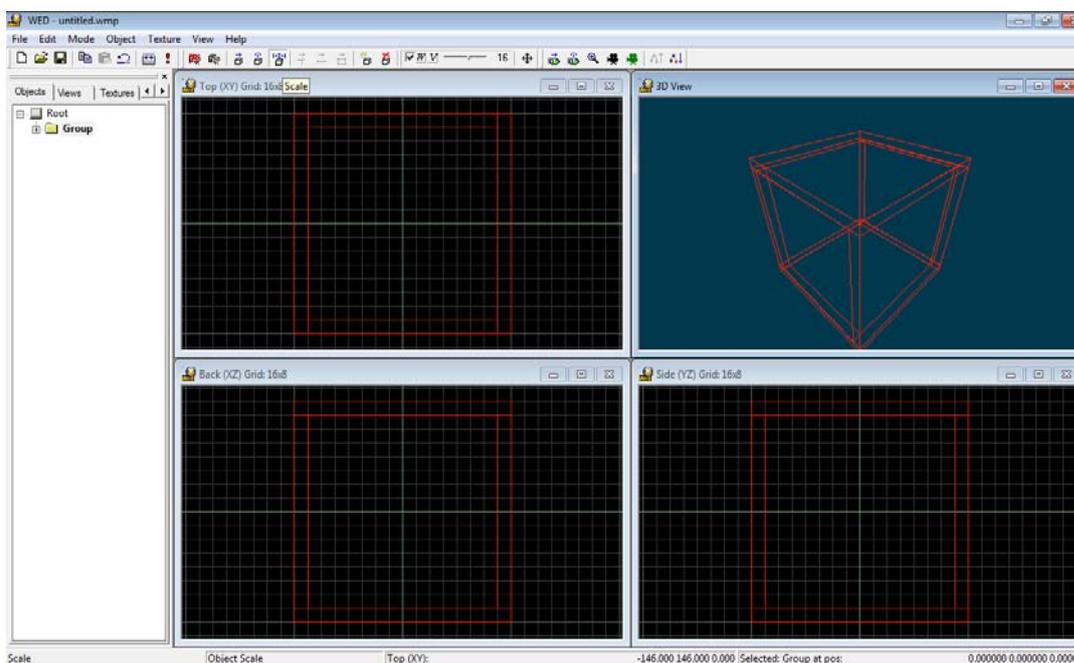
- **Textured:** Pulsamos ese boton para ver en la pantalla VIEW 3D nuestro escenario con sus texturas.

Tabla 5.1 Menú desplegable del level editor

## 5.1.2 Creación del escenario

Lo primero que se tuvo que hacer para construir la vivienda fue añadir un bloque, para lo cual se despliega el menú object y se selecciona la opción Add Hollow Cube, la cual añade un cubo con dimensión interna, es decir, un cubo hueco, como podemos observar en la *figura 5.3*.

Una vez se tiene el cubo únicamente hay que variar su tamaño alargándolo o ensanchándolo lo que se desee para obtener la forma buscada, ajustándolo mediante las tres vistas existentes, teniendo en cuenta que Top es el alzado, Back la planta y Side el perfil. Para poder hacer todo esto únicamente se debe pulsar el botón de scale tal y como muestra la *figura 5.3*, una vez se tiene esta opción seleccionada mediante el movimiento del ratón al tiempo que se mantiene pulsado el botón izquierdo se irá modificando la escala del objeto seleccionado.



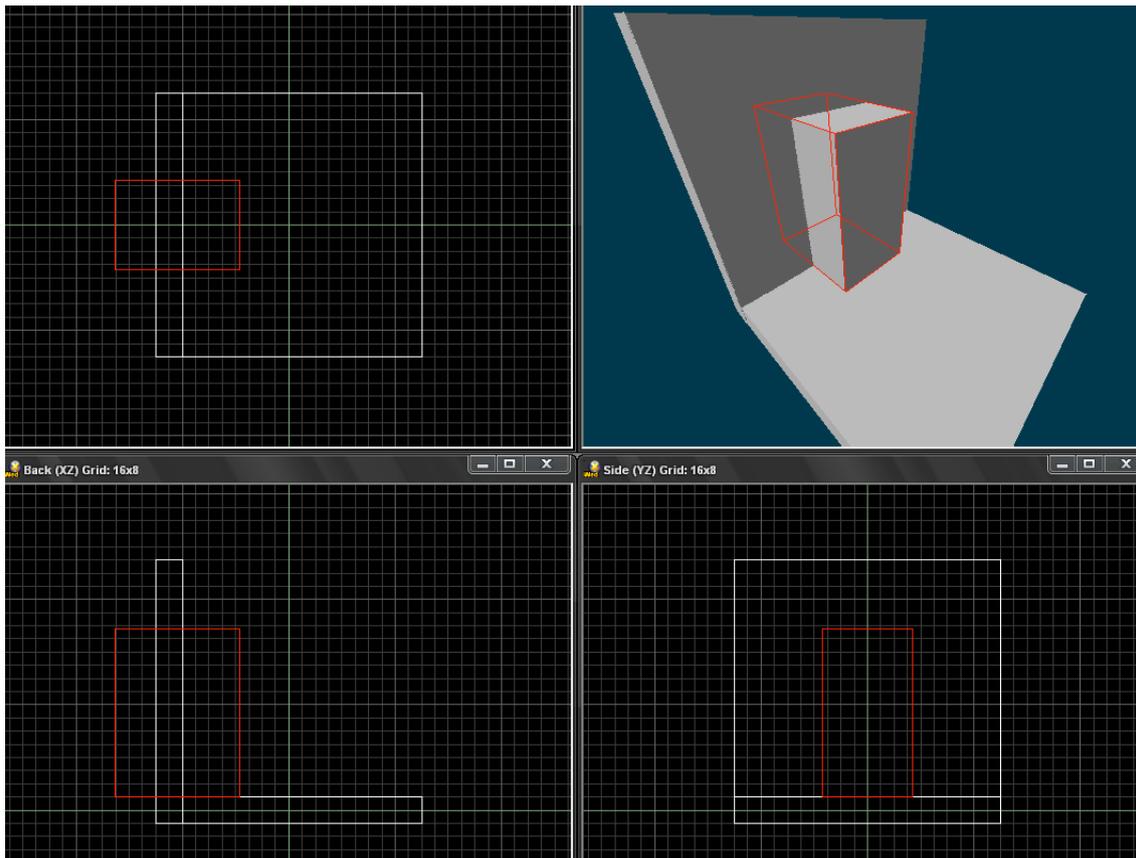
*Figura 5.3. Level editor con un bloque añadido*

Una vez las dimensiones y la forma del cubo son las deseadas, se procede a la creación de las paredes y el techo. Para ello se siguen exactamente los mismos pasos descritos anteriormente con la diferencia existente que en lugar de usar un cubo hueco se selecciona un cubo (add cube). La diferencia entre esta opción el el hollow cube radica en que el cubo es macizo mientras que, como se señaló anteriormente, el hollow cube es hueco.

Modificando las dimensiones de los cubos del mismo modo que se ha explicado, se forman tanto las paredes como el techo de la vivienda.

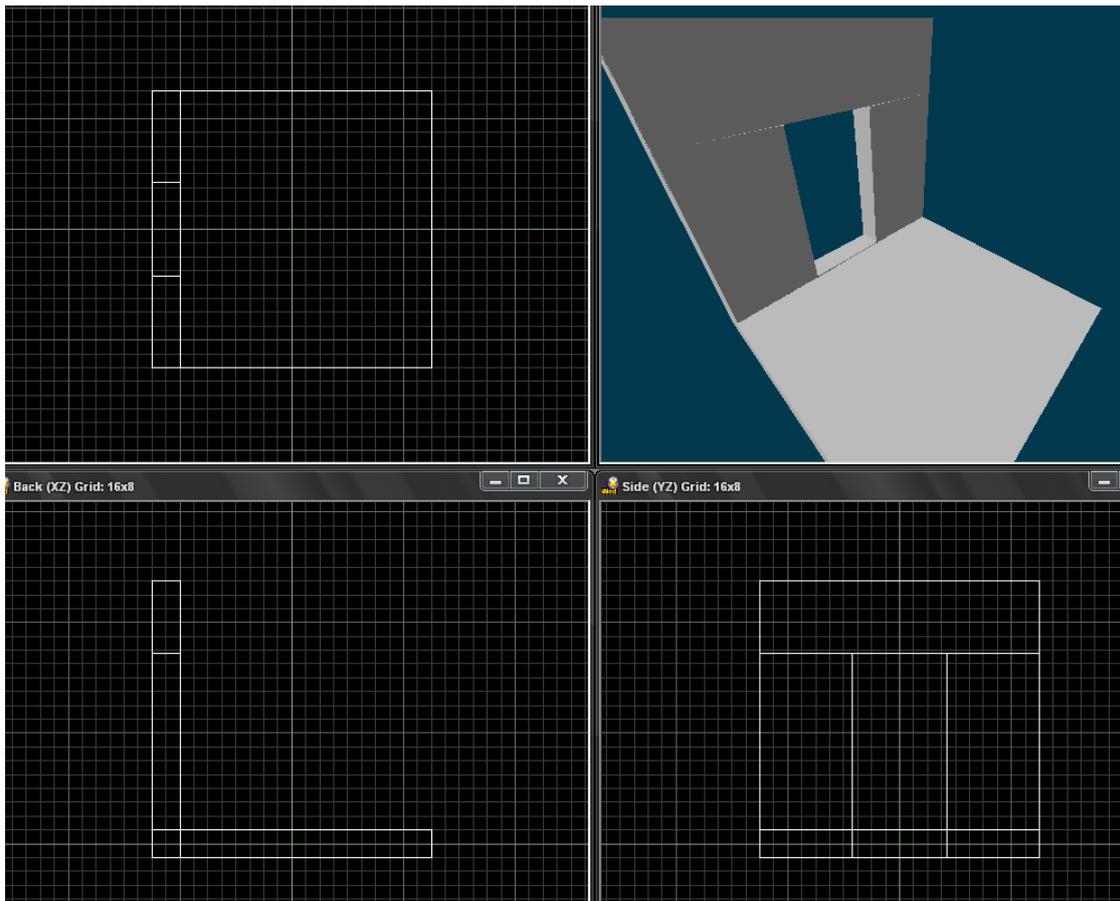
Una vez el esquema de la casa está creado, se procede a la creación tanto de las ventanas como de las puertas, para ello hay que hacer agujeros en las paredes y en el techo de la planta baja y de la primera planta, este hueco en los techos permitirá al personaje ascender a los pisos superiores con la implantación de una escalera.

Para hacer los agujeros en este programa se procede con la creación de un cubo que adaptamos según el tamaño del hueco desado, hay que observar las proporciones de las paredes para crear un agujero de un tamaño lógico. Una vez hecho esto el cubo debe situarse entre la pared en la que se desea hacer el hueco, asegurándose de que el cubo atraviesa totalmente la zona deseada para el agujero, tal y como muestra la *figura 5.4*.



*Figura 5.4. Posición del bloque para la creación de un hueco*

A continuación teniendo el cubo seleccionado, en el menú edit pulsamos CSG SUBSTRACT, que como se señaló en la *tabla 5.1* esta opción permite la creación de agujeros. Seguidamente se suprime el cubo seleccionado quedando lo mostrado en la *figura 5.5*. El procedimiento a la hora de crear una ventana o el hueco de la escalera, es exactamente el mismo, cabe señalar sin embargo, que a la hora de la realización de las ventanas se creó una forma determinada mezclando un cubo con una semiesfera, logrando así obtener una ventana con forma de arco.



*Figura 5.5. Creación de un hueco de puerta.*

Para la realización de las escaleras se optó por el uso de elementos prefabricados que vienen incluidos en el programa, concretamente se usó el elemento **Steps01.wmp**, el cual podemos observar en la siguiente imagen.

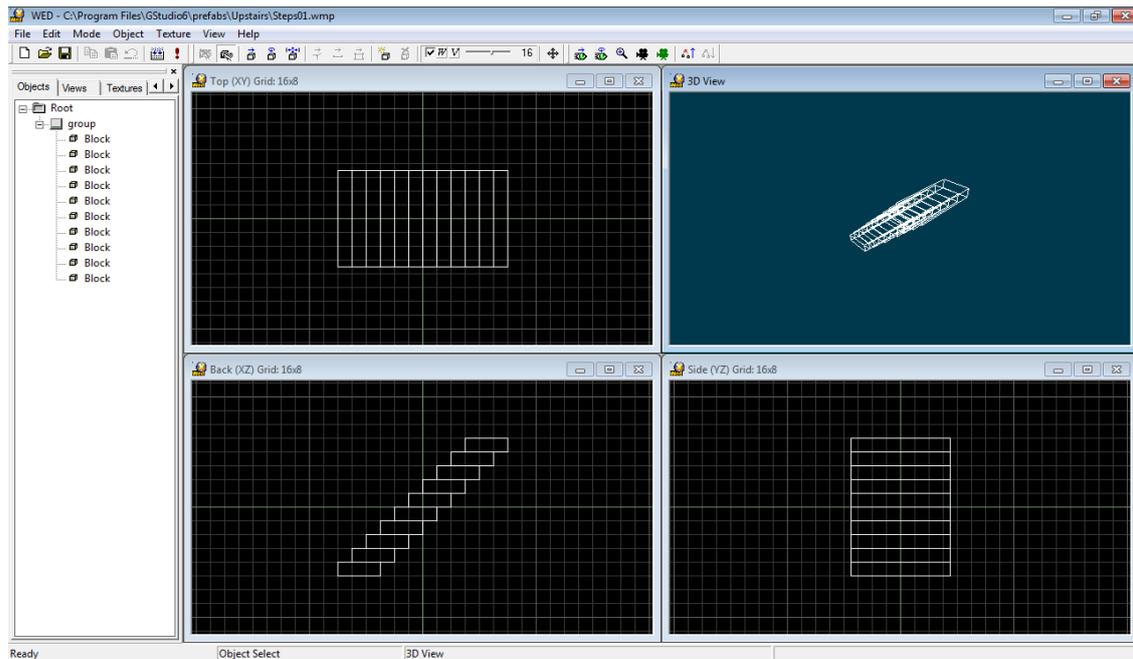


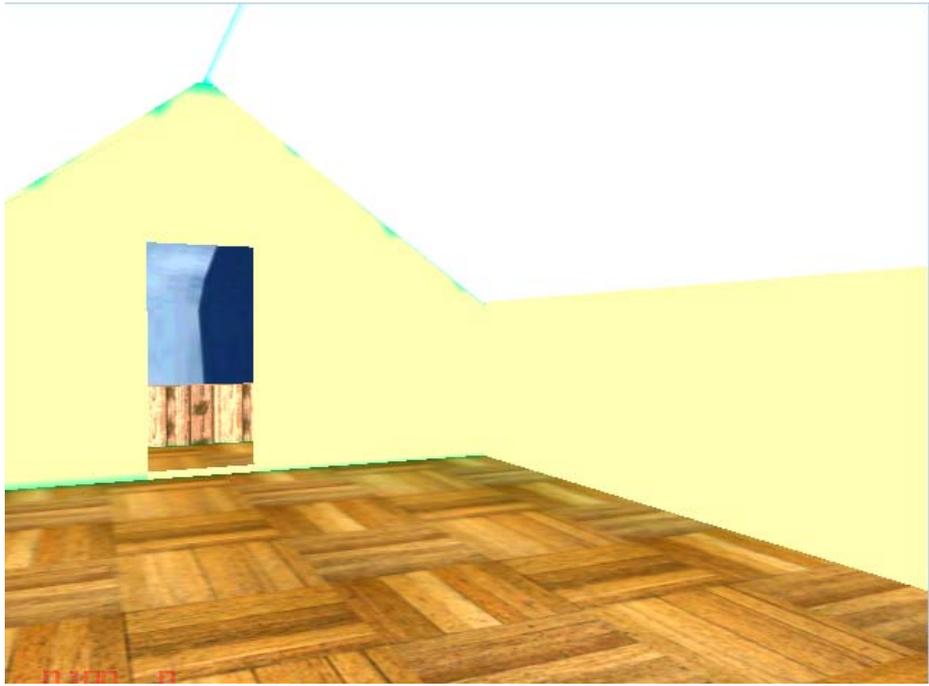
Figura 5.6. Elemento prefabricado Steps01.wmp

Al igual que con los elementos añadidos anteriormente se modificó su tamaño para ajustarlo a la vivienda.

### 5.1.3 Texturas

Una vez el escenario está creado, para conseguir una impresión más realista se procede a darle textura, es decir, a introducir imágenes del mundo real para conseguir de este modo que tanto las paredes como los suelos posean un aspecto lo más real posible.

Se procedió a crear texturas propias, ya que las presentadas por defecto en el programa, no satisfacían el objetivo deseado. Las texturas utilizadas se muestran en la *figura 5.7*. en donde se puede ver una habitación del entorno virtual realizado.



*Figura 5.7. Texturas empleadas en suelo y paredes*

Para realizar la textura deseada se debe seleccionar el bloque al que se desea aplicar dicha textura, entonces en el menú situado a la izquierda del programa se selecciona la pestaña de texturas y se escoge la textura que se desee aplicar.

Dado que un cubo presenta varias caras y no siempre se quiere aplicar la misma textura a todas las caras, como es el caso del suelo que separa la planta baja y la primera, donde por una cara del cubo se tiene el suelo mientras que la otra corresponde al techo. Para poder aplicar una textura diferente a cada cara del cubo se debe seleccionar el bloque y con el botón derecho del ratón, en propiedades nos aparece una ventana, tal y como se muestra en la *figura 5.8*, en la pestaña surface es posible seleccionar cada una de las caras del bloque y aplicarle la textura deseada en cada cara.

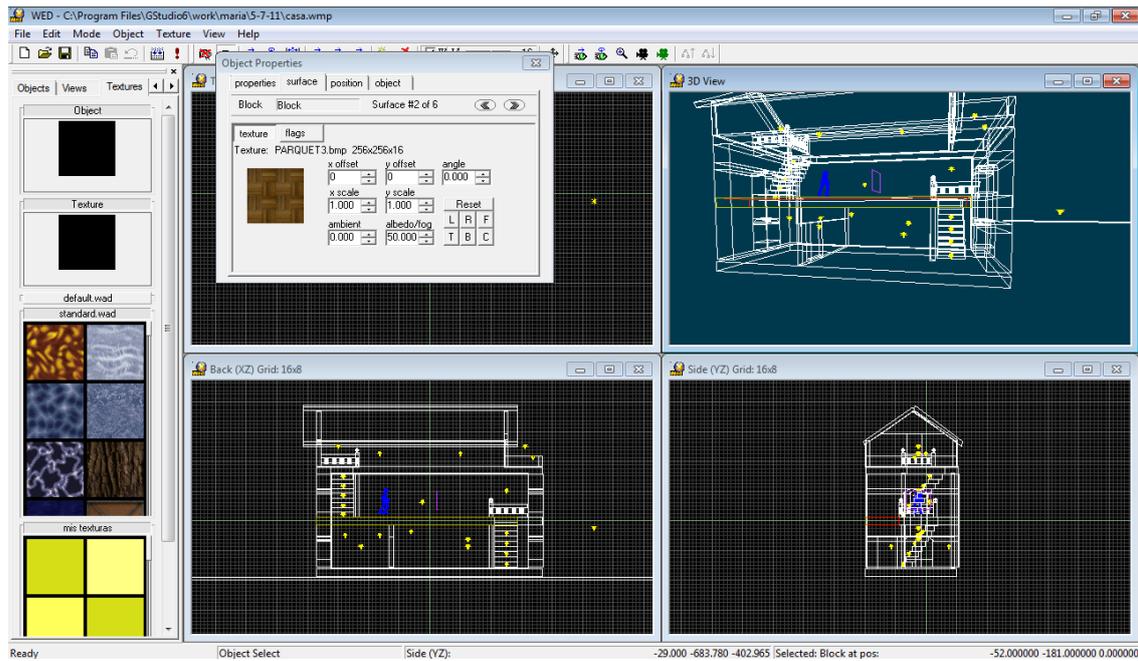


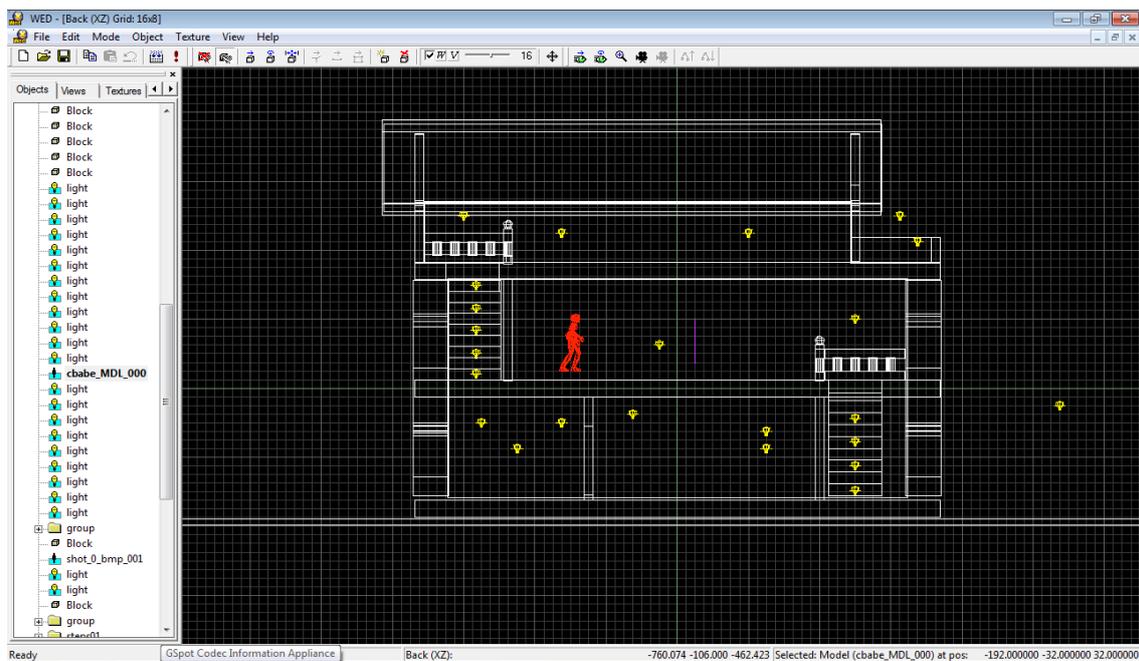
Figura 5.8. Texturas de la casa

En el caso del Hollow cube que conforma la estructura externa de la vivienda, nos encontramos con una dificultad diferente, puesto que como muestra el menú de la izquierda, en la pestaña Objet, aparece una carpeta Group. Esto indica que cada uno de los bloques que conforman el cubo hueco están unidos en un grupo, así pues, al aplicar una textura, ésta se aplicará al grupo entero sin posibilidad de variación.

Por tanto, para poder crear la textura buscada de la forma deseada, será necesaria la desintegración del grupo, para ello en el menú de la izquierda en la carpeta correspondiente al grupo a tratar, mediante el botón derecho se acciona el menú desplegable y se selecciona la opción ungroup (deshacer grupo). Con eso es posible la selección de cada uno de los componentes que conformaban el grupo y la aplicación de sus texturas, como se indicó anteriormente.

## 5.1.4 Creación de un modelo

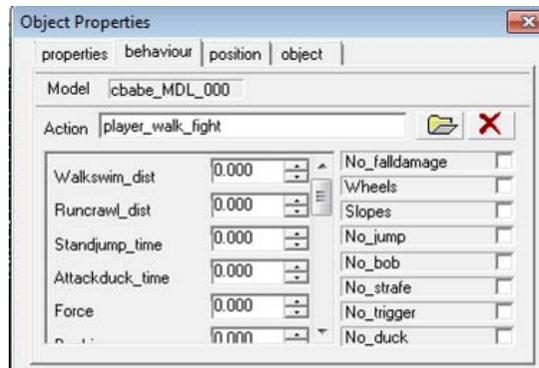
Una vez tenemos el escenario completo, quedará la creación de un modelo que se mueva e interactúe con el mismo. Para la creación de modelos en este programa se utiliza el editor Med, sin embargo, dado que el programa posee un conjunto de modelos prefabricados, se optó por la utilización de uno de ellos, puesto que la creación de un modelo propio no representa ninguna ventaja añadida, ni es el objetivo que se persigue en la realización de este proyecto. El modelo utilizado fue Cbade.mdl y su incorporación al mapa se realiza mediante el panel edit con la opción add model, buscando en la carpeta work.



*Figura 5.9. Personaje añadido a la vivienda*

Una vez el modelo está añadido hay que dotarlo de movimiento para poder manejarlo. Para ello en file y add script se selecciona A5\_template\_proyect, tras esto seleccionando al personaje, con el botón derecho en propiedades en la pestaña behaviour se

procede a deseleccionar todas las opciones marcadas, quedándose tal y como se muestra en la *figura 5.10*.



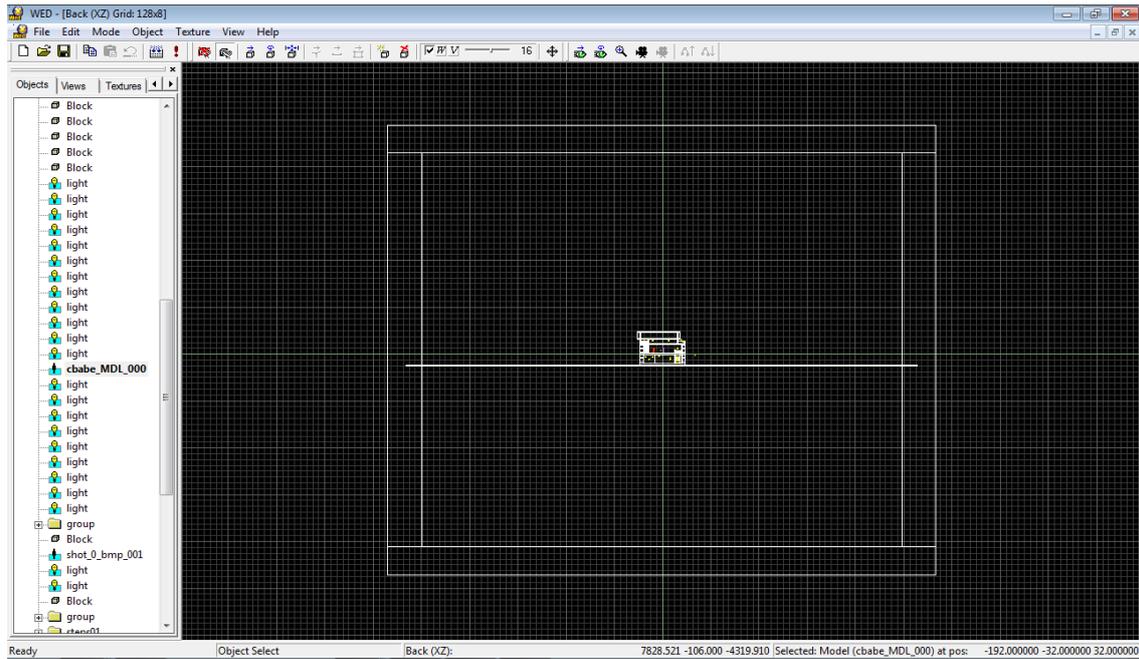
*Figura 5.10. Propiedades del personaje*

Con todo esto ya sería posible la compilación y ejecución del entorno.

### 5.1.5 Efectos exteriores

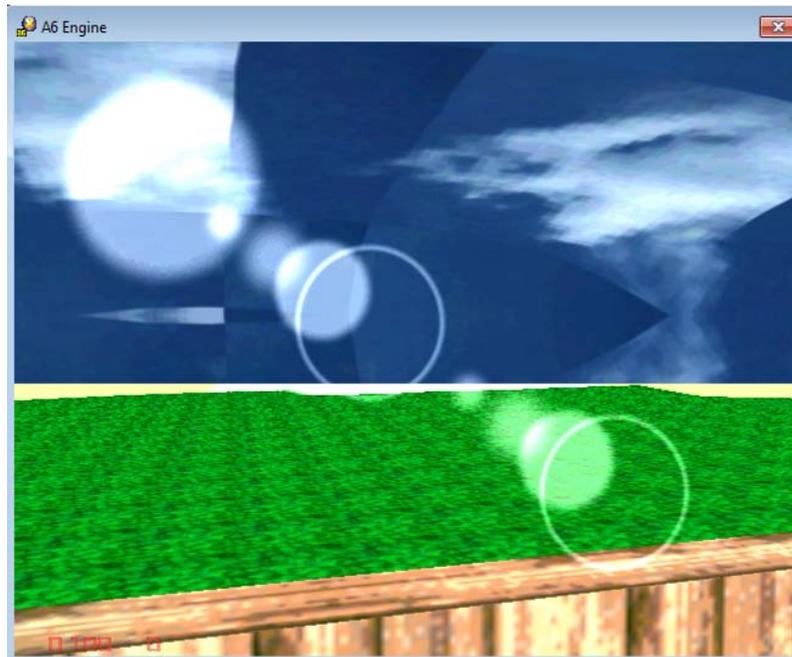
En este punto el escenario en el cual se trabaja, la vivienda, no se encuentra ubicada en ningún lugar, por lo que si el personaje es dirigido al balcón u observa por la ventana, se producirá una visión errónea, puesto que no hay nada que mostrar. Para evitar esto y lograr una visión más realista del proyecto, se debe crear lo que se conoce como cubo de cielo, es decir, un cubo hueco que representa un espacio real en el cual se ubica el escenario.

A la hora de la creación de este cubo se debe tener en cuenta que debe ser de un tamaño mucho mayor que el del escenario con el que trabajamos para poder lograr una sensación de horizonte, tal y como se muestra en la *figura 5.11*.



*Figura 5.11. Cubo de cielo y terrenos de la casa*

Como se observa en la imagen anterior no sólo se añadió un cubo de cielo, sino que además se creó mediante un bloque un suelo, que representará los terrenos o jardines de la vivienda, logrando así una sensación más realista, puesto que al salir al balcón se podrá observar que la casa está situada en un terreno real, tal y como se muestra en la *figura 5.12*.



*Figura 5.12. Visión desde el balcón con el programa en ejecución*

Como se observa en esta figura, se procedió a realizar la aplicación de la textura correspondiente tanto al cubo de cielo como al suelo, poniendo en este último una textura verdosa que emularía un jardín.

A la hora de dotar de textura al cielo se debe tener en cuenta que un cielo real posee un conjunto de efectos, tales como la inclinación del sol y el movimiento de las nubes. Para lograr estos efectos se selecciona una textura de cielo y en cada una de las caras del cubo en el panel surface/flags se selecciona la opción sky, la cual es la encargada de dar el efecto del movimiento del cielo. Por otro lado, como se observa en la *figura 5.12* el sol se encuentra en este escenario, para controlar esto tenemos en file/ map Properties en la pestaña Sun existen varias opciones, tal y como se ve en la *figura 5.13*, las cuales se explican a continuación.

- En la zona Sun: se pueden añadir unos valores numéricos o escoger el color del sol.
- Ambient: con esta opción es posible añadir valores numéricos o escoger el color de la luz del ambiente producida por el sol

- Sun elevation: podemos controlar la elevación desada del sol
- Sun azimuth: permite el control del azimuth del sol.

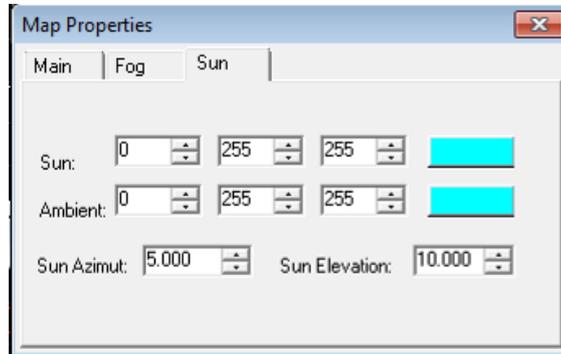


Figura 5.13. Propiedades del sol de este proyecto

Con todo esto el diseño del entorno quedaría finalizado, quedando para llegar al objetivo de este proyecto la interconexión con el exterior, es decir, la programación realizada para que el personaje pueda controlar la vivienda domotizada mediante una página web creada, con la cual es posible el control de la vivienda a través de internet, la creación de dicha página, así como sus características se explicará en el siguiente capítulo.

A continuación pasaremos a explicar cómo se realizó la comunicación con el exterior mediante el Script Editor.

## 5.2 Programación de las comunicaciones externas. Script Editor

Puesto que el objetivo de este proyecto radica en la interacción del usuario mediante el entorno virtual para un control de la vivienda domótica, se hace necesaria una comunicación vía internet para poder controlar la casa domotizada.

Para lograr que el personaje pueda acceder a internet es necesario la creación de un objeto en el entorno que posea la cualidad de conectarse con un explorador para así lograr el acceso a la red.

Como se explicó en el capítulo anterior, este game engine permite la creación de unas funciones específicas llamadas acciones, las cuales crean la posibilidad de una interacción del personaje con un objeto, es decir, permiten que el personaje pueda pulsar un objeto y que ello desencadene una acción real, tal como podría ser la apertura de una puerta, la iluminación de una habitación, la apertura de un video o del mismo explorador de internet.

Así pues deberemos crear un objeto dentro de la plataforma virtual que permita la implementación de una acción para poder realizar una comunicación con el exterior.

En este apartado se procederá a explicar cómo se creó el objeto en cuestión, así como el código usado para realizar la llamada al explorador.

### **5.2.1 Creación de la entidad**

Para crear un objeto que sirva de enlace entre el personaje y el mundo exterior, debe permitir la asociación de código, es decir, ha de ser un objeto al que sea posible insertarle un código.

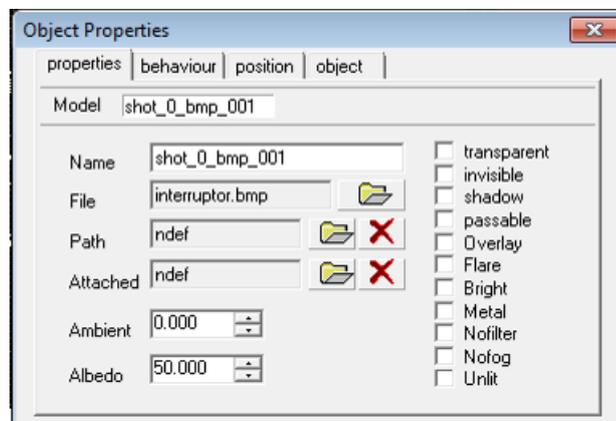
A diferencia de lo que se ha realizado hasta ahora, todos los objetos que se han creado han sido de decorado, es decir, se ha creado el escenario el lugar por donde el personaje se mueve, pero ninguno de los objetos creados tiene mayor función que la de mostrar un escenario. A excepción del personaje el resto de los objetos son estáticos y no es posible asociarles un código, puesto que se han creado como simple decoración o escenario.

Sin embargo ahora es necesaria la creación de un objeto con el cual sea posible realizar una interacción y por tanto, al que se le pueda asociar un código que permita el desarrollo de una acción. Para ello sería posible la utilización de un modelo med, puesto que este tipo de modelos, se crean para interactuar con el personaje que se maneja.

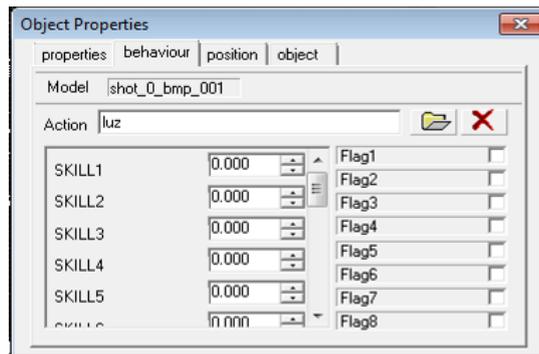
Aunque esta opción es muy válida, en este proyecto se optó por otra vía mucho más simple y en los efectos prácticos exactamente igual de efectiva, esto es la utilización de un sprite, que como explicamos en el capítulo anterior no es más que gráficos 2D creados con un programa de dibujo.

La ventaja de la utilización del sprite frente a un modelo es un menor coste gráfico, dado que el interés que tenemos en la creación del objeto es sencillamente la de un panel domótico es posible usar una entidad de sprite orientado para emularlo y lograr un menor consumo de tarjeta gráfica.

Para añadir el sprite se ha de ir al level editor en el menú **Object/Add Sprite** añadiéndose un cuadrado, el cual si es seleccionado con el botón derecho en propiedades aparece la ventana mostrada en la *figura 5.14*, en la cual es posible observar que una de las características que permite esta entidad es la de introducirle una imagen a mostrar. Por otro lado en la pestaña behaviour se selecciona la acción que se le desea aplicar, en este proyecto se le asoció la acción luz como se muestra en la *figura 5.15*, cuyo código se explicará en el siguiente apartado.



*Figura 5.14. Propiedades del sprite*



*Figura 5.15. Asociación de la acción al sprite*

Una vez asociada la acción al sprite, tal y como muestra la imagen anterior, cuando el usuario pique sobre el sprite se ejecutará la acción que se haya implementado. A continuación se explicará la implementación de dicha acción, desarrollada en el script editor.

## 5.2.2 Programación de la acción

Como se explicó en el capítulo anterior una acción es una función especial que se puede unir a una entidad: un jugador, un interruptor, etc.

A diferencia de una función una acción no necesita ser llamada, necesita ser unida a una entidad, es decir, mientras que a una función se le pasan una serie de parámetros para que realice la operación que deseemos, una acción es asociada a una determinada entidad lo cual le permite realizar diferentes operaciones.

A la hora de crear la acción se optó por añadirla al final del código del elemento dors.wdl, situado en la lista de elementos de la función principal del código de la plataforma virtual. No existe un motivo concreto para esta elección, ya que como se ha comentado las acciones son un tipo especial de funciones, y por tanto pueden escribirse en cualquier parte del código.

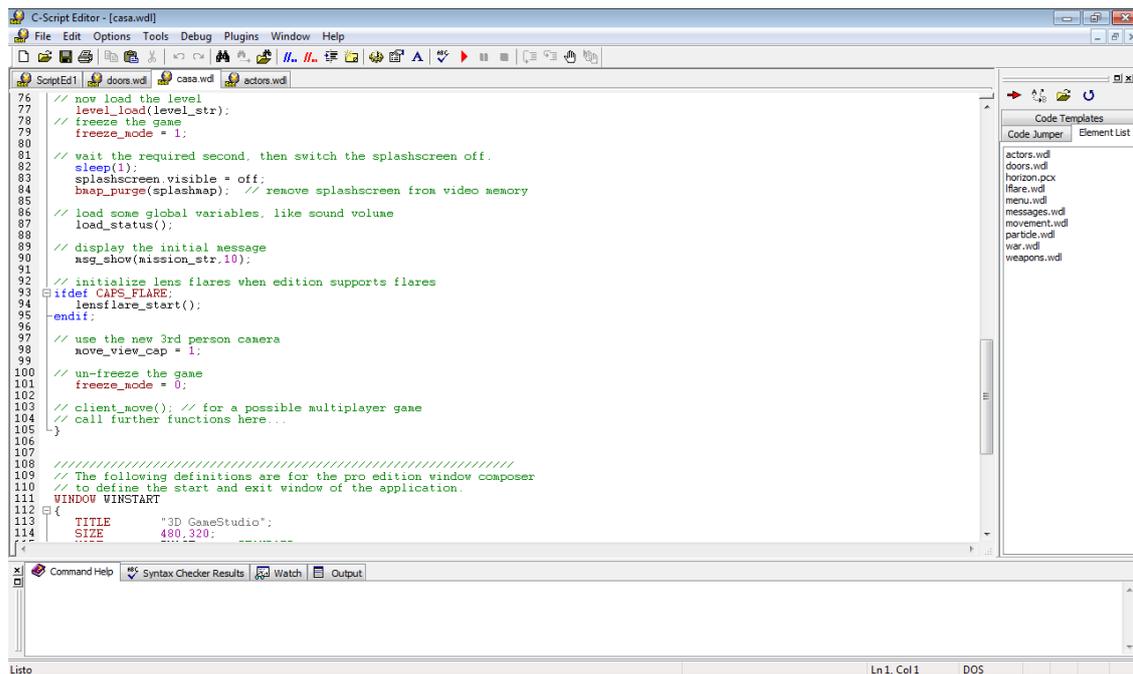


Figura 5.16. Función principal del código y lista de elementos

Al final del código del elemento doors.wdl se añadieron una serie de acciones que realizan una llamada a una función que responde por abrir, y la cual permite la ejecución de un archivo específico encargado de la transmisión de la información necesaria para el control del aparato domótico asociado a la acción concreta. Un ejemplo tanto de esta función como de la acción se encuentran representadas en la *figura 5.17* y a continuación se transcriben diversas acciones implementadas, junto con su función correspondiente explicándose su funcionamiento.

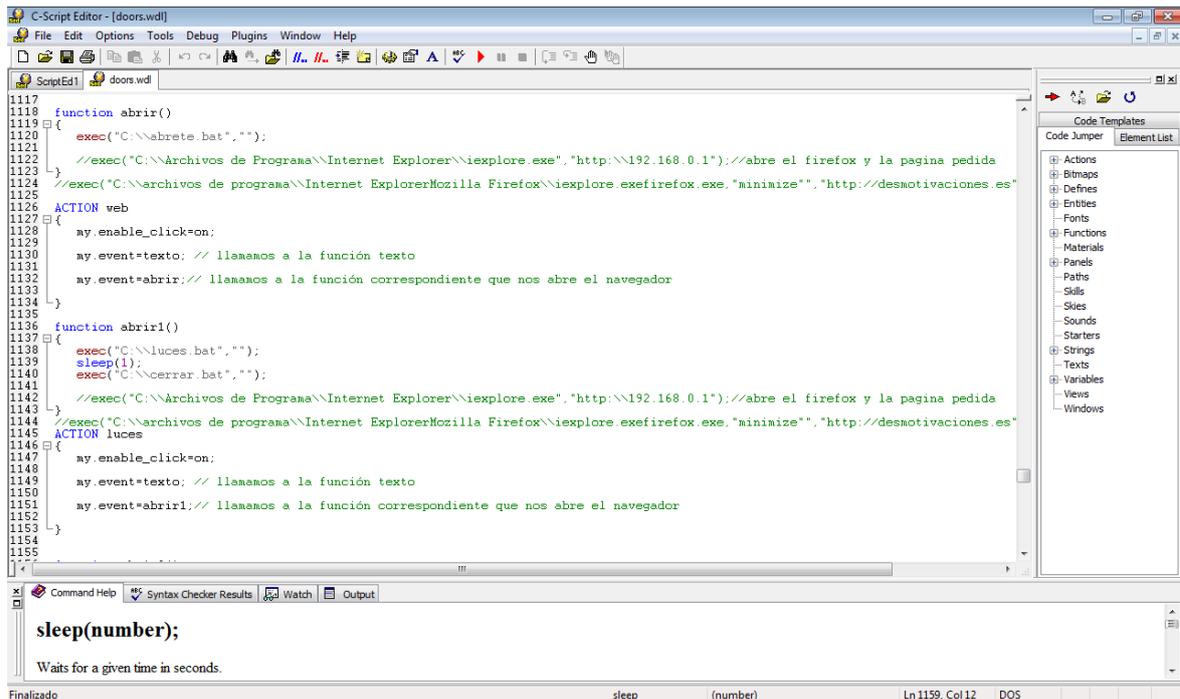


Figura 5.17. Función abrir y acción luz

La acción luz será la que se asocie al sprite añadido al entorno como se mostró en el punto anterior, esta acción recoge el click del ratón y realiza una llamada a la función abrir, la cual consiste en una ejecución de un archivo específico que permite abrir el explorador de internet en una página concreta, la cual se corresponde con la página de control del interfaz web creado para el control del sistema domótico.

A continuación se muestra el código añadido para la apertura de la página web de control del sistema domótico y su explicación:

```
function abrir()
{
    exec("C:\\abrete.bat",""); //ejecuta un bat que abre el explorador.
}
ACTION web
{
    my.enable_click=on //recogemos el click del ratón
    my.event=abrir; // llamamos a la función correspondiente que nos abre el navegador
}
```

Al igual que esta acción, se procedió a la implementación de otras acciones, encargadas del control de cada elemento domótico en concreto, colocando, por tanto, un interruptor para cada acción deseada del elemento. La explicación de los comandos utilizados para el control de los elementos se detalla en el siguiente capítulo. A continuación se muestra el código implementado de algunas de las diferentes acciones.

```
function abrir1()
{
    exec("C:\\luces.bat",""); //ejecuta un bat que abre el explorador.
    Sleep (1)
    exec("C:\\cerrar.bat",""); //ejecuta un bat que cierra el explorador
}
ACTION luces
{
    my.enable_click=on //recogemos el click del ratón
    my.event=abrir1; // llamamos a la función correspondiente que nos abre el navegador
}
```

```
function abrir3()
{
exec("C:\\dimmeron.bat",""); //ejecuta un bat que abre el explorador.
Sleep (1)
exec("C:\\cerrar.bat",""); //ejecuta un bat que cierra el explorador
}
ACTION dimmeron
{
my.enable_click=on //recogemos el click del ratón
my.event=abrir3; // llamamos a la función correspondiente que nos abre el navegador
}
```

```
function abrir8()
{
exec("C:\\persianaspcarriba.bat",""); //ejecuta un bat que abre el explorador.
Sleep (1)
exec("C:\\cerrar.bat",""); //ejecuta un bat que cierra el explorador
}
ACTION persianaspcarriba
{
my.enable_click=on //recogemos el click del ratón
my.event=abrir8; // llamamos a la función correspondiente que nos abre el navegador
}
```

Como podemos observar, el funcionamiento del código es idéntico, exceptuando el archivo al que se llama, puesto que cada aplicación a controlar tiene un archivo de llamada específico, debido a lo cual se han puesto únicamente tres funciones para mostrar el funcionamiento, siendo el resto idénticas a éstas. Así pues, con el correcto funcionamiento este código mencionado se da por finalizada la construcción del entorno virtual.

## Capítulo 6

# Diseño e implementación del sistema domótico

*“¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida más fácil, nos aporta tan poca felicidad? La respuesta es está, simplemente: porque aún no hemos aprendido a usarla con tino”*  
*Albert Einstein 1879-1955. Científico Alemán.*

Poco a poco hemos visto como nuestro entorno ha evolucionado de una forma casi exponencial, cosas que pocos años atrás se consideraban pura ciencia ficción las tenemos hoy en nuestro salón, así pues, no es de extrañar que el siguiente paso provenga de las tecnologías de la comunicación.

La comunicación es algo básico para el ser humano, debido a esto, cada vez se van creando más aparatos con esta posibilidad, la de la comunicación. Hoy en día no es de extrañar tener un televisor con el que podamos conectarnos a internet, o poder realizar llamadas IP con nuestro teléfono, internet está a nuestro alrededor permitiéndonos esa ansiada necesidad de comunicación con el mundo que nos rodea.

Sin embargo la tecnología no se ha detenido ahí, sigue avanzando, ofreciéndonos no sólo la posibilidad de comunicarnos con las personas que deseemos, sino también permitiéndonos comunicar unos aparatos con otros y logrando así hacer realidad situaciones, tales como encender la chimenea de nuestro hogar moviendo la nariz, que antes únicamente eran posibles en películas de fantasía.

Hoy por hoy los sistemas domóticos evolucionan hacia la integración de protocolos de comunicación domésticos, buscando en internet su integración para la automatización de funciones domésticas y el tratamiento de la información.

## **6.1 Integración de internet en la domótica**

La EIBA, el organismo destinado a hacer del EIB un estándar para cualquier desarrollo europeo en domótica, ha desarrollado un nuevo protocolo para conectar las redes EIB con Intranets e Internet. Se definen estándares abiertos para desarrollos de terceras partes, y es compatible con protocolos USB, OPC, SOAP, Ethernet, Gigabit Ethernet, redes locales inalámbricas, conexiones por módem bajo protocolo PPP, RDSI, ADSL, TV entre otras especificaciones. Nosotros únicamente trataremos sobre el sistema de interconexión mediante el protocolo TCP/IP.

### **6.1.1 TCP/IP no está optimizado para la Domótica**

A pesar de su uso y sus múltiples aplicaciones debemos destacar que la mayoría de los protocolos que han sido especialmente creados para implementar redes de control distribuidas (Lonworks, EIB, EHS, X-10), las tramas fueron diseñadas de forma que el espacio útil para datos de las aplicaciones fuera el máximo. Por ejemplo para encender y apagar una luz basta con una orden codificada en un par de octetos. Por lo tanto, se trata de minimizar los campos de control (direcciones, CRCs, etc) que el protocolo necesita para transferir estos dos octetos al dispositivo destino [21].

Existen dos formas para medir este factor de mérito:

1. Ancho de Banda Neto, medido en bits por segundo. Por ejemplo, un bus puede inyectar un flujo de datos de 5400 bps en los cables pero sólo 4800 bps son útiles para la aplicación. En este caso, 600 bps serían añadidos por el propio protocolo.
2. Overhead o tara, medido en tanto por ciento. Por ejemplo, de una trama 55 bytes, son útiles 50 bytes. La tara sería del 10%.

Como podemos ver ambos parámetros representan cuantos bits puedo transferir por el medio físico en un instante dado y cuantos bits son realmente útiles para la aplicación.

Por el contrario, el protocolo TCP/IP, que fue diseñado para transferir una ingente cantidad de datos entre dos máquinas, cuando se usa para transferir 2 o 3 octetos de información el coste en ancho de banda es muy alto. Por ejemplo, entrando directamente a nivel IP, el paquete mínimo necesita 20 octetos sólo de campos de control. Si entramos a nivel de TCP son 40 octetos de control como mínimo. Dando lugar a una tara del 900% y 1800 %, respectivamente

Por este motivo, el protocolo TCP/IP no está optimizado para su uso en redes de control distribuido en aplicaciones de automatización de viviendas u oficinas, sin embargo, a pesar de las razones técnicas que se acaban de mencionar, TCP/IP está siendo usado en infinidad de ordenadores y aplicaciones, de forma que ha conseguido un volumen de negocio tal que ha hecho de este protocolo la herramienta ideal para asegurar la interconectividad total entre máquinas en cualquier parte del mundo.

Seguramente estos sistemas son el top 10 de control domótico, desde la oficina, el coche, desde casa de un amigo o cualquier lugar donde tengamos conexión vía internet. Esto permite un mayor control de la vivienda del ya existente ahora mismo, encender la calefacción o el aire acondicionado, programar el sistema de riego, controlar todos los electrodomésticos, el estado de las luces, persianas, toldos, y un largo etcétera, todo puede ser controlado a distancia.

## 6.2 Pasarelas IP

Una pasarela IP es una puerta de enlace que permitirá lograr una comunicación y control de las diferentes tecnologías que compone nuestra vivienda mediante tecnología TCP/IP.

Gracias al uso de estas pasarelas y el uso de la tecnología TCP/IP el control de nuestra vivienda pasa de ser un sueño a una realidad actual, lo cual nos demuestra que el pensar que la domótica es el futuro no sería más que vivir en el pasado.

Dentro del mercado tenemos un amplio abanico de opciones para realizar nuestra conexión a la red, en este apartado vamos a ver brevemente alguna de esas opciones.

### 6.2.1 Combridge IPAS

Como pasarela TCP/IP, es un hardware que puede utilizarse en combinación con diferente software (todos ellos de adquisición por separado) para la realización de diversos tipos de servicios Web.

Incluye licencia de utilización para 1 pasarela y 1 usuario. Se compone de dos programas distintos [22]:

- **ComBridge Studio Configuration Manager:** Para la administración de una o varias pasarelas y clientes TCP.
- **ComBridge Studio InfoPoint Configurator:** Para importar las direcciones de grupo de un proyecto creado en el ETS, crear / configurar / visualizar Java applets que hacen las funciones de visualización tipo tabla con botones.

## 6.2.2 Falcon

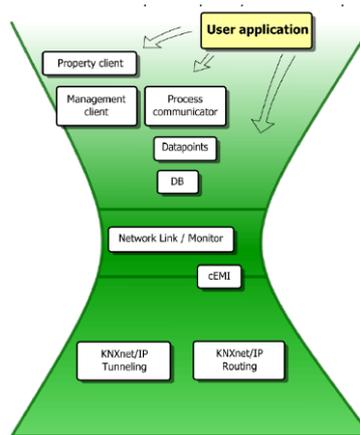
Las librerías Falcon son una herramienta extraordinaria, sobre todo para los programadores. Sienta las mejores bases para el acceso a KNX. Además de un sencillo API, ofrece una gestión del acceso para componentes Bus, para telegramas (direcciones de grupo), direcciones físicas, estados de los componentes y mucho más. El acceso al Bus posibilita la programación de lenguajes como Visual C++, Delphi, Visual Basic y otros. Falcon se utiliza como librería de acceso de red para ETS 3 y EITT. A partir de la versión 1.23, Falcon ofrece junto al IConnection Manager un interfaz de usuario fácil de usar por el integrador para crear y configurar la conexión entre KNX y otros medios [12].

## 6.2.3 Calimero

La librería Calimero no es más que una colección de APIs de código abierto escritas en Java que permite una fácil comunicación sobre sistemas EIB/KNX, incluyendo acceso remoto y control. Fue presentada por primera vez al público en la KNX Scientific Conference 2005 como parte del proyecto KNXLive!.

El software Calimero cierra la brecha que hay entre la aplicación y el medio KNX. Para conseguirlo se puede hacer con una variedad de protocolos, como a través de una interfaz de red IP o mediante una conexión por cable serie a un acoplador de bus. Además, la red KNX puede utilizar diferentes medios de comunicación, que requieren formatos específicos adecuados medio [24].

Calimero añade una interfaz abstracta bien definida que encapsula y oculta este tipo de diferencias. Como consecuencia, todo el diseño de Calimero sigue una arquitectura conocida como la arquitectura waist-line. La *figura 6.1* muestra cómo, en esta arquitectura, la funcionalidad se divide en tres capas. Las implementaciones de los servicios básicos, en nuestro caso la gama de protocolos de acceso a la red, están ubicados en la parte inferior. En el medio, el enlace de red ofrece una interfaz homogénea y estándar para la comunicación con las redes KNX, ocultando el servicio básico elegido.



*Figura 6.1 Arquitectura waist-line*

Este enlace es considerado la “cintura” de la arquitectura. En la capa superior encontramos los servicios de alto nivel para comunicarse con el medio KNX.

Calimero sigue un enfoque de capas no estricta. El usuario no está obligado a interactuar con las clases del nivel superior. Así, un usuario puede elegir el nivel deseado de la abstracción de la API.

La última versión liberada es la Calimero 2.0 alfa 4.

#### **6.2.4 Pasarelas IP comercializadas por KNX**

Los fabricantes de componentes KNX también han desarrollado sus soluciones particulares de pasarelas IP. Fueron diseñadas con el propósito inicial de funcionar como acopladores de línea o área de la red KNX, de manera que pudiese utilizarse las redes IP en lugar del cableado KNX estándar para unir varias instalaciones KNX dentro de una misma red. Pero también, se les añadieron funcionalidades para acceso y control de la instalación KNX desde un PC, como servidores web o ftp, para darles mayor aplicabilidad y valor añadido al producto. A continuación se muestran algunos ejemplos.

### 6.2.4.1 Central IP de Jung

La unidad central IP es la interfaz entre una red Ethernet (red LAN = Local Area) y el KNX / EIB. Con la ayuda de una conexión Ethernet, el usuario tiene acceso a su sistema inteligente de gestión del edificio a través de un PC local de su LAN o vía Internet. La conexión con la Internet no sólo puede ser establecida a través de una LAN, sino también con un módem analógico. La unidad central IP por lo tanto se puede integrar fácilmente en su casa nueva o ya existente o de las redes de oficinas.

La unidad central IP actúa como un servidor web y puede ser operado cómodamente desde un navegador como control, información y control de la unidad [25].

### 6.2.4.2 NK2 de Lingg&Janke

La pasarela IP con webserver NK2 conecta el Bus KNX con una red IP. Pone a disposición del instalador un servidor HTTP, un servidor FTP y una visualización con 104 puntos de datos. La visualización integrada puede abrirse directamente desde un navegador web estándar sin configuraciones adicionales.

Todos los ajustes necesarios se realizan mediante el ETS (Direcciones de grupo / tipos de datos / direcciones IP), así como mediante un navegador en la propia visualización (denominación de las teclas / denominación de las páginas / direcciones de hipervínculo).

Mediante una conexión externa de la red a Internet (router) puede controlarse también el NK2 con acceso a Internet. La posibilidad de introducir una contraseña evitará accesos indeseados [23 ,27].

En el servidor FTP integrado puede guardarse, por ejemplo, el archivo del proyecto del ETS correspondiente. Así, el programador podrá acceder siempre al archivo actual del proyecto desde cualquier lugar.

Tras evaluar las diferentes posibilidades que nos ofrecía el mercado optamos por trabajar con la plataforma TCP/IP NK2 de Lingg&Janke, esta decisión se basó en el hecho

que las librerías Calimero no se encuentran actualizadas y por otro lado las librerías Falcon, Central IP y Combridge son de pago, optando de este modo por una solución más económica y optimizada.

### **6.3 Implementación del sistema domótico**

En este proyecto se realizó una implementación domótica formada por reguladores de incandescencia, regulador de persianas, y controladores de luz mediante un sistema de entrada/ salida binaria. Esta implementación es clasificable como básica en una vivienda, puesto que permite el control tanto de la iluminación como de las persianas pudiendo así emplear simulación de presencia.

Estos dispositivos se encuentran conectados mediante una topología de línea, donde cada uno de los elementos se conectan a una línea principal a través de la cual va toda la información. La unión de los dispositivos se ha realizado mediante el uso de un entrenador domótico, sin que por ello se pierda generalidad ni utilidad, así pues este entrenador fue utilizado para comprobar el funcionamiento del sistema como si de una vivienda real se tratara, siendo la conexión de sus dispositivos la que se explica a continuación:

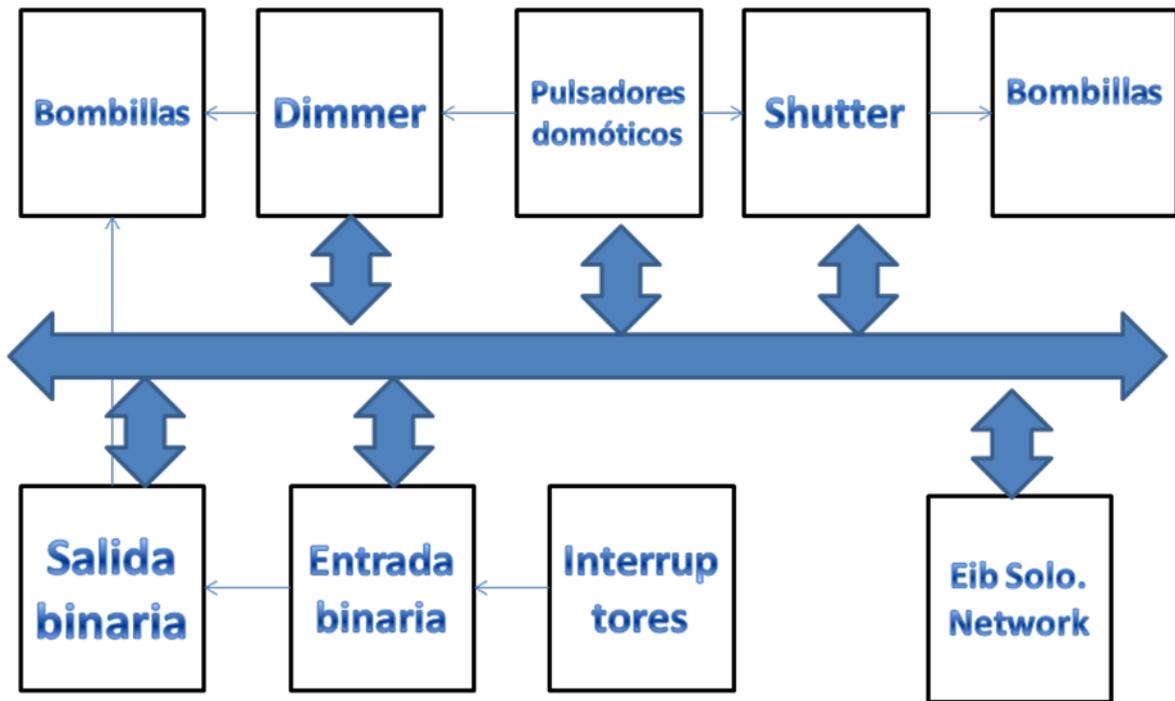


Figura 6.2 Esquema del sistema domótico implementado

En primer lugar existe una fuente de alimentación encargada de transmitir corriente a todos los aparatos, por otro lado hay dos interruptores se encuentran conectados a una entrada binaria que a su vez se haya unida a una salida binaria que llega finalmente a una bombilla.

Hay también un pulsador encargado del control de las persianas y del regulador de incandescencia, así pues se encuentra conectado tanto al dimmer como al shutter, el primero llega hasta una bombilla que se encenderá según la graduación que le indique el dimmer, el segundo está conectado a dos bombillas que representarán las persianas, tanto la opción de subir, bombilla superior, como la de bajar, bombilla inferior, dando las posibilidades de subida total, se iluminará totalmente la bombilla, o parada en seco, apagará la bombilla.

Para la configuración de todo este sistema se utilizó el software ETS el cual dispone de una serie de librerías, que se deben cargar previamente, con los diferentes modelos a utilizar. Gracias a este sistema una vez compramos un modelo podemos descargarlos su librería correspondiente e introducirla en el programa, de tal manera que hace posible la

programación del dispositivo en cuestión así como su conexión con el resto de dispositivos de la instalación.

Al introducir un dispositivo nuevo en el proyecto a crear, el ETS te genera automáticamente una dirección física al dispositivo, es posible también, cambiarla de forma manual siempre que sea una dirección libre.

Una vez se encuentran todos los dispositivos unidos se procede a la creación de los grupos, teniendo en cuenta que los componentes de cada grupo deben ser del mismo tamaño (1 bit, 2bit, 1 byte, etc.). Los dispositivos se unen entre sí mediante las diferentes direcciones de grupo que se les van asociando, para esto el programa posee unos menús desplegables que facilitan la visualización de los diferentes grupos y sus correspondientes direcciones de grupo, además de otro menú que muestra los diferentes dispositivos de nuestra instalación con su correspondiente dirección física así como la dirección de grupo a la que están asociados.

En este apartado se tratará de explicar los diferentes componentes utilizados en la domotización de la vivienda, así como su programación y puesta en funcionamiento.

### **6.3.1 Componentes**

En este apartado se explicarán los diferentes componentes a utilizar en el desarrollo de este proyecto así como las principales características que se han empleado de los mismos.

#### **6.3.1.1 Dimmer**

El dimmer o el regulador de incandescencia es un aparato que permite controlar la intensidad con la cual se desea iluminar una habitación, consiguiendo con ello un ahorro considerable al poder utilizar una determinada intensidad según la claridad que exista en la habitación.

En este proyecto se ha utilizado el Luzen Plus Universal Dimmer ZN1DI-P400 de Zennio, que presenta un amplio abanico de posibilidades tales como el encendido o apagado total de la luz, su encendido progresivo, pudiendo además seleccionar el porcentaje con el que se desee ascender (10%, 20%, etc), temporizadores que permitirán no sólo realizar un encendido y apagado, sino también una temporización intermitente, es decir, se llevará a cabo una secuencia a la salida de tipo ON OFF ON OFF. También permite la creación de escenas y secuencias pudiendo con todo ello realizar simulaciones de presencia más realistas.

Durante la realización de este PFC se ha utilizado una regulación simple en dos canales, uno de ellos con un encendido y apagado total mientras que en el segundo canal se ha realizado de tipo graduado

### 6.3.1.2 Shutter

El shutter, o en español, obturador, es un aparato con el cual se puede abrir o cerrar una abertura, en mayor o menor medida, permitiendo así el paso de la cantidad de luz deseada. Esto aplicado en una persiana, es lo que comúnmente se conoce como controlador de persianas, con el cual es posible el control de movimiento de las persianas de la vivienda logrando así que la luz solar penetre en la cantidad deseada.

Con este shutter existen diversas posibilidades según el tipo de persiana con la que se desee acondicionar la vivienda. Existe la opción de Shutter o Louvre, la primera haría referencia a una persiana de tipo horizontal plana con tablillas planas, la cual sencillamente puede subir o bajar, por otro lado, el segundo término se refiere a una persiana con tablillas móviles que permiten la entrada del aire y la luz.

Este aparato permite también la opción de usar las tipo louvre como tipo shutter en el control de una pulsación corta, es decir en el parar la ventana. En el desarrollo de este proyecto se ha trabajado únicamente con ventanas del tipo shutter, configurando las dos posibles opciones, la apertura o cierre total de la ventana o bien la pulsación corta que nos permite detener la ventana.

Este aparato está compuesto de dos canales para cada una de las dos opciones, por tanto permite el control de un total de cuatro ventanas diferentes, dos de cada tipo. Se trabaja con dos canales permitiendo así la apertura total de la persiana con un canal y su apertura regular con el segundo.

### **6.3.1.3 Entrada/Salida binaria**

Mediante el uso de una Entrada/Salida binaria es posible el control domótico de la iluminación con un interruptor común. Con el uso de las entradas binarias es posible aprovechar los interruptores comunes de un hogar logrando así una domotización más económica.

En este PFC se utilizó una entrada binaria conectada a su vez una salida binaria, permitiendo de este modo el aprovechamiento de un interruptor común para lograr un encendido de la luz con las ventajas que nos ofrece una salida binaria.

Dentro de las posibilidades que brinda una salida binaria cabe destacar el control sobre el encendido, el tiempo de espera para el mismo, pudiendo poner un retardo del tiempo que se desee desde que se presiona el pulsador hasta que la bombilla se enciende.

### **6.3.1.4 EIB Solo Netzwerk-Koppler**

Este es el dispositivo que permitirá la visualización y el control a través de internet de nuestra instalación (pasarela IP).

Podemos encontrar dos tipos de parámetros a la hora de configurarlo, los específicos a la configuración de la red y los referentes a los tipos de datos que se van a configurar.

Para explicar la programación realizada en este componente, se explicará cómo se programaron los tipos de datos en primer lugar y más adelante se explicará la configuración llevada a cabo para su correcta conexión a la red y unión a la plataforma IP.

### 6.3.1.5 Pulsadores

Los pulsadores son interruptores que poseen una entrada binaria integrada y, por tanto, se encuentran específicamente preparados para el control de instalaciones domóticas.

En este proyecto se utiliza un único pulsador con el que será posible el control tanto del dimmer como de la regulación de las persianas. El pulsador utilizado es un pulsador doble con IR de siemens, que se divide en hasta cuatro botones configurables, en este caso únicamente fue necesaria la configuración para dos botones, lado derecho y lado izquierdo, en los cuales se especificó el uso concreto que se les quería dar.

## 6.4 Programación del sistema mediante el software ETS3

La programación del sistema domótico se realiza mediante el software ETS, en el caso de este proyecto concretamente con la versión 3.

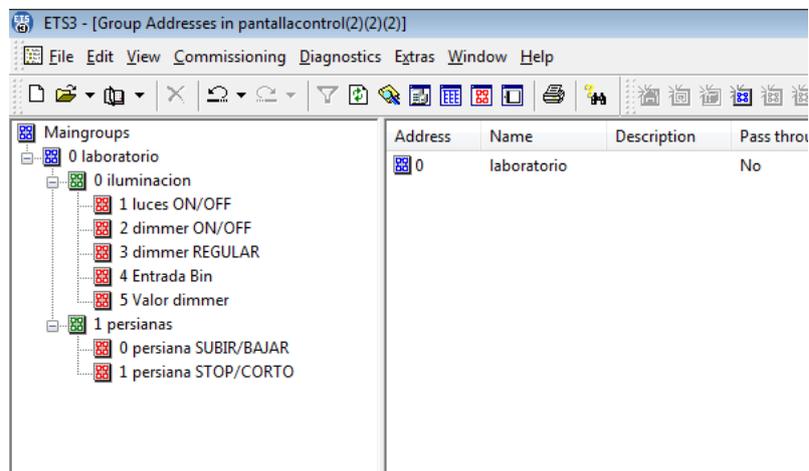
Para ello se comenzará importando una biblioteca que nos proporcionará la empresa del componente, en este trabajo se ha usado la biblioteca de siemens salvo en el componente del dimmer perteneciente a zennio. Tras tener la biblioteca se introducen los componentes en una línea y se comienzan a generar las direcciones de grupo, de tal manera que los componentes quedan interconectados como nos convenga.

A la hora de la conexión de cada componente se ha de tener muy en cuenta que función se espera del mismo, es decir, si el objetivo fuera un regulador de iluminación habría que trabajar con un dimmer y dentro de las opciones que presenta utilizar la necesaria para alcanzar el objetivo propuesto. A continuación vamos a explicar la configuración de las diferentes direcciones de grupo creadas y la programación usada en cada componente.

### 6.4.1 Direcciones de grupo

Para la realización de este proyecto fue necesaria la creación de dos direcciones de grupo diferentes, debido a las diferentes características deseadas de nuestro sistema, que a su vez fueron divididas en cuatro y dos grupos respectivamente.

Los dos grupos principales son de iluminación y persianas, a su vez dentro de los mismos tenemos distintos subgrupos para controlar de diversas formas los aparatos. Esta división puede observarse en la *figura 6.3*.



*Figura 6.3. Direcciones de grupo*

Como se puede observar, el grupo de iluminación se compone a su vez de cinco grupos, siendo sus direcciones 0/0/1, 0/0/2, 0/0/3, etc. A su vez el grupo de persianas se subdivide en dos grupos, con direcciones 0/1/1 y 0/1/2.

## 6.4.1.1 Iluminación

Dentro de este grupo se explicara la conexión de 3 de los subgrupos y la programación de sus componentes, que son los empleados para la domotización de la vivienda.

### 6.4.1.1.1 Dimmer ON/OFF

Este grupo se encarga del control de la iluminación a través del dimmer en un encendido y apagado total.

En este caso realizando una pulsación corta se generaría un cambio total de la iluminación, es decir, un encendido o apagado total mediante el uso del dimmer.

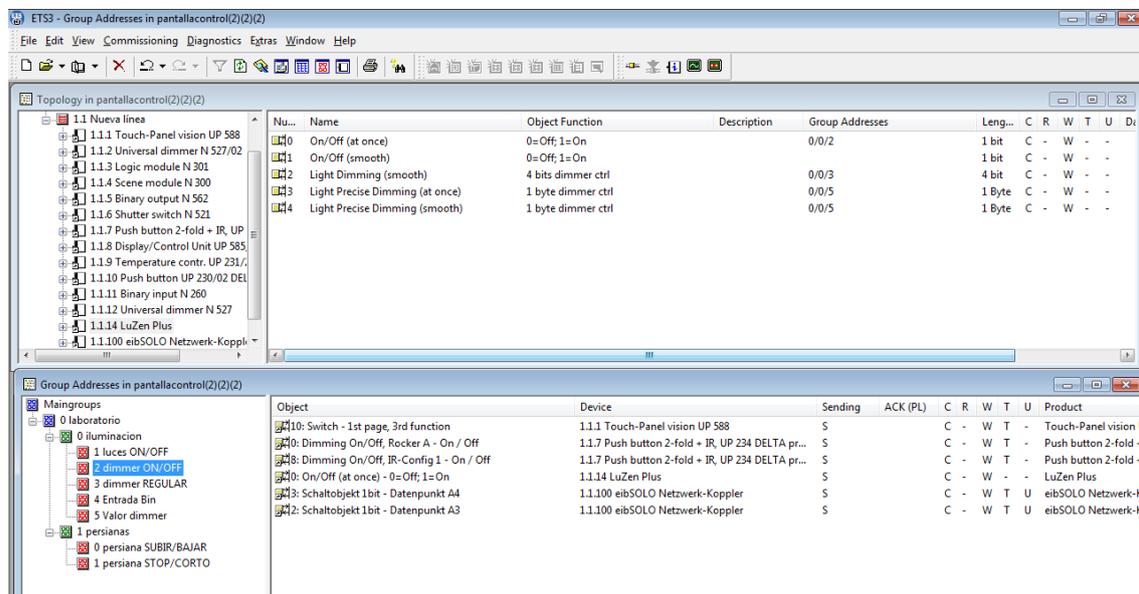


Figura 6.4. Componentes que conforma el grupo Dimmer ON/OFF

Como se muestra en la figura superior dentro del grupo que se está tratando tenemos diferentes componentes de la topología, ventana inferior, en ella se muestran los distintos componentes a los que se les ha dado esta dirección. En la ventana superior de la imagen se pueden ver las diferentes opciones que muestra el dimmer y las direcciones de grupo a las que están asociadas, la dirección correspondiente del grupo que se está tratando es la 0/0/2 que como se puede ver en la topología corresponde a la primera opción.

Además del dimmer en la ventana inferior, se puede observar que en este grupo también encontramos los componentes, eibSOLO y el pulsador. Al tener estos dos componentes estamos interconectándolos, con el pulsador se podrá controlar la luz, mientras que con el eibSOLO la conexión a la página web buscada será posible.

#### **6.4.1.1.2 Valor Dimmer**

Este grupo se encarga de la regulación de la iluminación, es decir, es capaz de encender o apagar la luz de forma gradual, manteniéndola al nivel deseado. En la *figura 6.5* se puede observar que en este grupo tenemos el dimmer y el eibSOLO para el control por internet, así pues ambos elementos están conectados entre sí.

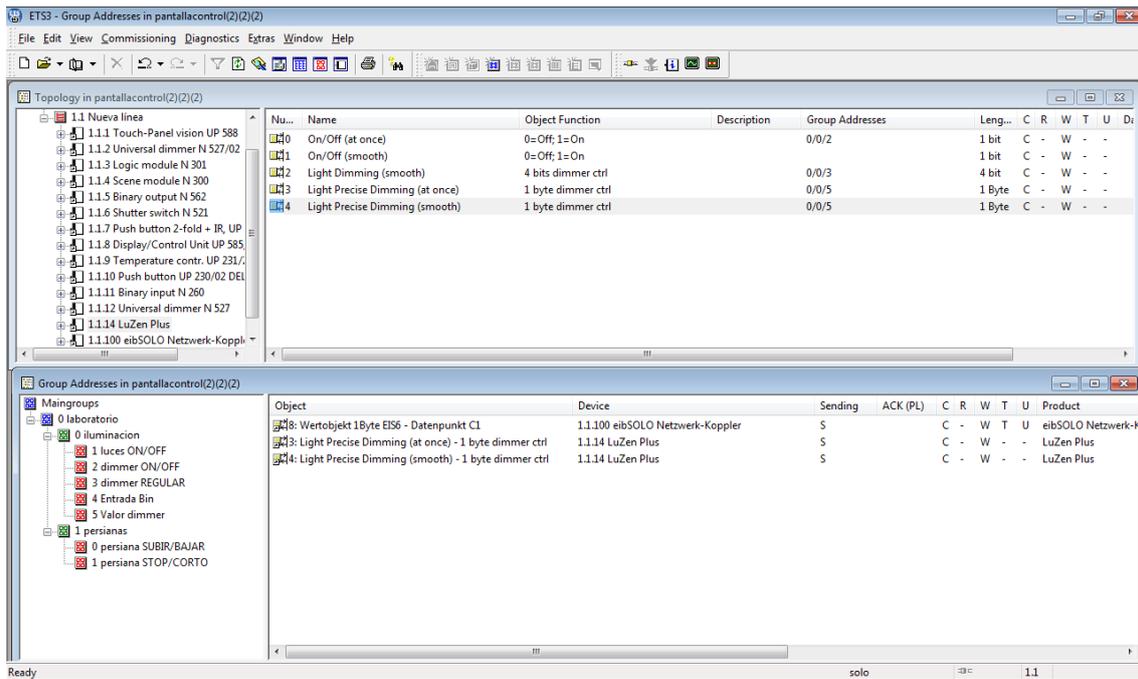


Figura 6.5 Componentes que conforma el grupo Valor dimmer

### 6.4.1.1.3 Entrada Bin

Aquí se encuentran los componentes de la entrada binaria y la salida binaria conectados, como se puede ver en la figura 6.6. Esto permite un control de la iluminación con una instalación normal, es decir, adapta un interruptor y una bombilla comunes y permite el control domotizado de los mismos. Además se encuentra el eibSOLO para el control por la red.

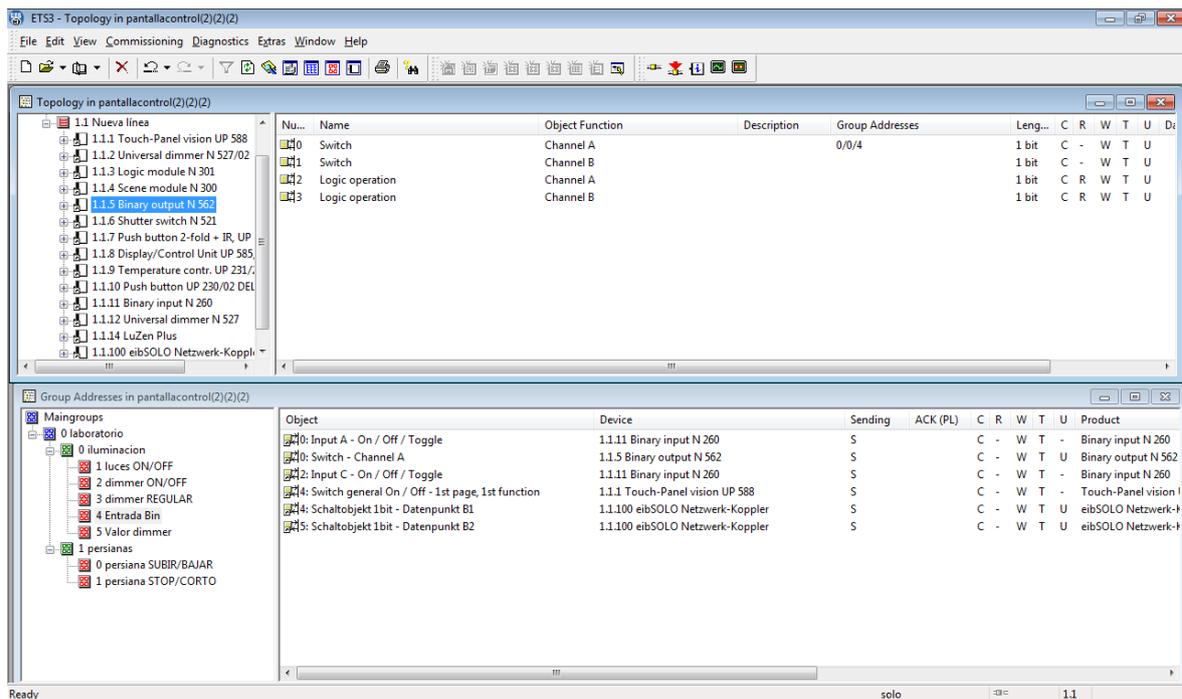


Figura 6.6. Configuración de la salida binaria

## 6.4.1.2 Persianas

Este grupo es el encargado del control de las persianas, como se ha comentado anteriormente utilizamos un único tipo de persianas, planas con tabillas fijas, en se hará que suban o bajen totalmente, con una pulsación larga, mientras que el otro grupo se encarga de parar el movimiento mediante una pulsación corta.

### 6.4.1.2.1 Persianas SUBIR/BAJAR

Observando detenidamente la figura 6.7 se puede comprobar que este grupo tiene conectado entre sí el shutter, o regulador de persianas, el eibSOLO y el pulsador con el cual se procederá al control de la persiana. En el shutter se está utilizando uno del tipo

shutter, lo cual implica que es plano de tablillas fijas, así pues, este grupo abrirá totalmente este tipo de persiana mediante una pulsación larga.

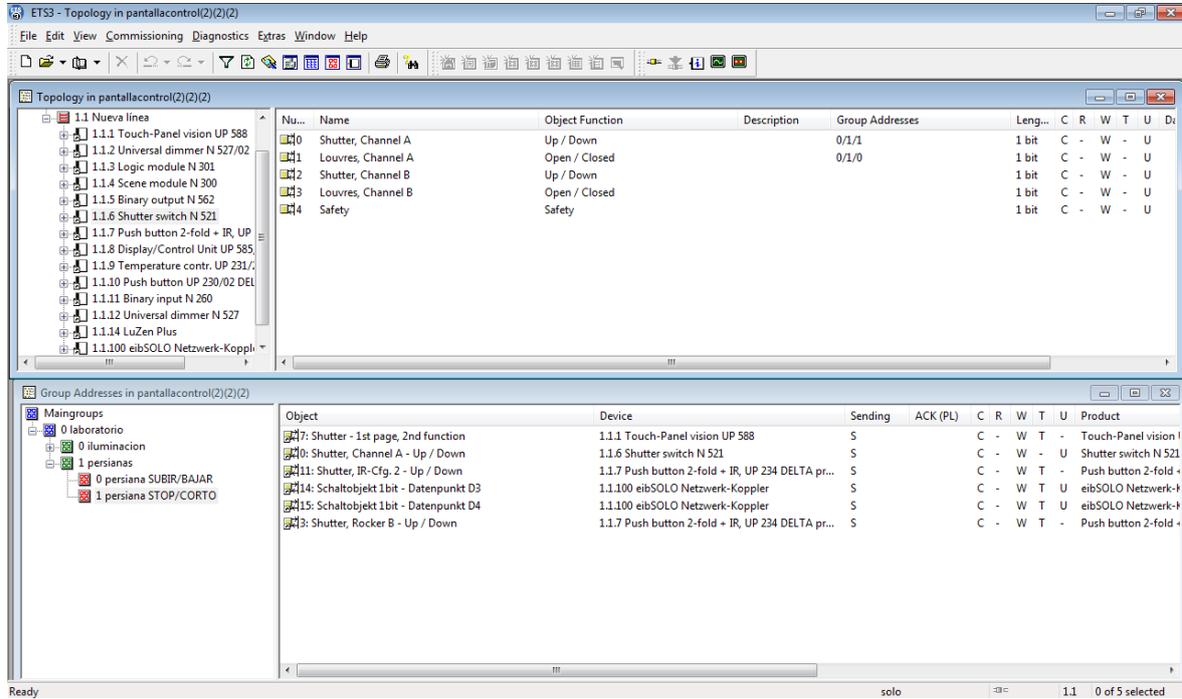


Figura 6.7. Conexión del grupo persiana SUBIR/BAJAR

#### 6.4.1.2.2 Persianas STOP/CORTO

Tal y como muestra la figura 6.8 en este caso la composición de los elementos interconectados en este grupo, es idéntica a la del grupo anterior con la excepción del tipo de persiana a utilizar en el shutter, que es una persiana de tipo Louvre configurada para su posible control de una persiana plana con tablillas fijas. Este grupo permite detener el movimiento de la persiana mediante una pulsación corta.

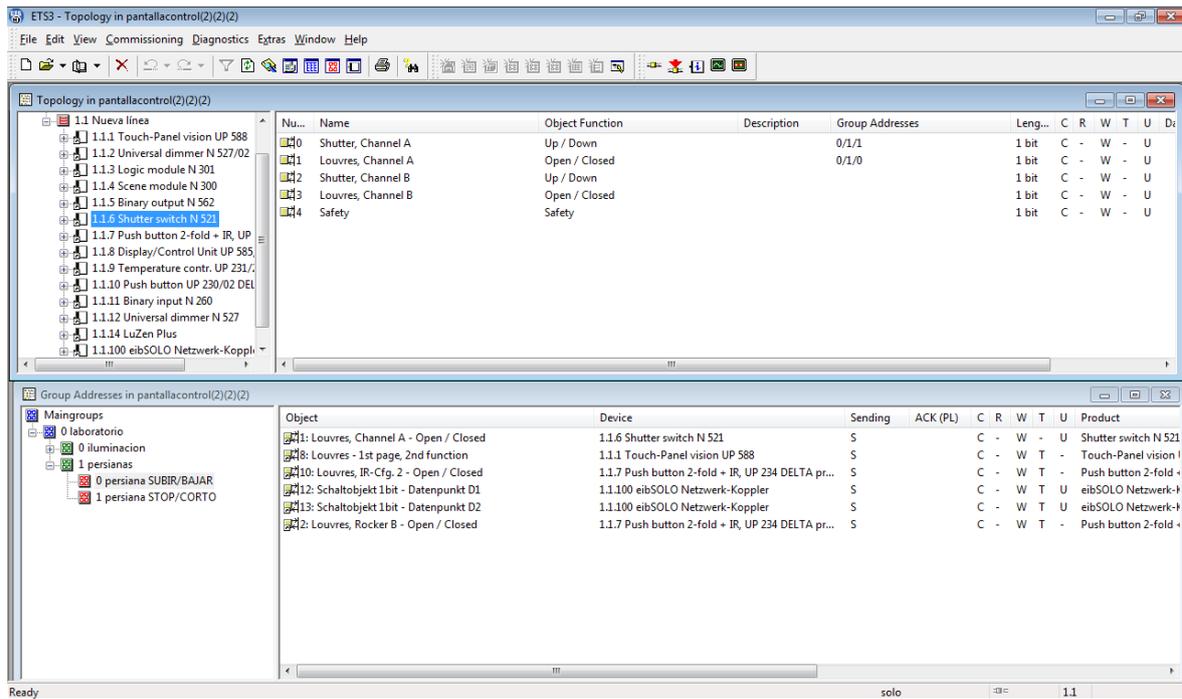


Figura 6.8. Conexión del grupo persiana STOP/CORTO

## 6.4.2 Programación de los componentes

En este apartado se tratará sobre la programación realizada a cada uno de los componentes anteriormente explicados para su correcto funcionamiento, tanto a nivel de control mediante pulsadores como a nivel de control mediante la red, siendo realmente este último punto en el que se centra este PFC.

### 6.4.2.1 EIBSolo. Network

Como ya se ha comentado con este aparato se puede lograr una conexión de nuestro equipo domótico a una red TCP/IP, por tanto cada uno de los aparatos que se desee controlar a través de internet deberán estar conectado al mismo.

A la hora de realizar la conexión a eibSOLO se ha de tener en cuenta una serie de cosas. Como se refleja en la figura, tenemos diferentes canales (A,B,C,D,..) a través de los cuales se puede realizar una conexión, dando a su vez cada uno un total de cuatro aparatos, es decir, es posible la conexión de cuatro aparatos por cada canal, mostrados en la pantalla de topología como A1,A2,A3 y A4 respectivamente.

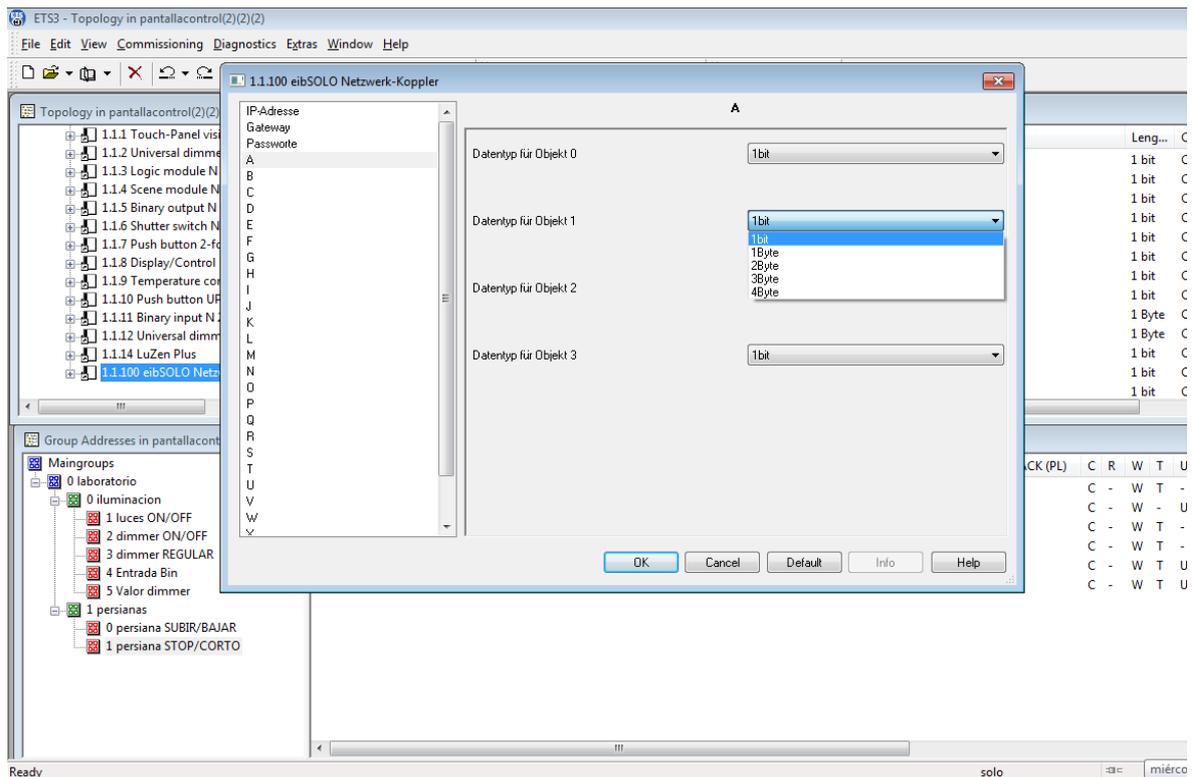


Figura 6.9. Programación de los datos del eibSOLO Netzwerk-Koppler

Cada uno de estas cuatro opciones tiene la posibilidad de soportar un tamaño de información diferente, pudiendo ser 1 bit, 1 Byte, 2Byte, etc. Así pues según el aparato que se desee conectar se deberá variar este punto, ya que el tamaño de la información ha de ser el mismo tanto en el aparato a conectar como en el eibSOLO.

En este proyecto se utilizó una configuración de 1 bit para todos los aparatos, puesto que no precisaban de mayor tamaño, excepto en el caso de dimmer que al necesitar

un tamaño mínimo de 4 bits se realizó el control mediante 1 Byte para posibilitar así su conexión a la red.

### 6.4.2.2 Dimmer

A la hora de programar el dimmer se buscó que cumpliera dos funciones principalmente, la de encendido total y la del encendido gradual. Como se observa en la imagen, la primera de las funciones se encuentra por defecto dentro de la configuración del dimmer, por tanto sólo importará el número de bits que necesita para poder realizar la acción deseada, siendo en este caso de un 1 bit, con lo cual al unirlo a su grupo correspondiente, dimmer ON/OFF, se deberá unir con un canal del eibSOLO que soporte 1 bit.

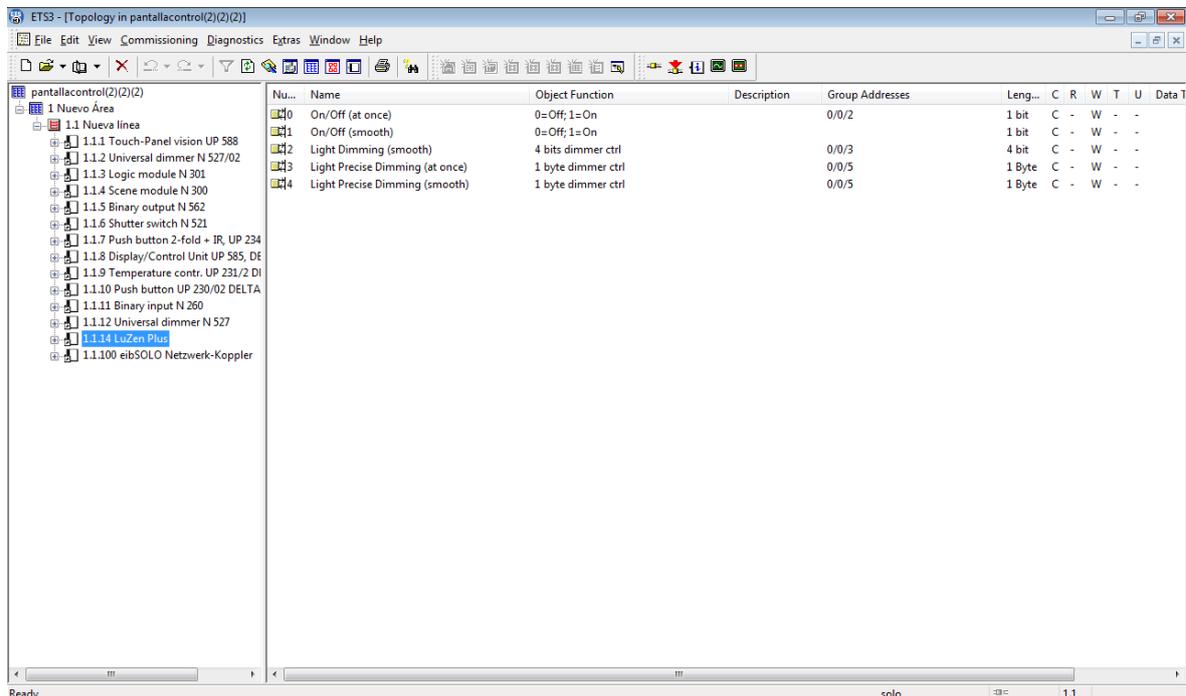


Figura 6.10. Configuración de los componentes del dimmer

En el caso de la segunda opción hubo que ajustar el porcentaje con el cual se deseaba realizar el incremento de la luz, se colocó a un 10%, consiguiendo por tanto que mediante una pulsación larga la luz se fuera incrementando a este porcentaje de iluminación.

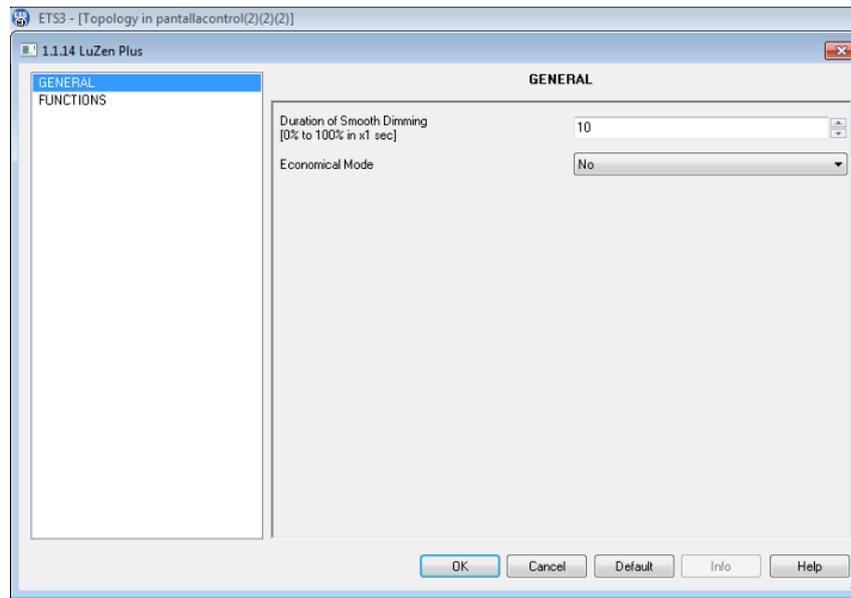


Figura 6.11. Programación del dimmer

Cabe destacar que en este caso el control de la iluminación necesita de al menos 4 bits para poder señalar el porcentaje de iluminación deseado, así pues al ser necesario el control a través del eibSOLO se optó por realizar el control mediante 1 Byte en lugar de los 4 bits, tal y como muestra la figura 6.12.

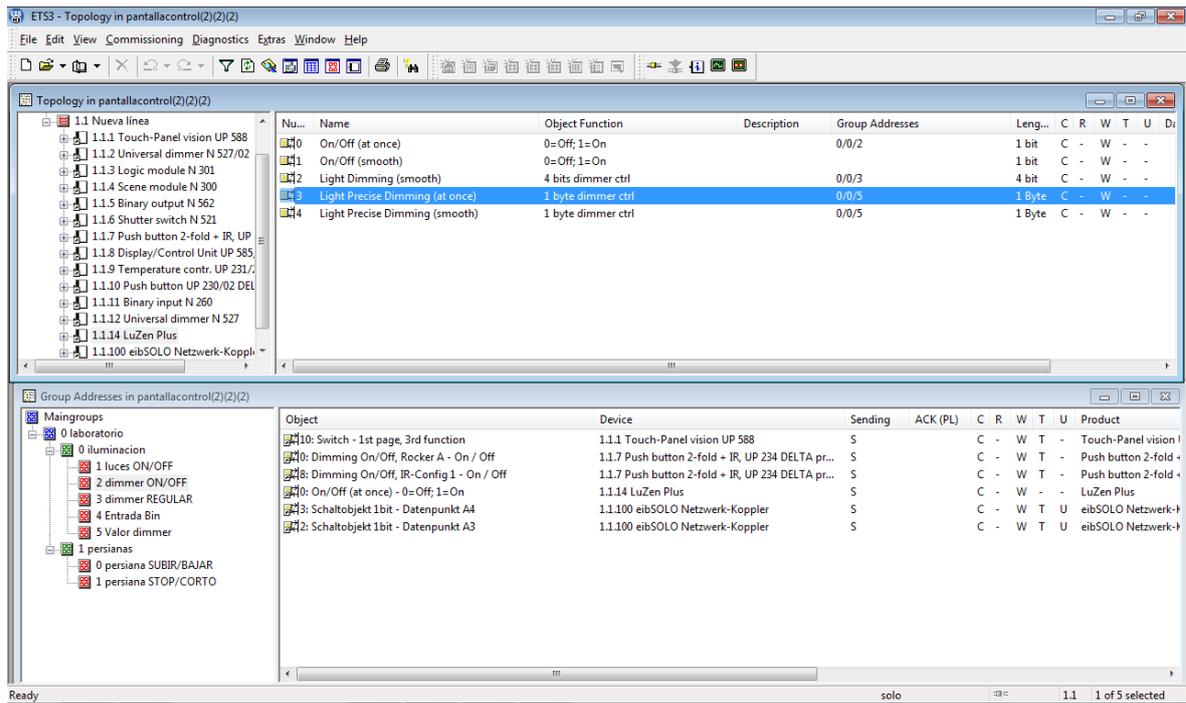


Figura 6.12. Tamaño de datos utilizado en el dimmer para su posible conexión

### 6.4.2.3 Shutter

A la hora de realizar la programación del controlador de persianas se ha de poner una configuración que permita el control de las persianas con la opción louvre, tal y como muestra la figura 6.13.

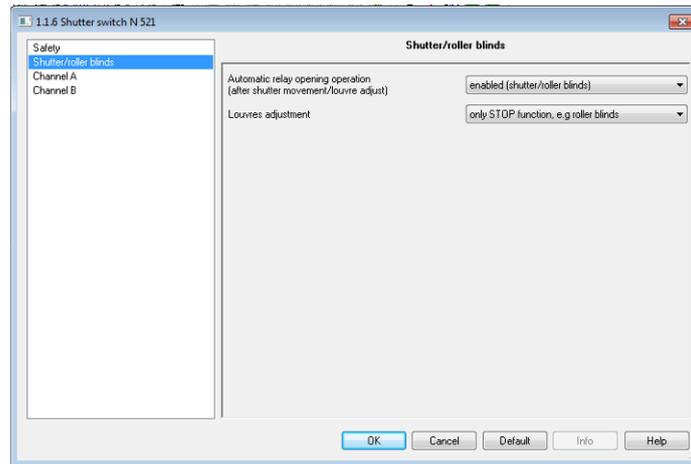


Figura 6.13. Programación del uso de las persianas louvre

Después de eso será necesaria la programación de los tiempos para el control de las pulsaciones, es decir, habrá que darle un tiempo de pulsación a las dos opciones a utilizar, (shutter y louvre) tal y como refleja la *figura 6.12*.

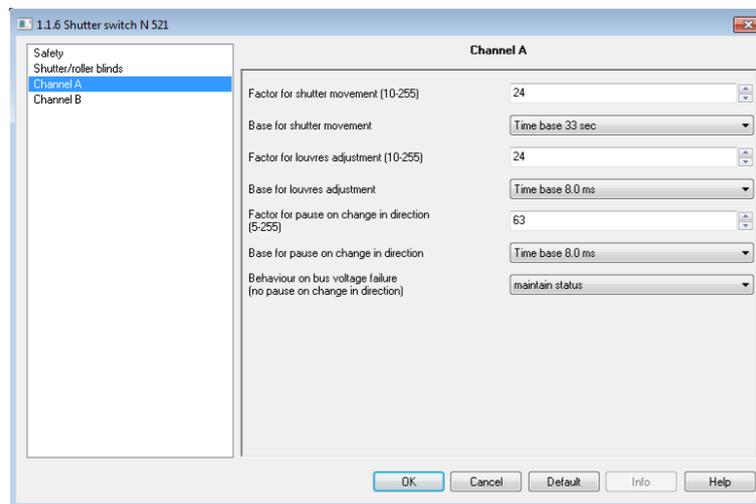


Figura 6.14. Programación de las pulsaciones a utilizar

### 6.4.2.4 Entrada binaria

Para la entrada binaria se han trabajado con dos interruptores simples que pudieran controlar la iluminación, de tal manera que, se buscó una programación que permitiera el control deseado independientemente del interruptor que se pulsara. Para lo cual se realizó la siguiente configuración.

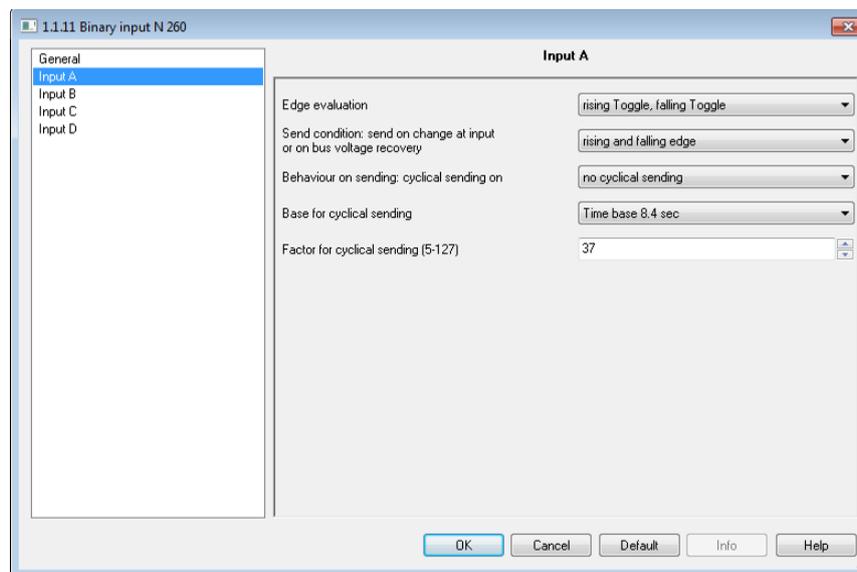
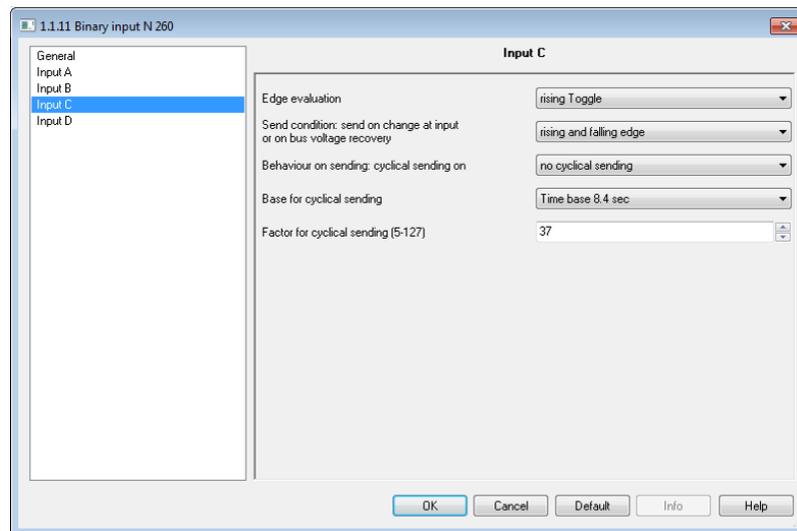


Figura 6.15. Programación de la entrada binaria

Como se puede observar de la figura 6.15 en el campo Edge evaluation, está seleccionada la opción rising toggle, falling toggle. Esto permite una inversión del valor cada vez que el flanco cambia, es decir, en este caso el aparato comprueba el valor existente y lo cambia en cada pulso de subida y bajada.

Para este caso se está utilizando un interruptor común de tipo on/off, debido a lo cual es necesaria la programación de una variación en ambos pulsos, subida y bajada.

Por otro lado, se tuvo que configurar otro canal para el otro interruptor el cual tiene los siguientes parámetros:



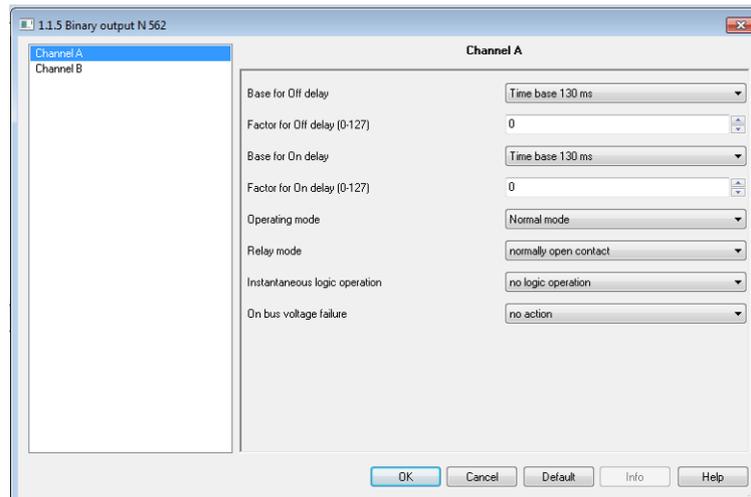
*Figura 6.16. Programación de la entrada binaria*

Como se ve en la figura superior en este caso está seleccionada la opción rising Toggle, en este caso el valor enviado se invierte únicamente después de cada flanco de bajada, es decir, al pulsar el interruptor enviamos un pulso que cambiará el estado en el que se encuentre el dispositivo.

En este caso se está utilizando un interruptor común tipo pulsador, es por tanto necesario la realización de una configuración que envíe pulsos, ya que con este tipo de interruptor no seleccionamos un estado como podría hacerse con un interruptor de on/off.

### 6.4.2.5 Salida binaria

En el caso de la salida binaria únicamente es necesario el uso de un canal el cual posee la siguiente programación

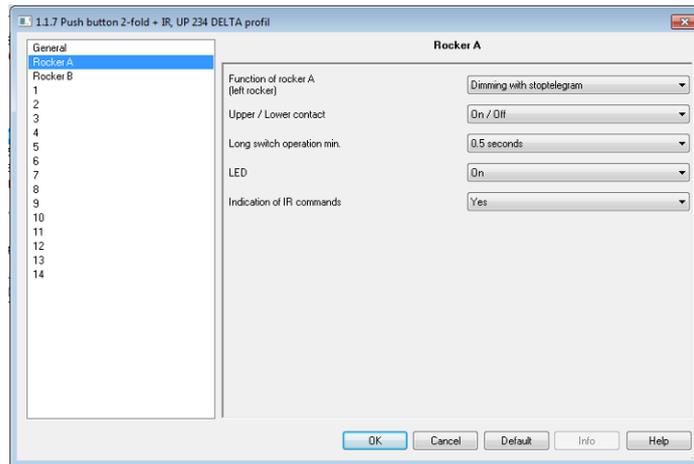


*Figura 6.17. Programación de la salida binaria*

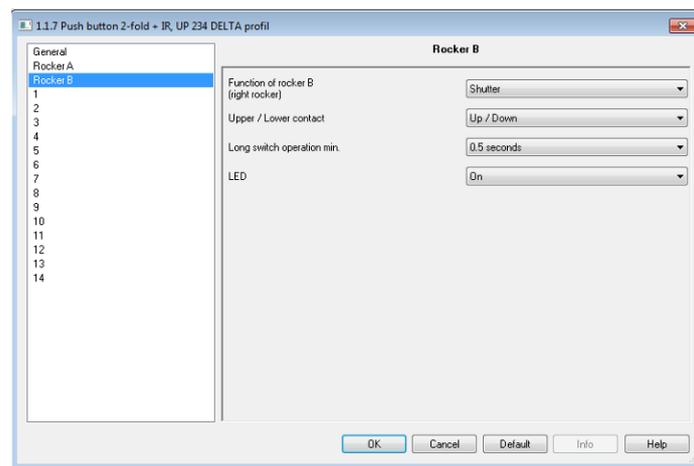
Con esta configuración se controla el retardo de encendido que llevará asociado la iluminación, esto se calcula a través de Base for Off delay y Factor for Off delay. Al multiplicar la base por el factor se obtendría el retardo, que como se puede observar en la figura anterior, en este caso al tener un factor 0 no habría retardo y por tanto se encendería automáticamente al pulsar el interruptor.

#### 6.4.2.6 Pulsadores

Como se mostró anteriormente en este proyecto se está usando un pulsador domótico de cuatro botones, con los cuales se controla el dimmer como las persianas. En la figura mostrada a continuación es posible observar como el botón derecho, correspondiente al Rocker A, se encuentra programado para el control del dimmer, mientras que en la *figura 6.19* se puede ver que el Rocker B, correspondiente al botón izquierdo está configurado para el control de shutter, o lo que es lo mismo, de las persianas.



*Figura 6.18. Programación del pulsador controlando el dimmer*



*Figura 6.19. Programación del pulsador controlando el shutter*

A continuación se explicará el software asociado a la pasarela IP eibSOLO, Nk2, necesario para la creación de la página web y del correcto funcionamiento de éste proyecto.

## 6.5 Diseño de la página web

Mediante el uso de una herramienta software proporcionada por el fabricante se creará una página web con la cual, mediante la programación de diferentes botones, se tendrá la vivienda controlada a través de internet.

Gracias a su interfaz muy intuitiva en poco tiempo se llega a tener creada la página web, con los botones que se consideren oportunos. Es posible la creación de varias páginas, por ejemplo, una página por habitación de la casa o por planta. En este proyecto, dado que no tenemos diferentes plantas, sino un entrenador domótico, nos decantamos por la creación de una única página que controla dicho entrenador y es accesible en cualquier planta de la vivienda. En la figura que tenemos a continuación se puede ver el resultado final de la edición de la página de este proyecto final de carrera.



Figura 6.20. Editor web NK2 con la página de este proyecto

Como podemos ver en la figura anterior, en este proyecto se realiza el control de las luces y las persianas de la vivienda. Para el control de la iluminación se crearon dos posibilidades:

- 1- Mediante el uso de entrada y salida binaria: Esto es una pulsación corta que enciende y apaga la luz completamente.

Esta opción queda representada en el panel como luces on off, así pues al pulsar el botón correspondiente se producirá un encendido o apagado.

- 2- Mediante el uso de dimmer: En este caso tenemos un regulador de incandescencia, que nos permitirá encender las luces gradualmente y mantenerlas en el porcentaje de iluminación que deseemos. Además, esta opción nos permite también el encendido y apagado total de la luz mediante una pulsación corta.

Esto queda representado en el panel con la opción Luces. Si pulsamos el botón se producirá un encendido o apagado total de la luz, mientras que, si enviamos un porcentaje en el botón inferior se producirá el incremento o decremento de la iluminación que deseemos.

En el caso de las persianas, podemos observar que se encuentran controladas mediante los botones de regulación de la parte izquierda, presentando también dos posibilidades

- 1- Subida total de la persiana: Con esta opción mediante una pulsación se produce un movimiento total de la persiana, es decir, la persiana que totalmente subida o bajada.

Esto se encuentra controlado mediante el botón de la izquierda del apartado de la persiana.

- 2- Regulación de la persiana: Esto nos permite la subida o bajada parcial de la persiana mediante una pulsación, quedando representado en la página mediante el botón de la derecha del apartado de las persianas.

Si observamos detenidamente podemos ver en la imagen que en las persianas existe un tercer botón amarillo. Esto es un indicador que nos mostrará cuando la persiana se encuentra totalmente subida pasando a un color rojo en dicho caso.

## 6.6 Producto final

Una vez la página web ha sido creada, la instalación domótica configurada y nuestro programa 3D finalizado, únicamente quedará la unión de estos tres proyectos para la creación del producto final.

Para ello se deberá comenzar con asociar nuestra instalación domótica con la página web creada. Gracias al software de NK2 esto se puede realizar de una forma muy simple. Primero se debe realizar la configuración correspondiente en el programa ETS, para lo cual se carga el proyecto deseado y en la opción editar parámetros del componente eibSOLONetzwerk-Koppler se comienza la configuración del mismo, tal y como nos muestra la *figura 6.21*.

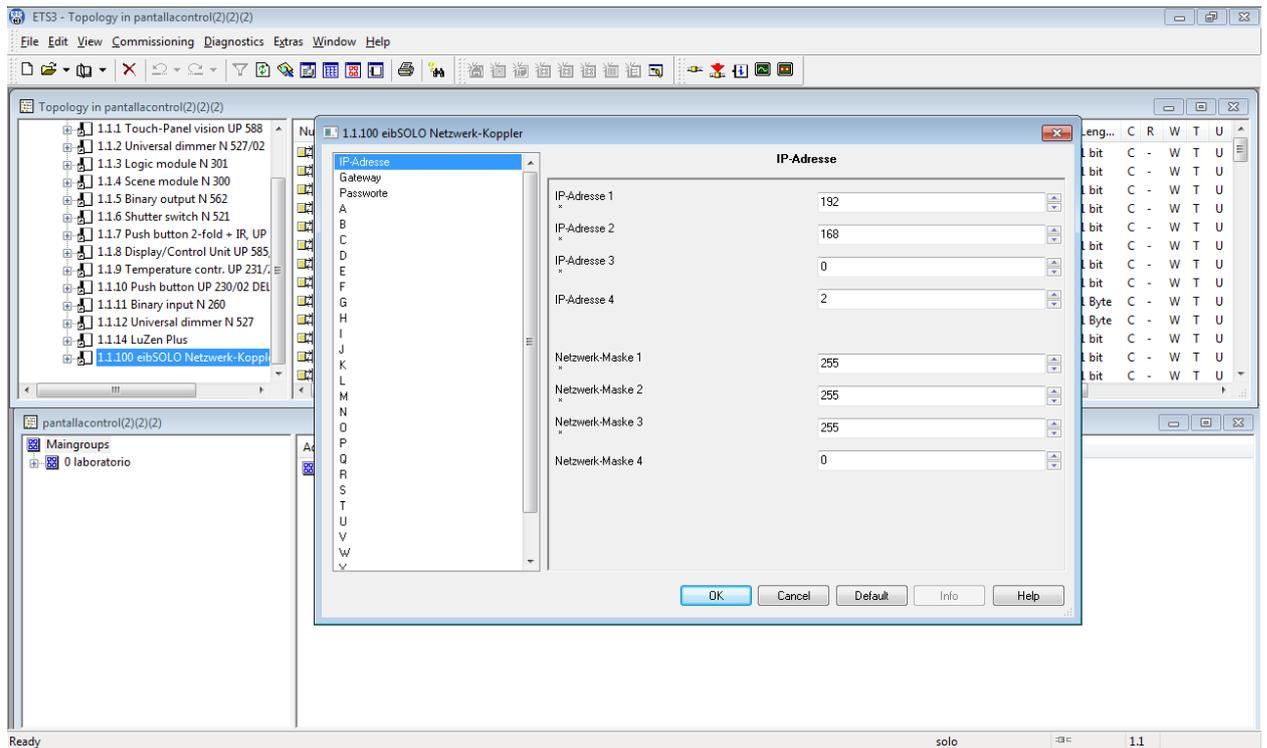


Figura 6.21. Parámetros de configuración del ETS

Hay tres opciones para realizar la configuración:

- 1- IP Adresse: Introducimos la dirección IP por defecto especificada en el manual que sería 192.168.0.2, con una máscara de red 255.255.255.0
- 2- Gateway: En este caso la puerta de enlace a usar es la 192.168.0.1
- 3- Passworte: Aquí se procede a introducir un usuario y contraseña, que en este caso fue user, Nk1 respectivamente, con el puerto estándar para http 80 y para ftp 21.

Con todo esto listo se debe pasar a introducir los datos del ETS al NK2, para lo cual tan sólo se irá a propiedades de proyecto como queda representado en la imagen y adjuntar los parámetros correspondientes a nuestro programa de ETS apareciéndonos la ventana de configuración, todo esto queda representado en las figuras mostradas a continuación.



Figura 6.22. Acceso a propiedades del proyecto del Nk2

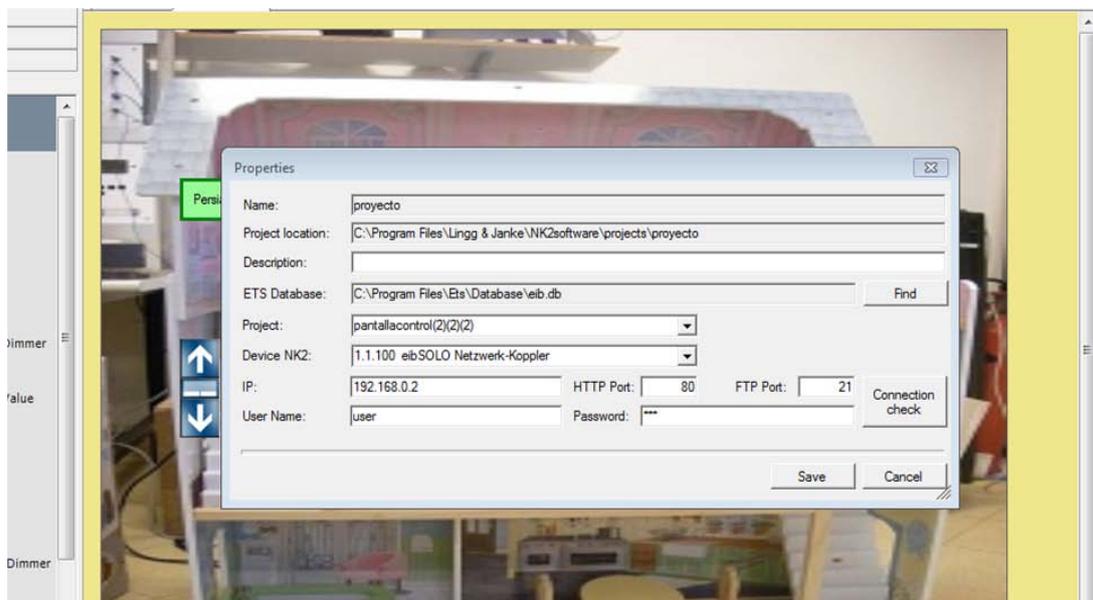


Figura 6.23. Parámetros de configuración del Nk2

En el apartado Proyec location introducimos la ubicación del proyecto de ETS que se desee unir, después se asocia la librería correspondiente en el apartado de Database, por último queda seleccionar el proyecto ETS que se vaya a asociar, en este caso fue pantallacontrol(2) (2) (2), de este modo ambos proyectos, el ETS y el NK2 quedan unidos y es posible el control mediante la página.

Una vez realizado todo esto, es posible el acceso a la página creada introduciendo la dirección `http://192.168.0.2/page1.htm` en el explorador, y desde ahí se realizará el control de la casa domotizada.

Ahora bien, en este proyecto se propone el uso de un interfaz gráfico 3D para el control de la vivienda. Para ello, se ha optado por una solución en la que dicho interfaz gráfico, no transmite primitivas KNX a la red domótica directamente (para lo cual sería necesario contar con un software adicional tipo falcon o calimero), sino que hace uso de llamadas en protocolo http a la página web de la pasarela IP. De manera que será ésta, la encargada de transmitir a la red domótica el comando KNX correspondiente en cada caso.

En el editor web del GameStudio nos crearemos un Sprite al cual le asociaremos una acción, la cual variará según lo que se desee controlar, es decir, se implementará una acción para las luces, otra para las persianas y otra para regular la intensidad.

La creación de los comandos apropiados para el control de los aparatos domóticos se obtuvo tras el análisis de las tramas http enviadas por el cliente web de un PC que se conecta al servidor web de la pasarela IP, cuando se activan los diferentes controles implementados en la página web de la instalación domótica. Estos comandos permiten controlar la instalación domótica a través del explorador, así pues, creando un código que realice el acceso al explorador en una página determinada hace posible la transmisión de las tramas necesarias para comunicarse con el sistema domótico. La composición de las tramas enviadas para el control de los elementos domóticos y los códigos empleados para su control mediante el entorno virtual son los que se muestran a continuación.

Las tramas enviadas por la página web al elemento domótico poseen la siguiente estructura:

```
http://192.168.0.2/GET      /.asdfg.htm?elemento-valor=+++0++&1320060286180
HTTP/1.1
```

Donde el elemento indicará el aparato domótico a tratar y su modo de funcionamiento, existiendo las siguientes posibilidades:

BO-- dimmer pulsación corta. La luz se enciende o apaga totalmente

B1-- luces de la entrada y salida binaria. La luz se enciende o se apaga totalmente.

D1-- pulsación corta de la persiana. Detiene la subida o bajada de la persiana.

D3-- pulsación larga de la persiana. Sube o baja totalmente la persiana.

C1-- dimmer en función de regulación. Enciende gradualmente la luz

Mediante la opción valor, se puede seleccionar si se desea apagar la luz, subir o bajar las persianas o especificar la graduación de iluminación que se desee tener. Para lograr esto se trabaja con un bit, tal y como vimos anteriormente, o en el caso del dimmer con 1 Byte. Con lo cual se tienen dos opciones 0, que apaga las luces o sube las persianas, y 1, que enciende las luces o bajan las persianas, por otro lado en el dimmer los valores posibles de luminosidad permitidos oscilan entre 0 a 255, siendo 0 el apagado total y 255 el encendido.

Teniendo esto en cuenta, es posible controlar cada uno de los elementos sin necesidad de realizar el acceso a la página web del panel de control. En este proyecto se ha optado por la realización de ambas opciones dentro del interfaz gráfico 3D, es decir, se han introducido botones para el control de cada una de las acciones de los elementos por separado y se ha agregado un botón más para el acceso a la página web de control de la instalación desde el propio entorno 3D. Como ya se ha mencionado en el capítulo anterior, cada uno de estos botones tiene asociada una acción que realiza una llamada a un archivo que permite el envío de la trama deseada al sistema domótico, evitando así el acceso a la web.

A continuación mostramos el código implementado para cada una de las acciones necesarias para el control del sistema domótico.

```
function abrir()
{
exec("C:\\abrete.bat","");
}
ACTION web
{
my.enable_click=on
my.event=texto;
my.event=abrir;
}
```

```
function abrir1()
{
exec("C:\\luces.bat","");
Sleep (1)
exec("C:\\cerrar.bat","");
}
ACTION luces
{
my.enable_click=on
my.event=texto;
my.event=abrir1;
}
```

```
function abrir2()
{
exec("C:\\lucesoff.bat","");

Sleep (1)

exec("C:\\cerrar.bat","");
}

ACTION lucesoff
{
my.enable_click=on

my.event=texto;

my.event=abrir;
}
```

```
function abrir3()
{
exec("C:\\dimmeron.bat","");

Sleep (1)

exec("C:\\cerrar.bat","");
}

ACTION dimmeron
{
my.enable_click=on

my.event=texto;

my.event=abrir3;
}
```

```
function abrir4()
{
exec("C:\\dimmer25.bat","");

Sleep (1)

exec("C:\\cerrar.bat","");
}

ACTION dimmer25
{
my.enable_click=on

my.event=abrir4;
}
```

```
function abrir5()
{
exec("C:\\dimmer50.bat","");

Sleep (1)

exec("C:\\cerrar.bat","");
}

ACTION dimmer50
{
my.enable_click=on

my.event=abrir5;
}
```

```
function abrir6()
{
exec("C:\\dimmer75.bat","");
Sleep (1)
exec("C:\\cerrar.bat","");
}
ACTION dimmer75
{
my.enable_click=on
my.event=abrir6;
}
```

```
function abrir7()
{
exec("C:\\dimmeroff.bat","");
Sleep (1)
exec("C:\\cerrar.bat","");
}
ACTION dimmeroff
{
my.enable_click=on
my.event=abrir7;
}
```

```
function abrir8()
{
exec("C:\\persianasparriba.bat","");

Sleep (1)

exec("C:\\cerrar.bat","");
}

ACTION persianasparriba
{
my.enable_click=on

my.event=abrir8;
}
```

```
function abrir9()
{
exec("C:\\persianaspcabajo.bat","");

Sleep (1)

exec("C:\\cerrar.bat","");
}

ACTION persianaspcabajo
{
my.enable_click=on

my.event=abrir9;
}
```

```
function abrir10()
{
exec("C:\\persianasplarriba.bat","");

Sleep (1)

exec("C:\\cerrar.bat","");
}

ACTION persianasplarriba
{
my.enable_click=on

my.event=abrir10;
}
```

```
function abrir11()
{
exec("C:\\persianasplabajo.bat","");

Sleep (1)

exec("C:\\cerrar.bat","");
}

ACTION persianasplabajo
{
my.enable_click=on

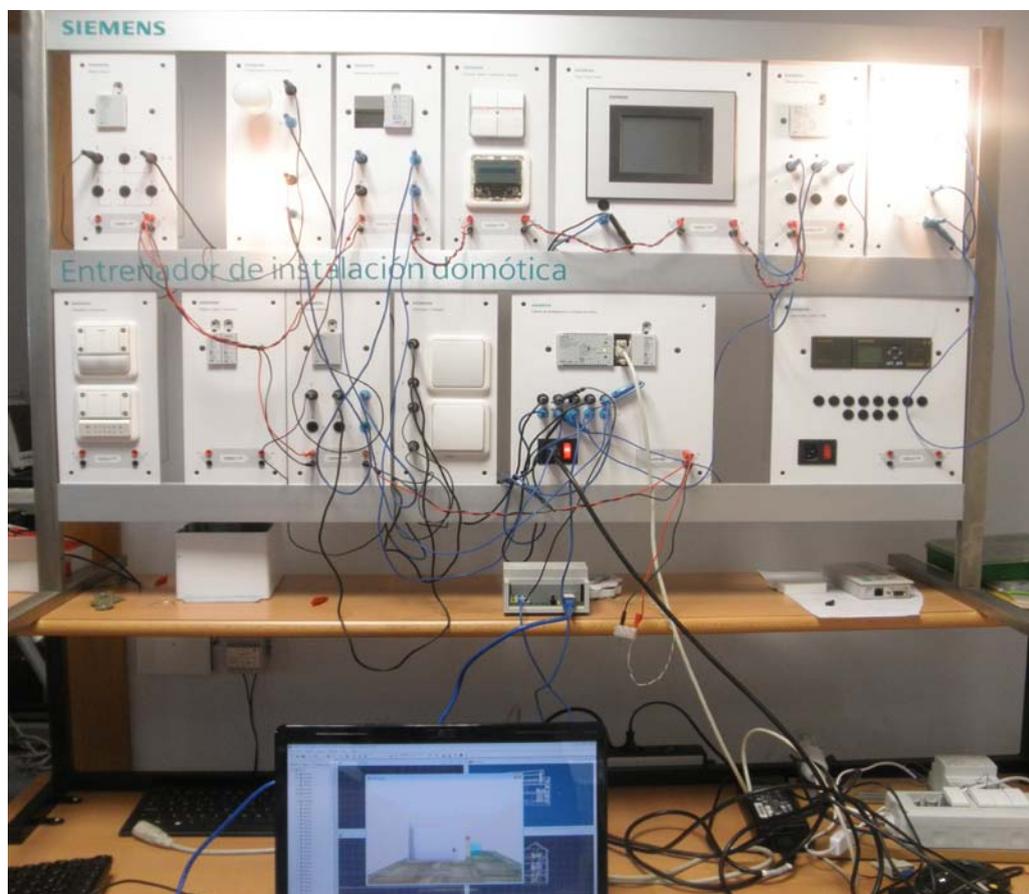
my.event=abrir11;
}
```

Tal y como se puede observar en el código, de las opciones de regulación que ofrece el dimmer se han implementado 3 de ellas, que consisten en una luminosidad del

25%, 50% y 75% (además del encendido total y el apagado), puesto que estos son los valores que permiten una mejor apreciación de la graduación de la luminosidad.

Mediante este código se vuelve posible el control del sistema domótico a través del entorno virtual, tal y como se buscaba al comienzo de este proyecto.

El sistema final que se logró en este PFC es el mostrado en la *figura 6.24*, donde es posible la visualización del entrenador utilizado para la realización de las pruebas y del ordenador con el entorno virtual en funcionamiento.



*Figura 6.24. Implementación del sistema final*

En la figura que se muestra a continuación podemos ver de forma más clara el entorno virtual con los diferentes botones de interacción desarrollados.



*Figura 6.25. Entorno virtual con los botones de control*

# **Bloque IV**

## **Conclusiones**



## Capítulo 7

### Conclusiones

*“No aprendemos para la escuela, sino para la vida”*

*Lucio Anneo Séneca*

*“En los momentos de crisis, sólo la imaginación  
es más importante que el conocimiento”*

*Albert Einstein 1879-1955. Científico Alemán.*

En este capítulo se comentan las conclusiones obtenidas tras la realización de este proyecto final de carrera explicado a lo largo de todo este documento.

En este proyecto se ha desarrollado un prototipo que realiza el control, mediante entorno virtual 3D, de una instalación domótica. Para ello es necesario, en primer lugar, la implementación y programación de la misma, que incluye una aplicación de control remoto, en este caso mediante un acceso vía internet. Una vez esto se encuentra operativo, se puede comenzar con la creación de la plataforma 3D con la cual se piensa realizar la interacción con el usuario y el control de la vivienda propiamente dicho. Finalmente, Tras el completo funcionamiento del entorno en la etapa final se procede a la conexión de dicho

entorno con internet, para acceder a la interfaz de control implementada en la instalación domótica y comprobar el correcto comportamiento del sistema total

El motivo por el cual se pretende realizar el control de la vivienda mediante internet, en lugar de controlarlo por un panel, o cualquier otro medio, radica en el crecimiento exponencial que ha sufrido esta tecnología, así como las múltiples posibilidades que ofrece para la vivienda, tales como la simulación de presencia o el encendido de la calefacción antes de llegar a la vivienda.

Para poder realizar este control por internet de los diversos aparatos, es necesario un mecanismo que permita realizar la conexión de los mismos a la red. Para ello, KNX ofrece diferentes posibilidades como por ejemplo el uso de unas librerías, sin embargo, en este proyecto final de carrera se optó por otro sistema, más moderno, que es conocido como pasarela KNX/IP. Esta plataforma consiste en un programa que mediante un aparato homologado por KNX que hace de puente entre el sistema domótico y una página web, de tal modo que, mediante el software ETS3 de KNX, es posible extraer la programación de los aparatos que conforman el sistema a la página web creada mediante el software de la plataforma. De esta manera es posible construir una página en la cual tengas las diversas posibilidades que ofrece la instalación domótica y así poder controlarla.

Sin embargo, aunque el control mediante una página web es muy simple y eficiente, no supone el objetivo final de este proyecto.

A lo largo de este PFC, se ha mostrado la necesidad existente del ser humano de no sólo comunicarse, sino de realizarlo en un entorno cómodo.

Diversos estudios han demostrado que una interfaz que el usuario considere amigable da mejores resultados que cualquier otro tipo. Así pues, tras estudiar las diferentes posibilidades en la interfaz a presentar, se llega a la conclusión que una mayor parte de la sociedad actual encuentra en los videojuegos una comodidad y un entorno conocido y por tanto más amigable del que puede ofrecer un explorador de internet.

A la hora de realizar un videojuego para el control de la vivienda, existen multitud de formas en las cuales sería posible presentar el entorno en cuestión. Con esto se vuelve necesaria la realización de un estudio a cerca de los diferentes videojuegos existentes y su interacción con el usuario y entornos externos al juego en cuestión. Tras una investigación apropiada, se llega a la conclusión de la necesidad de utilizar un videojuego

de tipo simulación basado en un modelo de tipo FPS, puesto que en el estudio se concluye que este tipo de videojuegos genera una mayor comodidad para el usuario, haciendo que se sienta parte de las acciones que realiza y con el cual la interacción con el mundo exterior es posible de una forma fácil y efectiva.

A la hora de realizar la interfaz se vuelve necesaria la utilización de un software especializado, denominado motor de juego o game engine, el cual es específico para el tipo de juego a desarrollar. Dado que, como se especificó anteriormente, se optó por la creación de un entorno de juego, del tipo FPS se vuelve necesaria la utilización de un motor específico para la creación de este juego.

Para la selección del motor apropiado dentro de la gama existente, se procedió a otro estudio, en el cual se concluyó con la elección del software GameStudio. Esta decisión se basó en que este software no sólo cumplía con las necesidades existentes para la creación de la plataforma virtual, sino que además proporcionaba un entorno simple, de fácil manejo y muy completo, que permite modificar todo lo que se desee hasta el grado de complejidad buscado.

Para la realización de los objetivos buscados, que es simplemente el control del sistema mediante un interfaz amigable para el usuario desde cualquier punto, se comenzó con realizar el apartado de la domotización. Para ello se seleccionó, tras un exhaustivo estudio, un tipo de estándar, concretamente se utilizó el sistema KNX, puesto que además de ser el más empleado dentro de Europa, posee una gran variedad de productos. Con ello permite una gran flexibilidad y comodidad a la hora de introducir diferentes componentes de cualquier empresa, ya que únicamente es necesario el cumplimiento de dicho estándar para su correcto funcionamiento en una red domótica ya creada, facilitando así la sustitución o ampliación de cualquier componente.

Se procedió a realizar el conexionado de los diferentes dispositivos que compondrían el sistema domótico del que dispondría la vivienda. Una vez realizado este conexionado se pasó a realizar la programación de cada uno de los componentes con el software ETS3 de KNX específico para esta labor. Mediante esta conexión y programación es posible el control del sistema con el uso de pulsadores o de una pantalla táctil, en el caso concreto de este PFC únicamente se hizo uso de los pulsadores, ya que el objetivo no radica en la domotización, sino en el control de la misma mediante una interfaz 3D a través de internet.

Puesto que en este proyecto no se ha realizado la domotización de una vivienda ni se contaba con una vivienda con instalaciones domóticas asociadas, a lo largo de este PFC se implementó un entrenador domótico, que incluía los dispositivos KNX seleccionados, para la realización de las pruebas y la comprobación de su correcto funcionamiento, sin que por ello se sufriera una pérdida de dificultad ni de generalidad. De esta manera, se puede asegurar que el sistema es perfectamente funcional para una vivienda tras realizar las pruebas en el demostrador domótico, siendo los resultados obtenidos, por tanto, igual de válidos que si se hubiera realizado en una vivienda real.

Tras realizar de forma deseada la programación del sistema domótico, se procederá a la creación de la plataforma IP, mediante la cual controlaremos toda la instalación. Con el uso del software específico Nk2 para el control del aparato eibSOLO, programado para la conexión del sistema a la red, se construye una página mediante la cual a través de internet podemos controlar la vivienda de un modo muy intuitivo.

La elección de esta plataforma concreta dentro de las diversas opciones que se presentan en la actualidad, es debido a su interfaz sencilla y de bajo coste, en donde con la introducción de nuestro entorno 3D se generarán las altas prestaciones que producirán el atractivo del producto. Así pues, el uso de esta interfaz web permite un acceso sencillo y estandarizado con primitivas específicas en los entorno de programación facilitando así el control del sistema, sin sobrecargarlo con un exceso de gráficos innecesarios para lograr los objetivos propuestos y pudiendo generar una ralentización del sistema.

Una vez la vivienda era perfectamente controlable mediante una página web en internet se procedió a la realización del entorno 3D. El entorno creado está formado por lo que sería el hogar del usuario, así pues se tendrá que el escenario en el cual se desarrolla toda la acción, es nada más y nada menos que una casa.

La vivienda representada en este entorno se basó en un modelo existente, que se corresponde a la maqueta de una casa de muñecas. Puesto que el objetivo del presente proyecto es el desarrollo de esta interfaz y al no existir una vivienda domotizada concreta, se optó por el uso de una maqueta, sin que por ello exista un cambio significativo en lo concerniente a la creación de la plataforma virtual y su funcionamiento. Dicha maqueta, está formada por tres plantas, en las cuales en el entorno virtual se situaron diversos botones con los que el personaje manejado por el usuario puede interactuar, logrando así un acceso al mundo exterior. Estos botones fueron programados para que cuando fueran

activados por el personaje se hiciera una llamada a un archivo encargado de transmitir una trama específica con la cual es posible el control del elemento domótico asociado al botón, además se añadió otro botón que permite la apertura del explorador en la página web desarrollada durante este PFC para el control de la red domótica completa desde ese punto.

A lo largo de este proyecto final de carrera fue necesaria la reprogramación de todos los componentes debido a un fallo en el dimmer existente, el cual tuvo que ser reemplazado, tras realizarle todas las pruebas necesarias para la comprobación de su correcto funcionamiento, demostrando con las mismas que el aparato se encontraba en mal estado haciendo, por tanto, imposible su uso.

La sustitución se hizo por un elemento equivalente de otra empresa, que pudo acoplarse satisfactoriamente gracias a que cumplía el estándar KNX, que como comentamos anteriormente, permite la conexión de diferentes aparatos siempre que cumplan con dicho estándar. Así pues, su inclusión y programación no supuso importantes cambios en la configuración del sistema ya implementado, reafirmando la elección realizada tras el estudio domótico para la selección del estándar a utilizar en la instalación de la red domótica que se desarrolló.

Por último tras la conexión del aparato sustituido y la reprogramación de todos los componentes del sistema domótico, se realizaron las diferentes pruebas para la comprobación del correcto funcionamiento del sistema, logrando el control del entorno domótico creado mediante el entorno virtual desarrollado. Concluyendo de este modo que el proyecto final de carrera “*Entorno 3d Interactivo para Control de Redes Domóticas*” ha alcanzado su fin logrando cumplir los objetivos inicialmente establecidos.



## Bibliografía

- [1] Cristóbal Romero Morales, Francisco Vázquez Serrano, Carlos de Castro Lozano. “*Domótica e Inmótica. Viviendas y Edificios Inteligentes*”. RA-MA 2005
- [2] José Manuel Huidobro Moya, Ramón J. Millán Tejedor. “*Domótica. Edificios Inteligentes*”. Creaciones Copyright, SL. 2005
- [3] Casell, J., Sullivan, J., Prevost, S., Churchill E (eds). “*Embodied Conversational Agents*”, Cambridge: MIT Press, 2000
- [4] Maldonado, H., Hayes-Roth, B. “*Toward cross-cultural believability in character design. In Agent Culture: Human-agent interaction in a multi-cultural world*”. S Payr and R. Trappl (eds), pp. 177-196, 2001
- [5] Nijholt, A. “*Where computers disappear, virtual humans appear*”, Computer & Graphics nº 28, pp. 467-476, 2004.
- [6] Creed, C., Beale, R. Embodied Interfaces: “*The Next Generation of HCI?*” Proc. of the Workshop on The Next Generation of HCI in cooperation with the Conference on Human Factors in Computing Systems (CHI2006), 2006
- [7] Nijholt, A. “*Multimodal interactions in ambient intelligence*”, in W. Verhaegh, J.Korst, E. Aarts (eds). Algorithms in ambient intelligence, Chapter 2. Phillips Research Book Series. Dordrecht: Kluwer Academic Publishers, 2003.

[8] Fernando Antón Alonso, Pedro Antón Alonso, Pablo Trinidad Martín Arroyo. “*Motores 3D y Lenguajes de Script. Un ejemplo práctico con 3D GameStudio*”. Ediversitas Multimedia, S.L.2003

[9] Daniel González. “*Diseño de Videojuegos. Da forma a tus sueños*”. RA-MA 2011.

[10] KNXA (KONNEX Association). “*Técnica de Proyectos en Instalaciones con EIB. Principios Básicos*”. KNXA, 2006

[11] Eduardo Hernández Pérez: “*La integración entre Internet, telefonía móvil y domótica en los sistemas electrotécnicos del hogar*”, Curso doctorado bienio 02-04: Redes no guiadas, 2004.

[12] Asociación KNX

<http://www.knx.org/es/>

Marzo de 2011 Última visita Enero 2012

[13] Simone Belli y Cristian López Raventós “*Breve historia de los videojuegos*”  
Athenea Digital núm. 14: 159179 Universidad Autónoma de Barcelona. Otoño 2008

[14] Unity Technologies

<http://unity3d.com/unity/>

Última visita Enero 2012

[15] Epic, Epic Games, Unreal, Unreal Development Kit

<http://udk.com/>

Última visita Enero 2012

[16] Autodesk, Inc. “Scaleform”

<http://gameware.autodesk.com/scaleform>

Última visita Enero 2012

[17] Sitio web de desarrollo del entorno ogre

<http://www.ogre3d.org/>

Última visita Enero 2012

[18] GarageGames LLC

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

Agosto 201. Última visita Enero 2012

[19] Sitio web de desarrollo del entorno nebula

<http://nebula.emulatronia.com/>

Última visita Enero 2012

[20] Conitec Datensysteme GmbH “3D GameStudio”

<http://www.3dgamestudio.com/>

Última visita Enero 2012

[21] Grupo Tecma Red S. “Tecnologías TCP/IP”

<http://www.casadomo.com/noticiasDetalle.aspx?c=27&idm=34>

Última visita Enero 2012

[22] Futurasmus-knxgroup “ComBridge Multi Control Gateway, de IPAS GmbH ®”

<http://descargas.futurasmus-knxgroup.org/DOC/ES/IPAS/3805/ComBridge%20esp.pdf>

Última visita Enero 2012

[23] Grupo Tecma Red S. “Productos KNX Lingg & Janke”

<http://www.casadomo.com/productosDetalle.aspx?id=443&idm=121&pat=121&act=&proy=>

Última visita Enero 2012

[24] Geeknet, Inc. “Calimero - KNXnet/IP ”

<http://calimero.sourceforge.net/>

Última visita Enero 2012

[25] Jung Electro Ibérica S.A “IP central unit Gateway”

[http://www.jung-catalogue.com/pdf/productdocumentation/en\\_ipz1000reg\\_td.pdf](http://www.jung-catalogue.com/pdf/productdocumentation/en_ipz1000reg_td.pdf) Última visita Enero 2012

[26] Futurasmus-knxgroup “Pasarela IP con Webserver de Lingg & Janke ®”

[http://descargas.futurasmus-knxgroup.org/DOC/ES/LINGG&JANKE/4645/Lingg-Janke\\_NK2\\_v202.pdf](http://descargas.futurasmus-knxgroup.org/DOC/ES/LINGG&JANKE/4645/Lingg-Janke_NK2_v202.pdf)

Última visita Enero 2012

[27] Lingg & Janke

<http://www.lingg-janke.de/index.php?id=56&L=1>

Última visita Enero 2012

[28] José María Quintero González, Javier Lamas Graziani y Juan D. Sandoval: “*Sistemas de control para viviendas y edificios: domótica*”, Thomson Paraninfo S.A., 1999.

# Pliego de condiciones

Para llevar a cabo este Proyecto Fin de Carrera ha sido necesario disponer de diversos elementos, tanto hardware como software, en cada una de las etapas y a lo largo de todo el desarrollo del proyecto<sup>1</sup>. En el presente Pliego de Condiciones se detallan los requisitos software y hardware necesarios para la realización de este PFC. Si se desean conocer más detalles sobre las características de los diversos elementos domóticos empleados consultar el anexo.

## P.C.1 Elementos Hardware

A continuación, se detallan todos los elementos hardware, tanto herramientas como materiales, que han sido utilizados y sus características específicas. Cada uno de los elementos domóticos usados en este PFC están homologados como elementos KNX. En cuanto a los recursos hardware para la realización del PFC, tanto para la programación del sistema 3D así como de los elementos domóticos han sido los siguientes:

---

<sup>1</sup>Como se puede ver, en este pliego de condiciones no se han considerado los conceptos asociados a la obra civil de instalación en una vivienda, puesto que en este proyecto no se ha realizado una infraestructura de este tipo. Todo lo concerniente a la parte del hogar ha sido sustituido por un demostrador domótico, sin que por ello se haya perdido generalidad o efectividad del proyecto desarrollado. Hay que tener en cuenta que el principal objetivo del proyecto es el desarrollo de la interfaz 3D y su conexión con una instalación domótica KNX para su control remoto. En ningún caso se ha contemplado en este proyecto la planificación diseño e implementación de una red domótica específica para una vivienda en particular.

- Un ordenador portátil HP Pavilion dv5 con un procesador Intel Core 2 Duo a 2,40GHz, 3GB de memoria RAM y 300 GB de disco duro. Este PC se ha utilizado para la programación de los elementos domóticos, la creación del sistema 3D así como la redacción de la memoria.
- Dimmer LuZenPlus ZN1DI-P400: Este aparato posee un canal de salida para la regulación de la iluminación de hasta 400W, permite cargas resistivas, capacitivas e inductivas. Se instala mediante un carril DIN. Al ser un producto homologado por KNX es sustituible por cualquier otro que cumpla esta misma condición.
- Shutter Switch N 521. Este elemento posee dos canales de salida que funcionan a 230 V y se ha usado para la regulación de las persianas de la vivienda. Se instala mediante un carril DIN y puede utilizarse para este fin cualquier producto de similares características que cumpla el estándar KNX.
- Nk2 network coupler. Este aparato tiene una cantidad máxima de direcciones de grupo de 111, trabaja a una tensión de 230V, se instala mediante un carril DIN y ha sido empleado para la conexión con la página web y el posible control de la vivienda mediante internet.
- Push button UP 234 + IR, 2-fold, neutral. Con este elemento se ha podido controlar tanto la regulación de la iluminación como las de las persianas de la vivienda. Está compuesto de dos botones y un receptor de infrarrojos que funciona en un radio de 25 metros. Se instala mediante un carril DIN y puede ser intercambiado por cualquier otro que cumpla el estándar KNX.
- Entrada binaria N 260. Este elemento se empleo para la adaptación de tanto un interruptor simple como de un pulsador simple en un sistema domótico. Está entrada binaria trabaja a una tensión de 230 V, posee cuatro posibles entradas y su instalación se realiza mediante un carril DIN. Estando homologado por KNX permite su sustitución por cualquier otro producto con similares características.

- Salida binaria N 562. Este aparato permitió la adaptación de un interruptor simple y un pulsador común en el sistema domótico de la vivienda. Posee dos salidas con un voltaje de 230 V, se instala mediante un carril DIN y al igual que los anteriores posee una homologación de KNX por lo cual puede ser sustituido por otra salida que cumpla el estándar KNX y, por tanto posea esa misma homologación.

## P.C.2 Elementos Software

Los elementos software que se han utilizado para el desarrollo de este PFC, son los indicados a continuación:

- Windows 7 Professional. Sistema operativo del PC en el que se redacta el PFC y en el que se procede a la programación y creación del sistema 3D.
- GameStudio Professional 6.11.4: Motor de juego mediante el cual se creó el sistema 3D.
- ETS3 Professional versión 3.0f: Programa domótico a partir del cual se programaron los diferentes aparatos de domotización.
- Nk2 Web Editor 2.0.2.12: Programa de edición web con el que se procedió a la construcción de la página web para el control de los elementos domóticos de la vivienda a través de internet.



# Presupuesto

## P.1. Introducción

En la elaboración del presupuesto del presente Proyecto Fin de Carrera se ha tenido en cuenta una serie de gastos asociados al mismo. Hay que destacar que en este presupuesto no se van a incluir los costes de instalación de la red domótica en una vivienda, puesto que se ha hecho un prototipo para generar un escenario mínimo para pruebas, pero no se ha instalado en una vivienda real. Así pues, los costes se han dividido en los siguientes apartados:

- Costes asociados a Recursos Humanos.
- Costes asociados a Recursos Hardware.
- Costes asociados a Recursos Software.
- Material fungible
- Derechos de visado del COIT
- Gastos de tramitación y envío
- Aplicación de impuestos.

A continuación, se detallará el coste asociado a cada uno de estas categorías y finalmente se indicará el coste total del PFC.

## P.2 Costes de Recursos Humanos

El coste de los Recursos Humanos está asociado al tiempo empleado por el personal necesario para la realización del presente Proyecto Fin de Carrera. Para la elaboración de este PFC, se ha estimado el empleo de un Ingeniero Junior durante unos 10 meses. Este tiempo comprende las etapas de formación, desarrollo y elaboración de la documentación. Según los baremos de honorarios orientativos facilitados por el Colegio Oficial de Ingenieros de Telecomunicación (COIT), el coste del trabajo en función del tiempo empleado viene dado por la Ecuación P.1:

$$H = C \times 75 \times H_n + C \times 95 \times H_e (\text{€})$$

*Ecuación P.1: Honorarios de un Ingeniero Junior*

**H** son los honorarios.

**C** es el coeficiente de corrección en función del número de horas trabajadas.

**H<sub>n</sub>** es el número de horas normales trabajadas (dentro de la jornada laboral).

**H<sub>e</sub>** es el número de horas especiales trabajadas.

La carga laboral del Ingeniero ha sido de 8 horas diarias a razón de 20 días mensuales durante 10 meses, por lo tanto el número total de horas normales trabajadas asciende a:

$$H_n = 8 \times 20 \times 10 = 1.600 \text{ horas.}$$

El número de horas especiales asciende a 0, dado que no se realizaron trabajos fuera de horario laboral:

$$H_e = 0 \text{ horas.}$$

El valor de este coeficiente de corrección se extrae de una tabla especificada por el COIT y que se presenta en la Tabla P.1:

<b>Horas</b>	<b>Factor de corrección</b>
<b>Hasta 36</b>	<b>1</b>
<b>Exceso de 36 hasta 72</b>	<b>0,9</b>
<b>Exceso de 72 hasta 108</b>	<b>0,8</b>
<b>Exceso de 108 hasta 144</b>	<b>0,7</b>
<b>Exceso de 144 hasta 180</b>	<b>0,65</b>
<b>Exceso de 180 hasta 360</b>	<b>0,6</b>
<b>Exceso de 360 hasta 510</b>	<b>0,55</b>
<b>Exceso de 510 hasta 720</b>	<b>0,5</b>
<b>Exceso de 720 hasta 1080</b>	<b>0,45</b>
<b>Exceso de 1080</b>	<b>0,5</b>

*Tabla P1. Coeficiente de corrección*

Dado que exceden de 1.080 horas de trabajo, es necesario aplicar un coeficiente corrector de 0,5 sobre el número de horas trabajadas, de acuerdo con los datos expuestos en los baremos propuestos por el COIT. Así pues, el coste total de los honorarios asciende a:

$$H = C \times 75 \times H_n + C \times 95 \times H_e = 0,5 \times 75 \times 1600 + 0,4 \times 95 \times 0 = 60.000,00 \text{ (€)}$$

Por otro lado, debido a las exigencias hardware y software de la realización del presente Proyecto Fin de Carrera, se ha considerado que no es necesario un técnico de mantenimiento, ya que al tratarse de un solo equipo, el mantenimiento del mismo así como de la red y del software utilizado puede ser realizado por el Ingeniero Junior

En la Tabla P.2 se recogen todos los costes asociados a los recursos humanos

Concepto	Tiempo empleado	Coste mensual	Importe
<b>Ingeniero Junior</b>	<b>10 meses</b>	<b>12.000,00 €</b>	<b>120.000,00 €</b>
<b>Coste corregido por superar las 1.080 horas</b>			<b>60.000,00 €</b>
<b>Coste Total</b>			<b>60.000,00 €</b>

*Tabla P.2: Costes asociados a los Recursos Humanos*

En definitiva, el coste de recursos humanos asciende a un total de sensata mil euros con cero céntimos de euro (60.000,00 €).

### **P.3.Coste de Recursos Materiales**

Entre los recursos materiales utilizados para la realización de este PFC se incluyen las herramientas software para la redacción de la memoria y para llevar a cabo las diferentes tareas de este proyecto. Se consideran también los equipos hardware empleados para dar soporte a estas herramientas.

Los recursos materiales se pueden clasificar como fungibles y amortizables. La amortización se reparte entre todos los periodos en que permanece el recurso material.

En este caso, se establece el coste de amortización para un periodo de entre uno y tres años, según el recurso del que se trate. Para ello, se utilizará un sistema de amortización lineal, en el que se supone que el inmovilizado material se deprecia de forma constante a lo largo de su vida útil. La cuota de amortización anual se calcula usando la siguiente fórmula:

$$\text{Cuota} = \frac{\text{Valor Adquisición} - \text{Valor Residual}}{\text{Periodo Amortización}}$$

donde el *Valor Residual* se refiere al valor que tendrá el bien una vez finalice su periodo de vida útil, estimado para este caso en 0 €

### **P.3.1. Costes de Recursos Hardware**

El coste de los Recursos *Hardware* viene determinado por los equipos informáticos empleados en la realización del presente Proyecto Fin de Carrera, listados a continuación:

- Ordenador portátil
- Impresora
- Aparatos domóticos

Dado que algunos de estos recursos son compartidos entre varios usuarios, el coste asociado debe calcularse en función del número de personas que los utilizan y del período de amortización aplicado. En este caso, se ha estimado un período de amortización de tres años para la impresora y se ha supuesto que es utilizada por 30 personas, el ordenador

portátil, por el contrario, es de uso exclusivo personal. Su período de amortización ha sido fijado en dos años.

En la tabla P.3 se recogen todos los costes asociados a los Recursos *Hardware* que se han utilizado. El importe del recurso es igual al producto de su coste unitario por el tiempo durante el cual ha sido empleado, dividido a su vez por el periodo de amortización y por el número de personas que han hecho uso de él.

Concepto	Coste unitario	Amortización	Tiempo empleado	Nº de personas con acceso	Importe
<b>Ordenador portátil</b>	<b>900,00 €</b>	<b>24 meses</b>	<b>10 meses</b>	<b>1</b>	<b>375,00 €</b>
<b>Impresora</b>	<b>2.400,00 €</b>	<b>36 meses</b>	<b>10 meses</b>	<b>30</b>	<b>22,23 €</b>
<b>Coste Total</b>					<b>397,23€</b>

*Tabla P.3: Costes asociados a los Recursos Hardware*

El coste total de los Recursos *Hardware* asciende a un total de trescientos noventa y siete euros con veintitrés céntimos de euro.

### **P.3.2.Costes de Recursos Software**

El coste de los Recursos *Software* se obtiene a partir del valor de las licencias y del mantenimiento de cada uno de los programas utilizados en este Proyecto Fin de Carrera. Se ha supuesto una amortización de 24 meses.

Concepto	Coste unitario	Amortización	Tiempo empleado	Importe
Microsoft Office 2007	610,00 €	24 meses	10 meses	254,20 €
ETS3	950,00 €	24 meses	10 meses	395,83 €
GameStudio	671,00€	24 meses	10 meses	279,60 €
Nk2	0,00 €	-	10 meses	0,00 €
Windows 7 OS	319,00 €	24 meses	10 meses	132,91 €
<b>Coste Total</b>				<b>1062,54 €</b>

*Tabla P.4: Costes asociados a los Recursos Software*

En la Tabla P.4 se recogen los costes asociados a cada uno de los Recursos *Software*, además del total, que es igual a mil sesenta y dos euros con cincuenta y cuatro céntimos de euro.

### **P.3.3.Material fungible**

A los gastos calculados en los apartados anteriores hay que unirles el coste asociado al material fungible y otros gastos generales. Los aparatos domésticos, dado que se quedarían en la vivienda correspondiente únicamente se considera el gasto de compra del aparato y, por tanto, se considerarían material fungible. Estos se recogen en la tabla P.4:

Concepto	Cantidad	Coste unitario	Importe
<b>Paquete de papel DIN-A4</b>	<b>4</b>	<b>4,00 €</b>	<b>16,00 €</b>
<b>Cartuchos de tinta de impresora</b>	<b>2</b>	<b>50,00 €</b>	<b>100,00 €</b>
<b>Encuadernación</b>	<b>3</b>	<b>60,00 €</b>	<b>180,00 €</b>
<b>Dimmer</b>	<b>1</b>	<b>155,63 €</b>	<b>155,63 €</b>
<b>Shutter</b>	<b>1</b>	<b>305,70 €</b>	<b>305,70 €</b>
<b>Nk2 network coupler</b>	<b>1</b>	<b>665,33 €</b>	<b>665,33 €</b>
<b>Entrada binaria</b>	<b>1</b>	<b>262,33 €</b>	<b>262,33 €</b>
<b>Salida binaria</b>	<b>1</b>	<b>302,80 €</b>	<b>302,80 €</b>
<b>Pulsador</b>	<b>1</b>	<b>183,99 €</b>	<b>183,99 €</b>
<b>Conectores</b>	<b>20</b>	<b>0,30 €</b>	<b>3,00 €</b>
<b>Cables</b>	<b>8 m</b>	<b>2,00 €</b>	<b>16,00 €</b>
<b>Bombillas</b>	<b>5</b>	<b>2,00€</b>	<b>10,00€</b>
<b>Coste Total</b>			<b>2200,78 €</b>

*Tabla P.5: Costes asociados a material fungible*

El coste total atribuido a otros gastos del Proyecto Fin de Carrera asciende a un total de dos mil doscientos euros con setenta y ocho céntimos de euro.

## P.4. Beneficio industrial

Todo proyecto conlleva un beneficio industrial que justifique la realización del mismo. Para este Proyecto Fin de Carrera, se ha estimado este beneficio en un 10% de los gastos totales.

## P.5. Redacción del Proyecto Fin de Carrera

De acuerdo a los honorarios orientativos del COIT, el importe de la redacción del presente PFC se calcula mediante la siguiente ecuación:

$$R=0,07 \times P_{tel} \times Cr + 0,03 \times P_{civil} \times Cr$$

Donde  $P_{tel}$  es el presupuesto de ejecución material de telecomunicaciones,  $P_{civil}$  es el presupuesto de obra civil y  $Cr$  es el coeficiente de ponderación por tramos en función del coste del presupuesto. Este Proyecto Fin de Carrera no tiene asociada ninguna obra civil, por lo que  $P_{civil}$  es nulo. Mientras que  $P_{tel}$  se corresponde con la suma de los apartados anteriores.

$$P_{tel} = 60.000 + 2273,01 + 1062,54 + 296 = 63.631,55 \text{ €}$$

Para este valor de  $P_{tel} = 63.631,55 \text{ €}$  el coeficiente  $Cr$  establecido por el COIT en este caso vale 0,8, ya que es el valor estipulado para presupuestos de hasta 90.152 € por lo que aplicando la ecuación anterior se tiene que:

$$R=0,07 \times 63.631,55 \times 0,8 = 3563,36 \text{ €}$$

Por tanto, los costes<sup>2</sup> asociados a la redacción del proyecto, libres de impuestos, ascienden a tres mil quinientos sesenta y tres euros con treinta y seis céntimos.

## P.6.Visado del COIT

Los gastos de visado del COIT se tarifican mediante la siguiente expresión:

$$V=0,006 \times P \times C_v$$

Donde  $P$  es el presupuesto que se obtiene sumando las cantidades obtenidas en los apartados anteriores, y  $C_v$  es un coeficiente reductor en función del presupuesto. Hasta el momento el presupuesto del proyecto  $P$  calculado asciende a la cifra de 67.194,91 €. Como el valor del coeficiente  $C_v$ , para presupuestos de más de 30.050 € y menos de 90.150 € viene definido por el COIT con un valor de 0,9, el coste de los derechos de visado de este PFC se obtiene de la siguiente forma:

$$V= 0,006 \times 67.194,91 \times 0,9= 362,85 \text{ €}$$

El coste de los derechos de visado del proyecto asciende a la cantidad de trescientos sesenta y dos euros con ochenta y cinco céntimos.

---

<sup>2</sup> Como se puede ver, en este presupuesto no se han considerado los conceptos asociados a la obra civil de instalación en una vivienda, puesto que en este proyecto no se ha realizado una infraestructura de este tipo. Todo lo concerniente a la parte del hogar ha sido sustituido por un demostrador domótico, sin que por ello se haya perdido generalidad o efectividad del proyecto desarrollado. Hay que tener en cuenta que el principal objetivo del proyecto es el desarrollo de la interfaz 3D y su conexión con una instalación domótica KNX para su control remoto. En ningún caso se ha contemplado en este proyecto la planificación diseño e implementación de una red domótica específica para una vivienda en particular.

## P.7.Gastos de envío

Los gastos de tramitación y envío son fijos, y se estipulan por el COIT en 9 €

## P.8.Aplicación de impuestos

La realización del proyecto está gravada con el Impuesto General Indirecto Canario (IGIC). Para la actividad económica en la que se encuadra, el IGIC correspondiente es de un cinco por ciento (5%) del valor presupuestado. Por tanto, al presupuesto calculado hay que añadirle el IGIC para obtener el coste total. Como el presupuesto calculado de este PFC es de € el valor del impuesto asciende a €

## P.6.Coste total del Proyecto Fin de Carrera

El cálculo del presupuesto de este PFC se realiza a partir de los costes hallados en los apartados anteriores, incluyendo el beneficio industrial y los impuestos pertinentes. En la Tabla P.6 se recogen estas partidas:

Concepto	Coste
<b>Recursos Humanos</b>	<b>60.000,00 €</b>
<b>Recursos Hardware</b>	<b>397,23€</b>
<b>Recursos Software</b>	<b>1062,54€</b>
<b>Material fungible</b>	<b>2200,78 €</b>
<b>Redacción del proyecto</b>	<b>3563,36 €</b>

<b>Derechos de visado del COIT</b>	<b>362,85€</b>
<b>Tramitación y envío</b>	<b>9 €</b>
<b>Coste acumulado</b>	<b>67.595,76 €</b>
<b>Beneficio industrial (10%)</b>	<b>6759,576 €</b>
<b>Subtotal</b>	<b>74.355,33 €</b>
<b>IGIC (5%)</b>	<b>3717,76 €</b>
<b>Total</b>	<b>78.073,09 €</b>

*Tabla P.6: Costes total del proyecto*

Dña. María Ferragut Fiol declara que el presupuesto del proyecto “*Entorno 3d Interactivo para Control de Redes Domóticas*” asciende a un total de **setenta y ocho mil setenta y tres euros con nueve céntimos de euro (78.073,09 €)**

Fdo.: María Ferragut Fiol

Las Palmas de Gran Canaria, de 2012

# Anexo



# Anexo (hojas de características de los elementos domóticos)

## Actuador persianas

### N521

Shutter Switch N 521  
2 x 230 V AC / 6 A

5WG1 521-1AB01

#### Product and Applications Description



The shutter switch N 521 is a N-system DIN-rail mounted device. Via its volt free contacts it can control two separate blinds actuators to raise and lower blinds and to gradually adjust the louvres.

Two channels (A and B) are available for two actuators each allowing control of up to four blind actuators. The actuator control outputs are connected quasi-parallel in pairs and locked via internal relay contacts to avoid mutual electrical interference when connecting two blind actuators. Otherwise, when one of the two blinds has reached an extreme position while the other is still moving, the windings of the inactive blinds actuator would be fed power via the capacitor of the actuator still running, moving it from the extreme position automatically.

The volt free contacts (two relays) can be assigned various tasks depending on the application program used, i.e. the shutter switch N 521 consists of the device (hardware) and its application programs (software) for controlling venetian blinds.

In the ETS (EIB Tool Software) the application program is selected, its parameters and addresses are assigned appropriately, and downloaded to the shutter switch N 521.

#### Application Programs

##### 11 A2 Shutter 520206

- 2 blinds/sliding shutter control with safety mode
- safety position (up or down) if in bus voltage failure can be set in the parameter list
- may be used as 2 standard binary output with two mutually independent change-over contacts

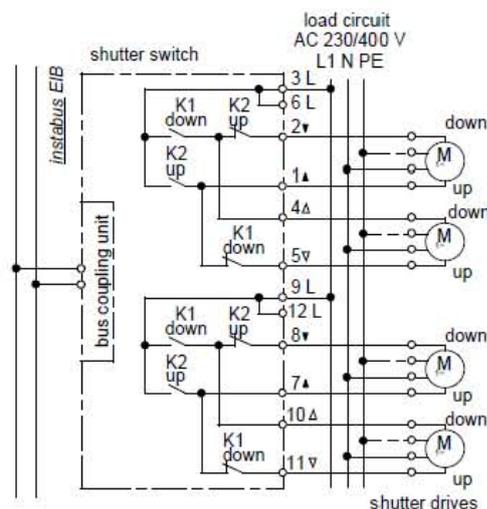


#### WARNING

When using the shutter switch N 521 for controlling blinds actuators the instructions below must be followed (failure to do so may result in the contacts welding):

- Do not use older application programs (e.g. 11 A2 Shutter 520203) anymore with the shutter switch N 521, use 11 A2 Shutter 520205 instead.
- The factor and base values of the parameter "Reverse delay" must be combined to establish a time period as given by the blinds manufacturer (usually > 500 ms).
- The parameter "on bus voltage failure" should be set to "maintain actual state" or "stop".

#### Example of Operation



**Shutter Switch N 521**  
2 x 230 V AC / 6 A

**5WG1 521-1AB01**

### Installation Instructions

- The device may be used for permanent interior installations in dry locations within distribution boards.



#### WARNING

- The device may be built into distribution boards (230/400V) together with appropriate VDE-devices and must be mounted and commissioned by an authorised electrician.
- A safety disconnection of the device must be possible.
- Free DIN rail areas must be covered with covers, order no. 5WG1 192-8AA01.
- The prevailing safety rules must be heeded!
- The device must not be opened. A device suspected faulty should be returned to the local Siemens office.

### Technical Specifications

#### Power supply

via bus cable

#### Outputs

- number: 2 output channels (voltage free contacts)
- rated voltage: AC 230 V, 47 ... 63 Hz
- rated current: 6 A resistive load
- switching current at AC 230 V:
  - 0,01 ... 6 A resistive load
  - tubular motors with auxiliary phase condenser  $\leq 14 \mu\text{F}$ , max. power 500 VA at 20000 load switching cycles (UP-DOWN-STOP) respectively max. 750 VA at 12000 load switching cycles
  - total maximum load at  $\cos\phi = 0,4$ ; 750 VA
- switching current at DC 24 V:
  - 6 A resistive load,
  - 4 A inductive load ( $L/R = 7 \text{ ms}$ )
- switching characteristic: set in parameter list according to application program
- switching repetition intervals: min. 150 ms

#### Control elements

1 learning button:  
for switching between normal operating mode and addressing mode

#### Display elements

1 red LED:  
for monitoring bus voltage and displaying mode, selected with the learning button

### Connections

- load circuit, physical:
  - strip insulation for 9 ... 10 mm
  - permissible conductor types/cross sections:
    - 0,5 ... 2,5 mm<sup>2</sup> single core or flexible conductor, 8 mm ultrasonically compacted
    - 0,5 ... 2,5 mm<sup>2</sup> flexible conductor with terminal pin, crimped on gas tight
    - 0,5 ... 1,5 mm<sup>2</sup> flexible conductor with connector sleeve
    - 1,0 and 1,5 mm<sup>2</sup> plain flexible conductor
- load circuit, electrical:
  - plain flexible conductor, min. 1 mm<sup>2</sup>: current carrying capacity max. 6 A
  - all other conductors, min. 1,5 mm<sup>2</sup>: current carrying capacity max. 10 A
  - The load circuits must be protected with a 10 A miniature circuit breaker A or B characteristic.



#### WARNING

When looping through the L-conductor (connection blocks 3 and 6, 9 and 12), take care that the maximum connection current of 10 A (as governed by the maximum permissible printed conductor load) is not exceeded!

- bus line, pressure contacts on data rail

### Physical specifications

- housing: plastic
- N-system DIN-rail mounted device, width: 3 SUs (1 SU = 18 mm)
- weight: approx. 225 g
- fire load: approx. 3600 kJ  $\pm$  10 %
- installation: rapid mounting on DIN EN 50022-35 x 7,5 rail

### Electrical safety

- fouling class (according to IEC 664-1): 2
- protection (according to EN 60529): IP 20
- overvoltage class (according to IEC 664-1): III
- bus: safety extra low voltage SELV DC 24 V
- relay with  $\mu$ -contact
- the device complies with EN 50090-2-2 and EN 60669-2-1

### Reliability

rate of failure: 522 fit at 40 °C

### Electromagnetic compatibility

complies with  
EN 50081-1, EN 50082-2 and EN 50090-2-2

**Shutter Switch N 521**  
2 x 230 V AC / 6 A

5WG1 521-1AB01

#### Environmental specifications

- climatic conditions: EN 50090-2-2
- ambient temperature operating: - 5 ... + 45 °C
- ambient temperature non-op.: - 25 ... + 70 °C
- relative humidity (non-condensing): 5 % to 93 %

#### Certification

EIB certificate

#### CE norm

complies with the EMC regulations (residential and functional buildings), and low voltage regulations

#### Location and Function of the Display and Operator Elements

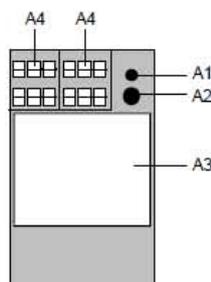


Figure 1: Location of the display and operator elements

- A1 LED for indicating normal operating mode (LED off) and addressing mode (LED on); upon receiving the physical address the device automatically returns to normal operating mode
- A2 Learning button for switching between normal operating mode and addressing mode for receiving the physical address
- A3 Type plate
- A4 Screwless plug-in terminals for connecting load circuits

#### Mounting and Wiring

##### General description

The N-system DIN-rail device (3 SUs) can be installed to N-system distribution boards, surface or flush mounted, or to any DIN-rail EN 50022-35 x 7,5 available that has a data rail installed.

The connection to the bus line is established by clicking the device onto the DIN-rail (with a data rail installed). Take care that the type plates of all devices on a DIN-rail can be read in the same direction, guaranteeing the devices are polarised correctly.

##### Mounting DIN-rail devices (Figure 2)

- Slide the device (B1) onto the DIN-rail (B2) and
- swivel back the device until the slide clicks into place audibly.

##### Dismounting DIN-rail devices (Figure 2)

- Remove all connected wires,
- press down the slide (C3) with a screw-driver and
- swivel the device (C1) from the DIN-rail (C2).

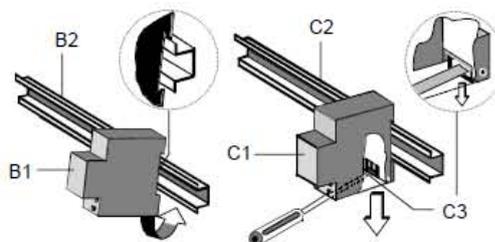


Figure 2: Mounting and dismounting a DIN-rail device

##### Connecting load circuits (Figure 3)

- The load circuits are connected via screwless plug-in terminals (D1).
- Remove approx. 9 to 10 mm of insulation from the wire (D1.1) and plug it into the terminal (D1).

**Shutter Switch N 521**  
**2 x 230 V AC / 6 A**

**5WG1 521-1AB01**

**Conductor cross sections:**

- load circuit, physical:
  - strip insulation for 9 ... 10 mm
  - permissible conductor types/cross sections:
    - 0,5 ... 2,5 mm<sup>2</sup> single core or flexible conductor, 8 mm ultrasonically compacted
    - 0,5 ... 2,5 mm<sup>2</sup> flexible conductor with terminal pin, crimped on gas tight
    - 0,5 ... 1,5 mm<sup>2</sup> flexible conductor with connector sleeve
    - 1,0 and 1,5 mm<sup>2</sup> plain flexible conductor
- load circuit, electrical:
  - plain flexible conductor, min. 1 mm<sup>2</sup>: current carrying capacity max. 6 A
  - all other conductors, min. 1,5 mm<sup>2</sup>: current carrying capacity max. 10 A

**Disconnect load circuits (Figure 3)**

- Press the terminal lock (E1.2) with a screw-driver and
- remove the wire (E1.1) from the terminal (E1).

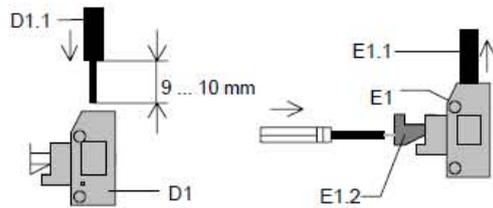
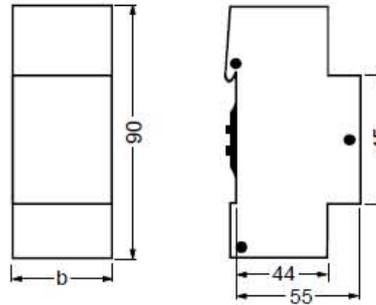


Figure 3: Connecting and disconnecting wires

**Dimension Diagram**

Dimensions in mm



b = 3 SU

# Actuador dimmer universal Luzen Plus



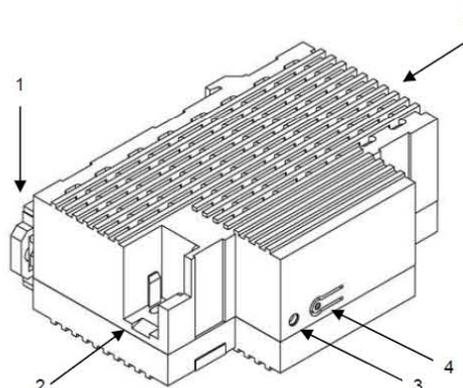
Luzen Plus

Luzen Plus. Actuador DIMMER Universal para regulación de iluminación  
ZN1DI-P400

Documentación Técnica

## CARACTERÍSTICAS PRINCIPALES

- Tamaño reducido: 90 x 60 x 35mm (2 unidades de carril DIN).
- 1 Canal de hasta 400W @25°C (230V – 50Hz)
- Detección automática del tipo de carga.
- Unidad de Acoplamiento al BUS KNX integrada.
- Diseñado para ser ubicado, bien en una caja de registro (caja de empalmes), bien en cualquier envoltorio eléctrica con carril DIN.
- Salvado de Datos Completo en caso de pérdida de alimentación.
- Conforme a la Directivas CE



**Pulsador de test/programación:** permite seleccionar el MODO PROGRAMACIÓN o el MODO TEST. Si se pulsa inicialmente, tras aplicar la tensión de bus, fuerza al aparato a colocarse en "MODO SEGURO". Si se mantiene pulsado durante más de 3 segundos, estando el actuador conectado al bus KNX, permite entrar o salir en modo test. Dentro del modo test, permite conmutar On/Off la carga.

**LED de test/programación:** indica que el aparato está en modo programación (color rojo). Cuando el aparato entra en modo seguro parpadea con un periodo de 0,5seg (color rojo). El modo test se indica en color verde.

1. Anclaje carril DIN
2. Conexión bus KNX
3. LED Programación/Test
4. Pulsador Programación/Test
5. Clema conexión carga (Carga, Neutro y Fase)

## ESPECIFICACIONES

Tipo de Dispositivo	Dispositivo de Control de funcionamiento eléctrico	
Alimentación KNX	Tensión de Operación	29V DC típicos
	Margen de Tensión	20...31V DC
	Consumo	150mW
	Tipo de Conexión	Conector típico de bus para TP1, 0,50 mm <sup>2</sup> de sección.
Alimentación Externa	<b>230V-50Hz</b>	
Máxima Capacidad de Carga	400W @ 25°C	
Mínima Capacidad de Carga	50W	
Tipo de Accionamiento del Dispositivo	Tipo I	
Periodo de Solicitaciones Eléctricas	Largo	
Grado de Contaminación	IP 20, Ambiente Limpio	
Temperatura de Trabajo	-5 °C a +45 °C	
Temperatura de Almacenamiento	-20 °C a +70 °C	
Humedad Relativa	30 a 85% HR (Sin condensación)	
Humedad Relativa de Almacenamiento	30 a 85% HR (Sin condensación)	
Montaje	Dispositivo de control de montaje independiente para montaje en el interior de cuadros eléctricos y envoltorios de empalmes y/o registro eléctricos	
Respuesta en caso de fallo de alimentación (bus).	Salvado de datos	
Indicador de operación	Al pulsar el botón de programación, debe encenderse el LED de programación (rojo). Al realizar una pulsación larga (> 3sg) debe encenderse el LED de modo test (verde)	
Peso aproximado	80 gr.	
Índice CTI de la PCB	175 V	
Material de la carcasa	PC-ABS, categoría de inflamabilidad clase D	



Luzen Plus

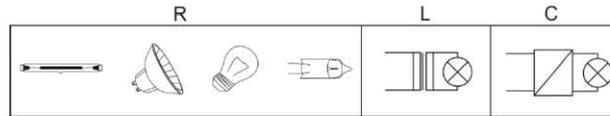
Luzen Plus. Actuador DIMMER Universal para regulación de iluminación  
ZN1DI-P400

Documentación Técnica

## CARGAS PERMITIDAS

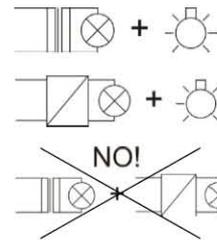


- R= Resistivas
- L= Inductivas
- C= Capacitivas

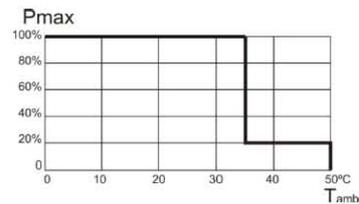


## MEZCLA DE CARGAS

- Se pueden mezclar cargas resistivas convencionales con cargas con transformador magnético. En ese caso, la parte resistiva de la carga no debe superar el 50%.
- Se pueden mezclar cargas resistivas convencionales con cargas con transformador electrónico. En ese caso, la parte resistiva de la carga no debe superar el 50%.
- **Prohibido terminantemente mezclar cargas con transformador electrónico convencional en cualquier proporción.**



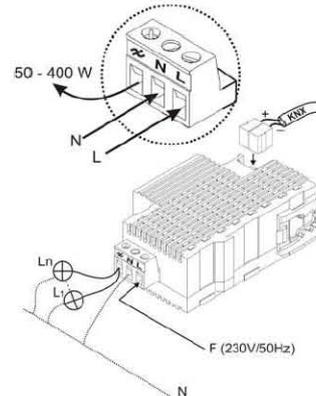
## PROTECCIÓN POR SOBRECALENTAMIENTO



- Regulación Automática de la Carga cuando la Tª Ambiente es excesiva. Nivel de regulación máximo: 20%.
- Una vez recuperada de nuevo la temperatura adecuada, el dispositivo retorna a su modo de funcionamiento normal. Ver manual de programación.

## CONEXIONADO DE SALIDA

<b>Tipo de Contacto</b>	Dispositivo de interrupción mediante Semiconductor
<b>Protección de la Carga</b>	Sí, por sobrecarga, sobre calentamiento y cortocircuito
<b>Capacidad de conmutación por salida</b>	400W @25°C (230V-50Hz)
<b>Caída de tensión asumible</b>	Despreciable
<b>Método de Conexión</b>	Bloque de Terminales (Tornillo)
<b>Sección de Cable</b>	1,5 mm <sup>2</sup> a 2,5 mm <sup>2</sup>
<b>Tipo de Cable</b>	Flexible con terminales (punteras) o Rígido
<b>Tiempo de respuesta</b>	Despreciable



## INSTRUCCIONES DE SEGURIDAD

- No se debe conectar el voltaje principal (230 V) u otros voltajes externos a ninguno de los puntos del bus KNX. Conectar un voltaje externo puede poner en peligro la seguridad eléctrica de todo el sistema KNX.
- Una vez instalado, el dispositivo no debe ser accesible desde el exterior.
- El equipo debe ser instalado y ajustado únicamente por electricistas cualificados y siguiendo las regulaciones aplicables de prevención de accidentes.
- Se debe asegurar durante la instalación que hay el suficiente aislamiento entre los conductores del voltaje principal de 230V y los conductores del bus KNX o sus extensiones.
- No exponga este aparato a la lluvia o a la humedad.
- No bloquee las aperturas de ventilación
- El hecho de no tener en cuenta estas instrucciones de instalación puede causar fuego y otros daños.

# EibSolo Network Coupler

Ling & Janke

## eibSOLO

*Network Coupler*

**NK1 / NK2 / NK-FW**

**Description**

The eibSOLO network couplers NK are DIN rail-mounted devices featuring the following:

- web server
- FTP server
- EIB visualisation with 104 data points maximum
- EIB programming via IP (EIBip/IP)

The visualisation is exclusively configured with ETS (group addresses/data types) and with a standard browser in the visualisation itself (labelling/page links).

The NK2 network coupler additionally allows separate visualisations to be created and uploaded into the device. The NK-FW allows access to all other devices of FacilityWeb.

**Technical Data**

max. group addresses	111 (dynamic)
max. associations	111 (dynamic)
power supply	230V / 50Hz network interface 24V DC via EIB
protection class	IP 20
dimensions	108 x 90 x 65 mm (6 RU*)
installation	35 mm DIN rail
operating temperature	-5 ... +45°C

\*RU = rail unit

**Terminals**

- terminal cross section: 0.08 - 2.5 mm<sup>2</sup>
- stripping length: 5 - 6 mm
- conductors permitted:
  - single core
  - multi-filar
  - fine-wired, including tin-plated individual wires
  - fine-wired, with wire end sleeves
- network: standard RJ45
- EIB: red-black bus terminal

D2 = network-collision  
 D3 = Network Link  
 D4 = 10Mbit  
 D5 = 100Mbit

network 10 / 100 Mbit RJ45  
 IP address standard / ETS  
 IP 192.168.0.2  
 IP user config

L  
N  
PE

EIB +

**factory settings (upper position of switch):**

**IP 192.168.0.2**  
 user: user / password: Nk1

**factory settings:**

**HTTP Server:**  
<http://192.168.0.2>

**FTP Server:**  
<ftp://192.168.0.2>

**visualisation:**  
<http://192.168.0.2/visu>

**labelling of standard visualisation:**  
<http://192.168.0.2/visuconfig>

**date/time adjustment NK V2.0:**  
<http://192.168.0.2/date>

**Warnings**

The device must only be installed and configured by a qualified professional!

Health and safety regulations have to be complied with!

Do not open the device!

A faulty device must be returned immediately to Ling & Janke OHG!

**Installation**

The device is mounted on a DIN rail, DIN EN 60715 TH35

Position the device on the DIN rail from above. Apply brief, strong pressure on the lower edge of the casing to engage the casing with the rail.

The device can be removed from the rail without any tools: simply slide it from the DIN rail upwards and remove it from the top of the rail. Do not apply any force lest the clamps be damaged.

To connect the wires to the screwless terminals, insert a slotted screwdriver into the respective mounting hole under the terminal, which opens the terminal. Insert the wire into

**Configuration Network**

A slide switch in the upper part of the device allows the selection of the IP address.

The upper position of the switch selects the pre-programmed IP address **192.168.0.2**. The factory login settings are for user: **user** and for password: **Nk1**.

The lower position activates the IP address parameterised with ETS.

These options allow access to the pre-programmed address if the customised IP address is not known, for example, for service or configuration purposes.

The device is connected to the network with a standard RJ45 connection. The data transfer rate (10/100Mbit) is automatically detected and is displayed by a LED.

**Configuration EIB**

The factory settings of the NK1 / NK2 do not feature any device or group addresses. The functions required are assigned when setting the parameters. During the planning phase with ETS, objects which are not assigned are not displayed either.

**important:**  
 The bus coupling unit (BCU 2.1) used in the NK1 / NK2 requires the following to be installed before first-time use of the device:

**programming only with ETS 2.0 V1.2 or later**

- product data base 12/2004 or later
- current service patch

The application program must always be fully downloaded to the device, never partially. Partial download of the program may lead to malfunctions.

Ling & Janke OHG  
 Zeppelinstraße 30  
 78315 RADOLFZELL  
 GERMANY

technical support:  
 tel. (+49) 7732 - 94557-71  
 www.ling-janke.de

© Del documento, de los autores. Digitalización realizada por ULPGC. Biblioteca universitaria, 2013

[211]



# Pulsador

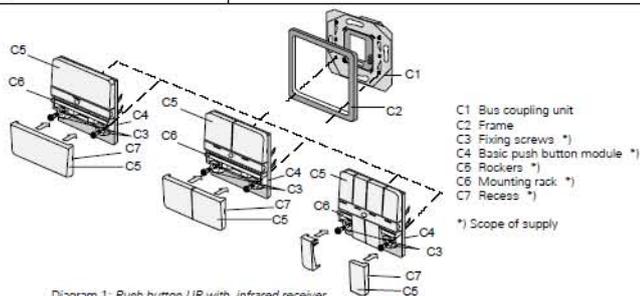
**SIEMENS**



Push button UP 233+IR, 1-fold 5WG1 233-2AB\_1 Push button UP 285+IR, 1-fold 5WG1 285-2EB\_1  
 Push button UP 234+IR, 2-fold 5WG1 234-2AB\_1 Push button UP 286+IR, 2-fold 5WG1 286-2EB\_1  
 Push button UP 235+IR, 4-fold 5WG1 235-2AB\_1 Push button UP 287+IR, 4-fold 5WG1 287-2EB\_1  
 - with infrared receiver

Issued: September 2002

Operating and mounting instructions		
Product	DELTA profil	DELTA style
Push button 1-fold + IR 	pearl grey 5WG1 233-2AB01 titanium white 5WG1 233-2AB11 anthracite 5WG1 233-2AB21 silver 5WG1 233-2AB71	titanium white 5WG1 285-2EB11 basalt black 5WG1 285-2EB21 titanium white metallic silver 5WG1 285-2EB81 basalt black metallic silver 5WG1 285-2EB01
Push button 2-fold + IR 	pearl grey 5WG1 234-2AB01 titanium white 5WG1 234-2AB11 anthracite 5WG1 234-2AB21 silver 5WG1 234-2AB71	titanium white 5WG1 286-2EB11 basalt black 5WG1 286-2EB21 titanium white metallic silver 5WG1 286-2EB81 basalt black metallic silver 5WG1 286-2EB01
Push button 4-fold + IR 	pearl grey 5WG1 235-2AB01 titanium white 5WG1 235-2AB11 anthracite 5WG1 235-2AB21 silver 5WG1 235-2AB71	titanium white 5WG1 287-2EB11 basalt black 5WG1 287-2EB21 titanium white metallic silver 5WG1 287-2EB81 basalt black metallic silver 5WG1 287-2EB01
Frame	ordered separately from the DELTA ranges	
Bus coupling unit	cut-out frames	UP 114



<p><b>Product and functional description</b></p> <p>The device push button 1; 2; 4-fold + IR is composed of 2 functional units: the push button and the infrared receiver.</p> <ul style="list-style-type: none"> <li>push button There is an upper and a lower operating field on the push buttons. In the middle of the push button there is a note panel in which pictograms can be inserted. This panel also contains display elements (LEDs) for orientation lighting and also for status displays. Opposing rockers are combined to form a pair e.g. for defined switching, dimming, controlling shutters and blinds. It is also possible to carry out the switching functions ON and OFF e.g. via a push button "UP".</li> <li>infrared (IR) receiver The IR receiving lens is incorporated in the note panel in the middle of the push button. Thus for carrying out the functions direct telegrams can also be transmitted onto the bus by IR signals, sent by a hand-held transmitter S 425 and/or a wall-mounted transmitter UP 420/421/422.</li> </ul> <p>Using an application program, the push buttons UP + IR give commands via the flush-mounted bus coupler for example to actuators for defined switching on/off, for dimming lamps, raising/lowering shutters or for louvre adjustment or other parameterisable functional units.</p> <p>The device push buttons + IR is placed together with the relevant DELTA frame on the flush-mounted bus coupler and can only function in combination with the bus coupler UP 114 and an appropriate application program, i.e. the push buttons (with bus coupler UP 114) consist of the devices (hardware) and the application programs (software).</p> <p>The bus coupler UP 114 and the relevant frame are not supplied with the device but must be ordered separately.</p> <p>Using the ET3 program (EIB Tool Software), the application programs can be selected and the specific parameters and addresses can be assigned.</p>	<p><b>Application programs</b></p> <p>see Siemens product database from version H onward or: <a href="http://www.siemens.de/Installationstechnik">http://www.siemens.de/Installationstechnik</a></p> <p><b>Technical data</b></p> <p><b>Power supply</b> Via the flush-mounted bus coupler</p> <p><b>Operating elements</b></p> <ul style="list-style-type: none"> <li>1, 2 or 4 pairs of rockers The pairs of rockers are interlocked via software so that malfunctions are not triggered when they are operated simultaneously.</li> <li>Number of switching cycles: &gt; 20,000</li> </ul> <p><b>Display elements</b></p> <ul style="list-style-type: none"> <li>Per pair of push buttons 1 LED red for status display e.g. 4fold-push buttons = 4 LED red. The parameter for a red LED can be set to "flashing" if a proper incoming IR telegram is received / recognised. Each LED can be parameterised for status display or as an orientation light.</li> </ul> <p><b>IR receiver</b></p> <ul style="list-style-type: none"> <li>Range of infrared beam: approx. 25 m if the following conditions are met:                             <ul style="list-style-type: none"> <li>with IR hand-held transmitter AP 425 (5WG1 425-7AB2)</li> <li>directed at the optical main axis,</li> <li>with 500 lux of diffuse daylight at the receiver</li> </ul> </li> <li>Device without pictogram strips</li> </ul> <p><b>Connections</b> 10-pole plug connector (PEI) for connection to the flush-mounted bus coupler</p>	<p><b>Mechanical data</b></p> <ul style="list-style-type: none"> <li>Housing: plastic</li> <li>Dimensions:                             <ul style="list-style-type: none"> <li>DELTA profil (L x W x D): 65 x 65 x 10 mm</li> <li>DELTA style (L x W x D): 68 x 68 x 14 mm</li> </ul> </li> <li>Weight: approx. 55 g</li> <li>Fire load: approx. 950 kJ ± 10 %</li> <li>Mounting: placed on the flush-mounted bus coupler</li> </ul> <p><b>Electrical safety</b></p> <ul style="list-style-type: none"> <li>Degree of pollution (according to IEC 60664-2): 2</li> <li>Type of protection (according to EN 60529): IP 20</li> <li>Protection class (according to IEC 60336): III</li> <li>Overvoltage category (according to IEC 60664-1): III</li> <li>Bus: safety extra-low voltage SELV DC 24 V</li> <li>Device complies with EN 50090-2-2 and IEC 60664-1: 1992</li> </ul> <p><b>Reliability</b> Normal service life: 10 years</p> <p><b>EMC requirements</b> complies with EN 50081-1, EN 61000-6-2 and EN 50090-2-2</p> <p><b>Environmental conditions</b></p> <ul style="list-style-type: none"> <li>Climatic withstand capability: EN 50090-2-2</li> <li>Ambient operating temperature: - 5 ... + 45 °C</li> <li>Storage temperature: - 25 ... + 70 °C</li> <li>Relative humidity (not condensing): 5 % to 93 %</li> </ul> <p><b>Approval</b> EIB-certified</p> <p><b>CE mark</b> in accordance with the EMC guideline (residential and functional buildings) and the low voltage guideline</p>
		<p>1 / 2 <span style="float: right;">Please turn over !</span></p>

**Location and function of the operating and display elements**

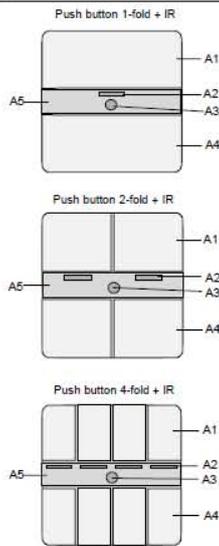


Diagram 2: Location of display and operating elements

- A1 Upper operating field
- A2 LEDs for display
- A3 IR receiver lens
- A4 Lower operating field
- A5 Cover for pictograms

**Notes for installation**

- The device can be used for permanent interior installations, in dry rooms and for insertion in flush-type boxes.
- The range decreases when the receiver lens is exposed to incandescent light.  
Reference point: max. 500 lux permitted.  
In practice: incandescent lamps should not be located less than 2 m away.
- Direct sunlight should be avoided. This can lead to the infrared receiver being completely immobilised.
- Range with wall-mounted transmitter: a value cannot be given as this is dependent on the actual reflections of infrared light at the installation site.
- Range with IR hand-held transmitter S 425 approx. 25 m. See data of the IR receiver.
- The use of pictogram strips reduces the range, depending on the number of strips.

**WARNING**

- The device may only be installed and commissioned by an authorised electrician.
- The device may not be inserted in the same box as 230 V devices.
- The device may be used in switch sockets, if VDE approved devices have been used.
- The prevailing safety and accident regulations must be observed.
- The device may not be opened.
- Any faulty devices should be returned to the local Siemens office.

**Mounting**

**General description**

The push button + IR is placed together with the relevant frame on the flush-mounted bus coupler.  
The flush-mounted bus coupler (C1) is connected to the flush-type box and fixed in position (refer to the installation instructions for the bus coupler).

**Mounting sequence**

- Place the basic push button module (C4, diagram 3) with the relevant frame (C2) on the flush-mounted bus coupler (C1) and press the one strongly upon the other.

**Using fixing screws:**

- If the fixing screws (C3) are used first of all the rockers (C6) have to be lifted out altogether. This can be achieved by inserting the screwdriver between the frames in the recess (C7) and applying leverage (diagram 4).

Important: The push button module (C4, diagram 3) and the rockers (C6) together with the frame (C2) must be held down / pressed down.

- Screw down the fixing screws (C3).
- The rockers (C6) have to be orientized properly and clicked upon the mounting rack (C6).

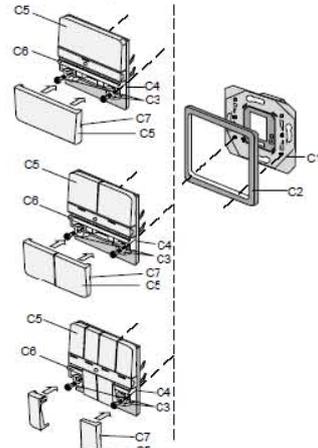


Diagram 3: mounting sequence

**Drawing for mounting the fixing screws**

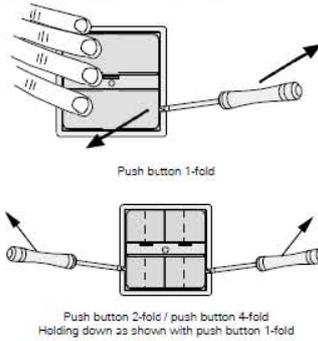


Diagram 4: Lifting out the rockers

**Inserting pictograms**

- Lift out the pictogram cover (A5) with the screwdriver by applying leverage, during this hold down the push button module (C4).

**DELTA profil**

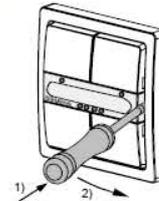
- Insert the screwdriver into the slot (A2) until it reaches the stop and then wiggle it.



Diagram 5: Lifting out the pictogram cover

- Lay the pictograms onto the foundation.
- Snap on the pictogram cover (A5) again.

**DELTA style**



- 1) Insert the screwdriver between the frame and the push button module
- 2) Swivel the screwdriver towards the wall.

Diagram 6: Lifting out the pictogram cover

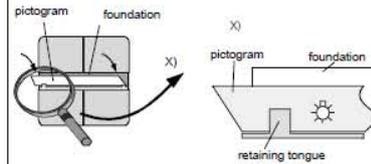


Diagram 7: inserting the pictograms

- Snap on the pictogram cover (A5) again.

**Dismantling**

**Without fixing screws**

- Remove the push button (C4, diagram 3) completely together with the frame from the flush-mounted bus coupler (C1):  
a) with the mere hand  
b) with a screwdriver beneath the frame / wall

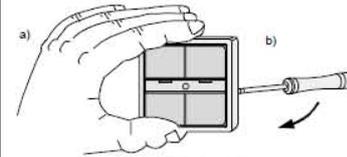


Diagram 8: Dismantling

**With fixing screws**

- Remove the rockers as described at „Mounting / Using fixing screws“.
- Screw out the fixing screws (C3)
- Snap on the rockers (C6) again
- Remove the whole push button module (C4) as described at „Without fixing screws“.

**Note for functions / mounting of the push button + IR receiver**

- The more directly the IR-radiation is received the more certain is the receiving.
- The receipt sensitivity is reduced considerably by objects (persons inclusively) in front of the flush-mounted push button + IR.

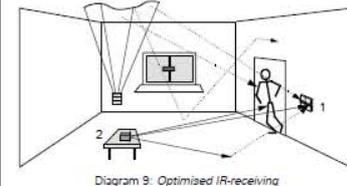


Diagram 9: Optimised IR-receiving

- 1 Flush-mounted push button + IR
- 2 Hand-held transmitter S 425
- 3 Wall-mounted transmitter UP 420/421/422

— direct radiation  
- - - reflected radiation

# Salida binaria N 562

**SIEMENS**   
**Binary Output N 562** **5WG1 562-1AB01**  
**2 x 230 V AC / 10 A**

Issued: March 2003

### Product and Applications Description

The binary output N 562 is a N-system DIN-rail mounted device. It has two outputs to switch on/off two separate groups of electric loads.

Each of the outputs can be assigned various tasks depending on the application program used, i.e. the binary output N 562 consists of the device (hard-ware) and its application programs (software).

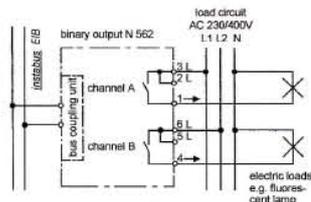
Appropriate application programs are available for the different tasks the binary output N 562 can handle, e.g. for direct on/off switching, time switch (non-delayed on, delayed off), delayed on/off switching or for controlling electrothermal actuators.

With the ETS (*ETS Tool Software*) the application program is selected, its parameters and addresses are assigned appropriately, and downloaded to the binary output N 562.

### Additional Informations

<http://www.siemens.de/garima>

### Example of Operation



### Technical Specifications

**Power supply**  
via bus line

#### Outputs

- number: 2 outputs (volt free contacts)
- rated voltage: AC 230 V, 47... 63 Hz
- rated current: 10 A resistive load
- switching current at AC 230 V:
  - 0,01 ... 10 A resistive load
  - switching current at DC 24 V:
    - 10 A resistive load
    - 4 A inductive load ( $L/R = 7$  ms)
- switching characteristic: set in parameter list according to application program

#### Switching power at AC 230 V

- at incandescent lamp load: max. 1000 W
- at fluorescent lamp (FL) load:
  - uncorrected FL,  $\cos \varphi = 0,5$ : max. 500 W
  - parallel corrected FL,  $\cos \varphi = 1$  (at  $C_{tot} <= 14 \mu F$ ):
    - 2 x 58 W or 3 x 36 W or 6 x 18 W
    - twin-lamp circuit,  $\cos \varphi = 1$ : max. 1000 W
  - Osram ECG for 58 W FL: max. 10 units
  - Osram ECG for 36 W FL: max. 15 units
  - Osram ECG for 18 W FL: max. 20 units

#### Connections

- load circuit, physical:
  - strip insulation for 9 ... 10 mm permissible conductor types/cross sections:
    - 0,5 ... 2,5 mm<sup>2</sup> single core or flexible conductor, 8 mm ultrasonically compacted
    - 0,5 ... 2,5 mm<sup>2</sup> flexible conductor with terminal pin, crimped on gas tight
    - 0,5 ... 1,5 mm<sup>2</sup> flexible conductor with connector sleeve
    - 1,0 and 1,5 mm<sup>2</sup> plain flexible conductor
- load circuit, electrical:
  - plain flexible conductor, min. 1 mm<sup>2</sup> current carrying capacity max. 6 A
  - all other conductors, min. 1,5 mm<sup>2</sup>: current carrying capacity max. 10 A
  - the load circuits have to be saved by a circuit breaker with A or B characteristic with a maximum nominal current of 10 A!

### WARNING

When looping through the L-conductor (connection blocks 3 and 2, 6 and 5), take care that the maximum connection current of 10 A (as governed by the maximum permissible printed conductor load) is not exceeded!

- bus line:
  - pressure contacts on data rail

#### Physical specifications

- N-system DIN-rail mounted device, width: 2 SUs (1SU = 18mm)
- weight: approx. 160 g

#### Electrical safety

- protection (according to EN 60629): IP 20

#### Environmental specifications

- ambient temperature operating: - 5 ... + 45 °C
- ambient temperature non-op.: - 25 ... + 70 °C
- relative humidity (non-condensing): 5 % to 93 %

### Location and Function of the Display and Operator Elements

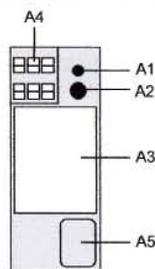


Figure 1: Location of the display and operator elements

- A1 LED for indicating normal operating mode (LED off) and addressing mode (LED on); upon receiving the physical address the device automatically returns to normal operating mode
- A2 Learning button for switching between normal operating mode and addressing mode and for receiving the physical address
- A3 Type plate
- A4 Screwless plug-in terminals for connecting load circuits
- A5 Label for noting the physical address

### Mounting and Wiring

- The device may be used for permanent interior installations in dry locations within distribution boards or small casings with DIN rail EN 60715-TH35-7.5.

### WARNING

- The device may be built into distribution boards (230/400V) together only with appropriate VDE-devices.
- The device must be mounted and commissioned by an authorised electrician
- A safety disconnection of the device must be possible. Especially if the device is connected to different phases.
- Free DIN rail areas with slot-in data rails must be covered with covers, order no. 5WG1 192-8AA01.
- The prevailing safety rules must be heeded.
- The device must not be opened.
- For planning and construction of electric installations, the relevant guidelines, regulations and standards of the respective country are to be considered.

### General Notes

- Any faulty devices should be returned to the local Siemens office.
- If you have further questions about the product, please contact our Technical Support:

☎ +49 (0) 180 50 50-222  
 ☎ +49 (0) 180 50 50-223  
 ✉ [info.usg@siemens.com](mailto:info.usg@siemens.com)

251501.41.51"e"



# Entrada binaria N 260

**SIEMENS**



**Binary Input N 260 PL 4 x 230 V AC**      **5WG1 260-1PB01**

Issued: March 1999

**Product and Applications Description**

The binary input N 260 pl (powerline) is an *instabus pl EB* N-system DIN-rail mounted device. The data communication takes place via the 230 V installation fitting.

4 inputs for 230 V AC switching or keying signals from different outer conductors (L1, L2, L3) with the same basic potential are available.

Each of the inputs can be assigned various tasks depending on the application program used, i.e. the binary input N 260 pl consists of the device (hardware) and its application programs (software).

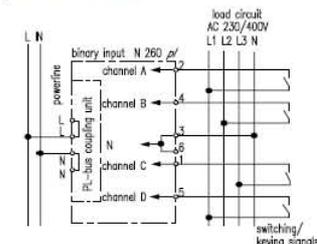
What application program comes into use depends on the individual task: the binary input N 260 pl can e.g. transmit switching on/off telegrams based on varying input switching edges either event controlled or cyclic with parametrisable repetition intervals.

With the ETS (*EB* Tool Software) the application program is selected, its parameters and addresses are assigned appropriately, and downloaded to the binary input N 260 pl.

**Application Programs**

See Siemens product database from version E onward

**Example of Operation**



**Technical Specifications**

**Power supply**

via mains fitting

- mains voltage: 230 V AC ±10%
- frequency: 50 Hz
- current consumption: typ. 65 mA AC (capacitive)
- no-load power consumption: approx. 2 W
- max. level of transmission: 116 dbμV (according to EN 50065-1)

**Inputs**

- number: 4 inputs
- input signal voltage
  - rated value: AC 230 V
  - frequency: 47 ... 63 Hz
  - for signal "0": 0 ... 170 V
  - for signal "1": 198 ... 264 V
- input signal current (input signal): on "1": typ. 1 mA (at AC 230 V)
- input signal delay:
  - at raising input signal edge: max. 5 ms
  - at falling input signal edge: max. 30 ms
- input signal length: min. 50 ms
- input characteristic: set in parameter list according to application program
- signal line length: max. 100 m unshielded
- glow lamps for push button illumination: following max. number of glow lamps per input can be connected via a chip bridge STCS 015:
  - increased lighting strength (STG 342): max. 12 units
  - middle lighting strength (STG 332): max. 18 units
  - low lighting strength (STG 321): max. 60 units

**Control elements**

1 learning button: for switching between normal operating mode and addressing mode

**Display elements**

1 red LED: for monitoring bus voltage and displaying mode, selected with the learning button

**Connections**

- signal inputs, screwless plug-in terminals
- load circuit, physical: strip insulation for 9 ... 10 mm permissible conductor types/cross sections:
  - 0,5 ... 2,5 mm<sup>2</sup> single core or flexible conductor, 8 mm ultrasonically compacted
  - 0,5 ... 2,5 mm<sup>2</sup> flexible conductor with terminal pin, crimped on gas tight
  - 0,5 ... 1,5 mm<sup>2</sup> flexible conductor with connector sleeve
  - 1,0 and 1,5 mm<sup>2</sup> plain flexible conductor

**WARNING**

When looping through the L-conductor (connection blocks 3 and 6), take care that the maximum connection current of 2 A (as governed by the maximum permissible printed conductor load) is not exceeded!

- Powerline: mains terminal (screw blocks): double formation for downloading Permitted conductor cross sections:
  - 0,5...4 mm<sup>2</sup> single core
  - 0,5...2,5 mm<sup>2</sup> flexible conductor with terminal pin

**WARNING**

When looping through the L- and N-conductor take care that the maximum connection current of 16 A (as governed by the maximum permissible printed conductor load) is not exceeded!

**Physical specifications**

- housing: plastic
- N-system DIN-rail mounted device, width: 2,5 SUs (1 SU=18mm)
- weight: approx. 180 g
- fire load: approx. 2610 kJ ± 10 %
- installation: rapid mounting on DIN EN 50022-35 x 7,5 rail

**Electrical safety**

- fouling class (according to IEC 664-1): 2
- protection (according to IEC 529): IP 20
- overvoltage class (according to IEC 664-1): III
- the device complies with EN 50090-2-2 and EN 60669-2-1

**Reliability**

rate of failure: 2047 fit at 40 °C

**Electromagnetic compatibility**

complies with EN 50081-1, EN 50082 and EN 50065

**Environmental specifications**

- climatic conditions: EN 50090-2-2
- ambient temperature operating: - 5 ... + 45 °C
- ambient temperature non-op.: - 25 ... + 70 °C
- relative humidity (non-condensing): 5 % to 93 % (dewing not allowed)

**Certification**

EIB certificate

**CE norm**

complies with the EMC regulations (residential and functional buildings), and low voltage regulations

**Location and Function of the Display and Operator Elements**

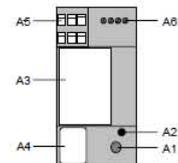


Figure 1: Location of the display and operator elements

- A1** LED for indicating normal operating mode (LED off) and addressing mode (LED on); upon receiving the physical address the device automatically returns to normal operating mode
- A2** Learning button for switching between normal operating mode and addressing mode for receiving the physical address.
- A3** Type plate
- A4** Label for noting the physical address
- A5** Screwless plug-in terminals for connecting load circuits
- A6** Screw blocks to connect the PL-mains fitting

**Installation Instructions**

- The device may be used for permanent interior installations in dry locations within distribution boards.

**WARNING**

- The device may be built into distribution boards (230/400V) together only with appropriate VDE-devices and must be mounted and commissioned by an authorised electrician.
- A safety disconnection of the device must be possible. Especially if the device is connected to different phases.
- The prevailing safety rules must be heeded.
- The device must not be opened. A device suspected faulty should be returned to the local Siemens office.

**Mounting and Wiring**

**General description**

The N-system DIN-rail device can be installed to N-system distribution boards, surface or flush mounted, or to any DIN-rail EN 50022-35 x 7,5 available.

**Mounting DIN-rail devices (Figure 2)**

- Slide the device (B1) onto the DIN-rail (B2) and
- swivel back the device until the slide clicks into place audibly.

**Dismounting DIN-rail devices (Figure 2)**

- Remove all connected wires,
- press down the slide (C3) with a screw-driver and
- swivel the device (C1) from the DIN-rail (C2).

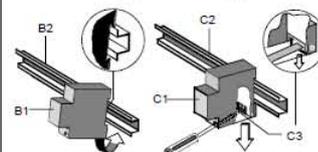


Figure 2: Mounting and dismantling a DIN-rail device

**Connecting load circuits (Figure 3)**

- The load circuits are connected via screwless plug-in terminals (D1).
- Remove approx. 9 to 10 mm of insulation from the wire (D1.1) and plug it into the terminal (D1).

Conductor cross sections: see technical specifications

**Disconnecting load circuits (Figure 3)**

- Press the terminal lock (E1.2) with a screw-driver and
- remove the wire (E1.1) from the terminal (E1).

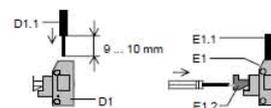


Figure 3: Connecting and disconnecting wires

**Connecting the PL-mains fitting (Figure 4)**

- The terminals are screw blocks in double formation for downloading
- Remove approx. 9 to 10 mm of insulation from the wire (F1.1), plug it into the terminal (F1) and tighten the screws (F1.2).

Conductor cross sections: see technical specifications

**Disconnecting the PL-mains fitting (Figure 4)**

- Loosen the screw (G1.2) and slip the conductor (G1.1) out of the block

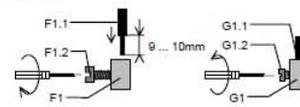


Figure 4: Connecting and disconnecting powerline wires

251561.41.11..b\*

