



UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
INGENIERÍA INFORMÁTICA

PROYECTO FIN DE CARRERA

PROGRAMACIÓN EN JAVA
DE UN ENTORNO DE CONTROL
PARA UN SISTEMA DOMÓTICO

AUTOR

CARLOS HORRO MARCOS

DIRECTOR

MARIO MUÑOZ ORGANERO



AGRADECIMIENTOS

A mi familia, y muy especialmente a mi abuela Florina, por todo.

A Haydeé, por su apoyo vital.

A Raquel y Rodrigo, por cuanto me han enriquecido como persona.

Gracias a todos.



ÍNDICE DE CONTENIDOS

CAPITULO 1. INTRODUCCIÓN

1.PROPÓSITO DEL PROYECTO.....	12
2.PLAN DE TRABAJO.....	13
3.ORGANIZACIÓN DEL PROYECTO.....	14

CAPITULO 2. ESTADO DE LA CUESTIÓN

1.LA DOMÓTICA ACTUAL.....	16
1.INTRODUCCIÓN.....	16
2.HOGAR TRADICIONAL VS. HOGAR DOMÓTICO.....	18
2.TECNOLOGÍAS EMPLEADAS.....	21
1.LA HISTORIA DE JAVA.....	21
2.CARACTERÍSTICAS DE JAVA.....	23
3.HERRAMIENTAS Y LIBRERÍAS JAVA UTILIZADAS.....	25
3.OPENOFFICE.....	27

CAPITULO 3. ESTUDIO DEL SISTEMA DE ERGODOMUS

1.LA EMPRESA.....	28
2.DESCRIPCIÓN DEL SISTEMA.....	29
1.UNA VISIÓN GENERAL.....	29
2. EL BUS ED.....	30
3. CONSIDERACIONES FINALES.....	32
3.ESTUDIO DE LA SOLICITUD.....	33
4.CONSIDERACIONES AMBIENTALES.....	34
5.REQUISITOS DE USUARIO.....	35
1.REQUISITOS DE CAPACIDAD.....	37
2.REQUISITOS DE RESTRICCIÓN.....	41

CAPITULO 4. ANÁLISIS DEL SISTEMA

1.ESPECIFICACIÓN DE CASOS DE USO.....	43
1.DIAGRAMA DE CASOS DE USO.....	43
2.DESCRIPCIÓN DE LOS CASOS DE USO.....	44
2.MODELO CONCEPTUAL.....	51
1.DIAGRAMA DE CLASES CONCEPTUAL.....	51
2.DESCRIPCIÓN DEL DIAGRAMA.....	52
3.OBTENCIÓN DE REQUISITOS SOFTWARE.....	54



4.REQUISITOS FUNCIONALES.....	57
5.REQUISITOS DE RENDIMIENTO.....	67
6.REQUISITOS DE INTERFAZ.....	71
7.REQUISITOS OPERACIONALES.....	72
8.REQUISITOS DE RECURSOS.....	75
9.REQUISITOS DE VERIFICACIÓN.....	75
10.REQUISITOS DE PRUEBAS DE ACEPTACIÓN.....	76
11.REQUISITOS DE DOCUMENTACIÓN.....	78
12.REQUISITOS DE SEGURIDAD ANTE AMENAZAS.....	79
13.REQUISITOS DE PORTABILIDAD.....	79
14.REQUISITOS DE CALIDAD.....	80
15.REQUISITOS DE FIABILIDAD.....	80
16.REQUISITOS DE MANTENIMIENTO.....	80
17.REQUISITOS DE SEGURIDAD ANTE DAÑOS FÍSICOS.....	81
6.MATRIZ DE TRAZABILIDAD.....	82
7.DIAGRAMAS DE SECUENCIA.....	86
1.DIAGRAMA DE SECUENCIA DE PETICIÓN DE ESTADO POR EL USUARIO.....	87
2.DIAGRAMA DE SECUENCIA DE LA ACTUALIZACIÓN DEL ESTADO DE LOS DISPOSITIVOS DEL SISTEMA.....	88

CAPITULO 5. DESCRIPCIÓN DEL SISTEMA

1.INSTALACIÓN.....	89
2.CONFIGURACIÓN DE LA APLICACIÓN.....	90
3.EJECUCIÓN DE LA APLICACIÓN.....	95
1. LA VENTANA PRINCIPAL DE LA APLICACIÓN.....	95
2. LA TERMINAL DE COMANDOS.....	128

CAPITULO 6. CONCLUSIONES

CAPITULO 7. LÍNEAS FUTURAS DE DESARROLLO

1.NUEVAS INTERFACES DE COMUNICACIÓN.....	132
2.NUEVOS TIPOS DE DISPOSITIVOS.....	132
3.NUEVAS INTERFACES DE USUARIO.....	132
4.VISTA EN 3D EN LA INTERFAZ GRÁFICA.....	133
5.CONTROL VÍA INTERNET.....	133

ANEXO I. GLOSARIO

1.DEFINICIONES.....	134
2.ACRÓNIMOS.....	136

ANEXO II. BIBLIOGRAFÍA

1.BIBLIOGRAFÍA TRADICIONAL.....	138
---------------------------------	-----



2.BIBLIOGRAFÍA EN FORMATO ELECTRÓNICO.....	139
1.REFERENCIAS SOBRE DOMÓTICA.....	139
2.REFERENCIAS SOBRE OTROS TEMAS INFORMÁTICOS.....	140

ÍNDICE DE ILUSTRACIONES

Ilustración 1: El Hogar Digital actual.....	17
Ilustración 2: Evolución en los últimos años del uso de Java [I-2].....	22
Ilustración 3: Pasos de la ejecución de un programa Java.....	24
Ilustración 4: Estructura de la Plataforma Java.....	25
Ilustración 5: Logotipo de SwiXML.....	25
Ilustración 6: Código fuente del presente proyecto en Netbeans 6.1.....	26
Ilustración 7: Openoffice writer editando la presente memoria.....	27
Ilustración 8: Logotipo de Ergodomus.....	28
Ilustración 9: Pila de protocolos OSI [I-3].....	30
Ilustración 10: Diagrama de Casos de Uso.....	43
Ilustración 11: Diagrama de clases conceptual.....	51
Ilustración 12: Diagrama de Secuencia: petición de estado por el usuario.....	87
Ilustración 13: Diagrama de Secuencia: actualización del estado de los dispositivos del sistema.....	88
Ilustración 14: Estructura de directorios de Netbeans.....	89
Ilustración 15: Estructura de directorios necesaria	90
Ilustración 16: Estructura del directorio de recursos	91
Ilustración 17: Ventana principal al iniciarse por primera vez.....	95
Ilustración 18: Las diferentes secciones de la ventana principal	96
Ilustración 19: Menú de herramientas con el submenú 'Archivo' desplegado.....	97
Ilustración 20: Menú de herramientas con el submenú 'Edición' desplegado.....	97
Ilustración 21: Ventana “Acerca de” el Sistema domótico.....	98
Ilustración 22: Menú de herramientas con el submenú 'Ayuda' desplegado.....	98
Ilustración 23: Barra de botones de la ventana principal.....	98
Ilustración 24: Consola de usuario	99
Ilustración 25: Ventana de opciones	101
Ilustración 26: Ventana de adición de aparatos	102
Ilustración 27: Ventana de adición de aparatos con los elementos disponibles de un módulo.....	103
Ilustración 28: Ventana de adición de aparatos con las direcciones disponibles para conectarse.....	104
Ilustración 29: Ventana de adición de aparatos con los tipos de aparatos permitidos...104	
Ilustración 30: Ventana principal con un módulo y un aparato (bombilla) añadido.....	105

Ilustración 31: Ventana de adición de módulos.....	106
Ilustración 32: Diferentes partes del mapa de dispositivos.....	107
Ilustración 33: Mapa de Dispositivos con varios módulos y aparatos cargados.....	108
Ilustración 34: Mapa de Dispositivos mostrando un 'tooltip' con información de un aparato.....	109
Ilustración 35: Menú contextual de un dispositivo.....	110
Ilustración 36: Cambio del tipo de un aparato.....	111
Ilustración 37: Mapa de Dispositivos con un filtrado por tipo de aparato.....	112
Ilustración 38: Mapa de Dispositivos con un filtrado por módulo de conexión.....	113
Ilustración 39: Mapa de Dispositivos con un filtrado por habitación.....	113
Ilustración 40: Mapa de Dispositivos con dos filtrados a la vez: por módulo y por habitación.....	114
Ilustración 41: Ventana de modificación de módulos de E/S.....	115
Ilustración 42: Reprogramación del identificador de un módulo.....	116
Ilustración 43: Opciones de actualización y reprogramación masiva de un módulo....	117
Ilustración 44: Configuración detallada de un módulo.....	117
Ilustración 45: Configuración de las entradas analógicas de un módulo.....	118
Ilustración 46: Configuración de las entradas digitales de un módulo.....	119
Ilustración 47: Configuración de la salida analógica de un módulo.....	120
Ilustración 48: Configuración de la salidas digitales de un módulo.....	121
Ilustración 49: Configuración de los grupos de un módulo.....	122
Ilustración 50: Configuración de las persianas de un módulo.....	123
Ilustración 51: Panel con la información y opciones de cambio de estado de una bombilla conectada a una salida analógica.....	124
Ilustración 52: Panel con la información y opciones de cambio de estado de una bombilla conectada a una salida digital.....	125
Ilustración 53: Panel con la información y opciones de cambio de estado de un aparato de tipo persiana.....	125
Ilustración 54: Panel con la información y opciones de cambio de estado de un grupo de acciones.....	126
Ilustración 55: Panel con la información de un aparato de entrada de datos.....	126
Ilustración 56: Ejemplo del estado del panel de eventos tras una actividad mediana..	127
Ilustración 57: Terminal mostrando el proceso de carga de la aplicación.....	128
Ilustración 58: Terminal mostrando las entradas y salidas de la aplicación.....	129
Ilustración 59: Terminal mostrando el error java ante la falta de una librería.....	130
Ilustración 60: Terminal mostrando errores en la actualización de los dispositivos....	130



ÍNDICE DE TABLAS

Tabla I. Caso de uso de ejemplo: descripción.....	44
Tabla II. Caso de uso CU-01: Añadir Módulo E/S.....	44
Tabla III. Caso de uso CU-02: Modificar Módulo E/S.....	45
Tabla IV. Caso de uso CU-03: Eliminar Módulo E/S.....	45
Tabla V. Caso de uso CU-04: Ver estado módulo E/S.....	46
Tabla VI. Caso de uso CU-05: Modificar programación módulo E/S.....	46
Tabla VII. Caso de uso CU-06: Almacenar programación módulo E/S.....	47
Tabla VIII. Caso de uso CU-07: Añadir aparato.....	47
Tabla IX. Caso de uso CU-08: Modificar aparato.....	48
Tabla X. Caso de uso CU-09: Eliminar aparato.....	48
Tabla XI. Caso de uso CU-10: Ver estado aparato.....	48
Tabla XII. Caso de uso CU-11: Modificar estado aparato.....	49
Tabla XIII. Caso de uso CU-12: Modificar opciones.....	49
Tabla XIV. Caso de uso CU-13: Enviar instrucción al bus.....	50
Tabla XV. Tipos de aparatos soportados por la aplicación.....	70
Tabla XVI. Matriz de Trazabilidad.....	85
Tabla XVII. Archivo de configuración configuracion.properties.....	92
Tabla XVIII. Archivo de configuración puertoSerie.properties.....	92
Tabla XIX. Archivo de configuración interfaz.properties.....	93



ÍNDICE DE REQUISITOS

Identificador: Requisito de Ejemplo.....	36
RU-C01: Administración de Módulos.....	37
RU-C02: Administración de Aparatos.....	37
RU-C03: Componentes físicos de los módulos.....	37
RU-C04: Propiedades de una entrada digital.....	37
RU-C05: Propiedades de una salida analógica.....	38
RU-C06: Propiedades de una salida digital.....	38
RU-C07: Componentes lógicos de los módulos.....	38
RU-C08: Propiedades de un grupo de acciones.....	38
RU-C09: Propiedades de una persiana lógica.....	39
RU-C10: Situación de los módulos.....	39
RU-C11: Propiedades de los aparatos.....	39
RU-C12: Interacción con el sistema domótico.....	39
RU-C13: Reprogramación de los Módulos.....	40
RU-C14: Backup de la programación de los Módulos.....	40
RU-C15: Registro de actividad.....	40
RU-R01: Comunicación por puerto serie.....	41
RU-R02: Funcionamiento de la aplicación.....	41
RU-R03: Tipo de dispositivos conectados a un módulo.....	41
RU-R04: Opciones de la aplicación.....	41
RU-R05: Mantenimiento de la aplicación.....	42
Identificador: Requisito software de ejemplo.....	56
RS-F01: Representación interna del sistema.....	57
RS-F02: Administración de módulos.....	57
RS-F03: Obtención del estado de los módulos.....	57
RS-F04: Administración de aparatos.....	58
RS-F05: Obtención del estado de los aparatos.....	58
RS-F06: Modificación del estado de los aparatos.....	58
RS-F07: Eliminación de aparatos en cascada.....	59
RS-F08: Características físicas de los módulos de E/S.....	59
RS-F09: Características de una entrada analógica.....	59
RS-F10: Características de una entrada digital.....	60
RS-F11: Características de una salida analógica.....	60
RS-F12: Características de una salida digital.....	60



RS-F13: Condiciones de una salida digital.....	61
RS-F14: Características lógicas de los módulos de E/S.....	61
RS-F15: Características de un grupo de acciones programadas.....	62
RS-F16: Características de una persiana lógica.....	62
RS-F17: Características de situación de los módulos.....	63
RS-F18: Características de los aparatos.....	63
RS-F19: Múltiples interfaces de usuario.....	64
RS-F20: Múltiples buses de comunicación.....	64
RS-F21: Interfaz gráfica completa.....	64
RS-F22: Visualización selectiva.....	65
RS-F23: Reprogramación de los módulos de E/S.....	65
RS-F24: Actualización de la representación interna de los módulos de E/S.....	65
RS-F25: Historial de eventos.....	66
RS-R01: Tiempo de respuesta de estado.....	67
RS-R02: Tiempo de respuesta a cambio de estado.....	67
RS-R03: Módulos y aparatos ilimitados.....	67
RS-R04: Valores iniciales en los parámetros configurables.....	68
RS-R05: Tipos de aparatos válidos	69
RS-I01: Elementos para la conexión.....	71
RS-I02: Comunicación de la aplicación con el sistema domótico.....	71
RS-I03: Almacenamiento de la información.....	71
RS-O01: Interfaz gráfica.....	72
RS-O02: Visualización de la información de un módulo.....	72
RS-O03: Visualización rápida de la información de un módulo.....	73
RS-O04: Visualización de la información de un aparato.....	73
RS-O05: Visualización rápida de la información de un aparato.....	73
RS-O06: Modificación del estado de un aparato.....	74
RS-O07: Sistema de colores en la interfaz gráfica.....	74
RS-O08: Menús contextuales en la interfaz gráfica.....	74
RS-O09: Movimiento de dispositivos en la interfaz gráfica.....	75
RS-U01: Requisitos del PC.....	75
RS-V01: Comprobación del paradigma.....	75
RS-A01: Comprobación de estados.....	76
RS-A02: Comprobación de modificación de estados.....	76
RS-A03: Comprobación de opciones.....	76
RS-A04: Comprobación del funcionamiento de la consola.....	77



RS-A05: Comprobación de la reprogramación de un módulo.....77

RS-A06: Comprobación del registro de actividad.....77

RS-A07: Comprobación de información válida.....78

RS-D01: Manual de usuario.....78

RS-D02: Documentación Javadoc.....78

RS-SA01: Copias de seguridad.....79

RS-SA02: Protección de las copias de seguridad.....79

RS-P01: Entorno Java 6 y RXTX.....79

RS-FB01: Fiabilidad del sistema.....80

RS-M01: Corrección de errores y nuevas funcionalidades.....80

RS-SF01: Control de dispositivos inoperativos.....81

RS-SF02: Desconexión y apagado de aparatos ante riesgos.....81



INTRODUCCIÓN

1. PROPÓSITO DEL PROYECTO

El presente proyecto de fin de carrera surge por el interés de la empresa *Ergodomus S.L* en realizar una aplicación informática que permita controlar y visualizar el estado de todos los componentes que formen parte de un determinado sistema domótico, que podría ser distinto en función del lugar donde se instale.

Para ello se ha tenido que trabajar con los módulos de control desarrollados por la empresa, su protocolo de comunicaciones, las características de las comunicaciones vía puerto serie (a través de la cual la aplicación se comunicará con los módulos) y la programación de los propios módulos de control.

De todo ello me ha proporcionado la información necesaria para el proyecto Enrique Verdú Calatayud, líder de *Ergodomus S.L* y desarrollador de todas las tecnologías propias implicadas en el sistema.



2. PLAN DE TRABAJO

Para llevar a buen término el proyecto, se han delimitado los siguientes bloques de trabajo:

1. Estudio sobre domótica general:

Se estudiarán las posibilidades de estos sistemas para familiarizarse con ellos y poder sacarles el máximo partido.

2. Estudio del sistema domótico de Ergodomus:

Es necesario conseguir una alta comprensión de la infraestructura con que cuenta el sistema físico antes de comenzar con el desarrollo de la aplicación.

3. Análisis de requisitos:

Una vez familiarizado con el sistema domótico de *Ergodomus*, se procederá a la recogida y análisis de requisitos de usuario que deberá contemplar la aplicación para satisfacer las necesidades de la empresa.

4. Elaboración de la memoria:

En el presente proyecto la elaboración de la memoria irá codo con codo con la construcción del sistema en sí, debido a la importancia de un correcto análisis de la aplicación para conseguir un sistema robusto y con una amplia modularidad y facilidad de expansión.

5. Programación del sistema:

Será realizada en Java sobre la plataforma J2SE 6 y el entorno de desarrollo Netbeans. Es el objetivo final del proyecto.

3. ORGANIZACIÓN DEL PROYECTO

La presente memoria consta de una serie de capítulos, dedicados a las diferentes fases y partes del proyecto:

- **Capítulo 1. Introducción** ofrece un resumen del marco de trabajo en que se ha realizado el Proyecto, comentando las motivaciones que han llevado a su realización, la línea de desarrollo marcada y la estructura de la presente memoria.
- **Capítulo 2. Estado de la Cuestión** ofrece un resumen de la tecnología disponible actualmente relativa a la domótica y de las tecnologías asociadas que han posibilitado la realización del presente Proyecto Fin de Carrera.
- **Capítulo 3: Estudio del sistema de Ergodomus** realiza una descripción de las características de la infraestructura del sistema domótico de Ergodomus y lo que esta empresa espera de la aplicación de control domótico, reflejado principalmente en los requisitos de usuario de la aplicación.
- **Capítulo 4: Análisis del sistema** realiza un análisis detallado de las necesidades de la aplicación a través de la elaboración de los requisitos software a partir de los requisitos de usuario. Por último se ofrece una matriz de trazabilidad para un seguimiento más sencillo de la correspondencia entre dichos requisitos.
- **Capítulo 5: Manual de usuario** ofrece una guía de uso detallada de la aplicación, orientada principalmente a nuevos usuarios.
- **Capítulo 6: Conclusiones** contrasta los resultados conseguidos en el proyecto con los esperados y el grado de satisfacción del cliente con la aplicación entregada.
- **Capítulo 7: Líneas futuras de desarrollo** ofrece una descripción de las ampliaciones y mejoras posibles a realizar en la aplicación a corto-medio plazo, además de nuevas e interesantes posibilidades que podrían incorporarse en un futuro más lejano.



- **Capítulo 8: Glosario** ofrece una serie de definiciones de términos y acrónimos usados a lo largo de la memoria.

- **Capítulo 9: Bibliografía** detalla las fuentes de información usadas en la elaboración del presente proyecto, tanto de tipo impreso (libros, revistas u otros proyectos de fin de carrera) como de tipo electrónico (páginas web).

Capítulo 2

ESTADO DE LA CUESTIÓN

1. LA DOMÓTICA ACTUAL

1. Introducción

El desarrollo de las tecnologías de la información y de las telecomunicaciones durante los últimos 40 años ha sido probablemente superior al conseguido durante toda la historia anterior. La Informática se ha introducido en buena parte de los hogares, al mismo tiempo que aparatos como los teléfonos móviles, las redes WIFI, la televisión digital y por satélite, etc. son algo ya habitual entre nosotros.

Todos estos avances, aún siendo muy loables, son el producto de la concentración de la industria en ciertos campos que demuestran ser rentables económicamente; otros muchos, sin embargo, permanecen prácticamente en el anonimato a la espera de encontrárselos fórmulas comerciales con altas posibilidades mercantiles.

Todo esto responde al hecho de que, como es sabido, el impulso tecnológico no es arbitrario, sino que en buena medida es dirigido en función de su rentabilidad para la industria (ya sea civil, militar, etc).

Lo anterior se pone de relieve claramente en el campo de la domótica. Como claro ejemplo, según el diccionario de la RAE, domótica es:

(Del lat. *domus*, casa, e *informática*). **1. f.** Conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda.

Esta definición de domótica se ciñe en exclusiva a los sistemas que se han venido instalando en la gran mayoría de los casos, consistentes en automatismos para las luces de la casa, sistemas de climatización, etc. Sin embargo la domótica nació sin ponerse esos claros límites, abierta a todo aquello que pudiera hacer de un hogar un lugar acogedor y al mismo tiempo altamente funcional, que facilitase la vida diaria de sus inquilinos en aspectos tan variados como, por ejemplo, la cocina, la vigilancia de la casa, su soporte multimedia, etc.

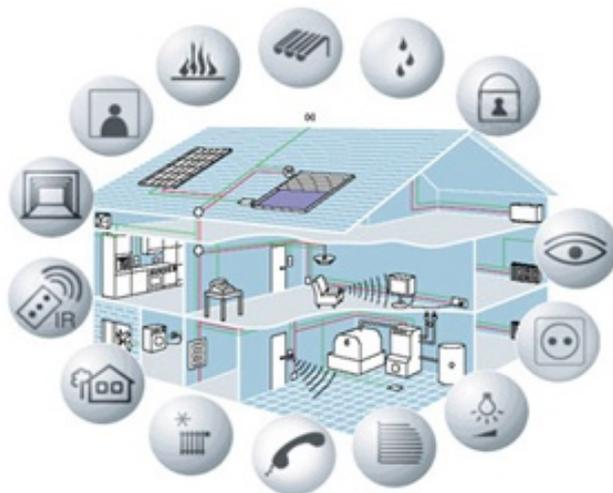


Ilustración 1: El Hogar Digital actual

Es por este motivo por el que recientemente han empezado a surgir términos como *Hogar digital*, que intentan englobar todos los campos que de momento no consiguen abarcar la domótica actual.

También es cierto, no obstante, que no sólo motivos de mercado han restringido la expansión de la domótica. La globalización de las tecnologías digitales (conexiones a Internet de banda ancha, tratamiento digital fluido de imagen y sonido, tecnologías ecológicas que buscan instalaciones sostenibles, etc.) no ha ocurrido hasta fechas más o menos recientes (y no es posible imaginar un sistema domótico moderno sin este tipo de componentes), lo que unido al hecho inevitable de que los precios iniciales fueran muy elevados, hacían prohibitiva para la gran mayoría de la población la adquisición de un sistema domótico, por sencillo que fuera.

Otro problema ha sido, en buena medida, el no haber “sabido vender” bien sus bondades, ya que todavía persiste la idea general de que la domótica es una especie de lujo caro destinado a facilitar actividades muy prescindibles.

Todos estos problemas, no obstante, están siendo salvados poco a poco con una progresiva reducción de costes, un progreso imparable en todos los campos tecnológicos y las aplicaciones cada vez más amplias (y vistosas) que pueden ofrecer los sistemas domóticos modernos.

Teniendo en cuenta lo explicado anteriormente, una buena clasificación de los distintos aspectos que puede englobar un Hogar Digital sería la de Junstrand [JUN-05]:

- Sistemas de Domótica (referidos a automatismos)
- Sistemas de Seguridad
- Sistemas de Multimedia
- Sistemas de Telecomunicación

Sin embargo, es de esperar que los términos Hogar Digital y Domótica converjan según las compañías de domótica vayan desarrollando y ampliando el campo de acción de sus sistemas dentro de la propia competencia entre ellas, el abaratamiento de los costes y la disponibilidad de tecnologías cada vez más potentes.

Esto ya es una realidad en algunas compañías, como la que ha fomentado este Proyecto Fin de Carrera, *Ergodomus*. Por lo tanto, en adelante ambos términos se usarán indistintamente.

2. Hogar tradicional vs. Hogar domótico

Según los aspectos que cubra el sistema instalado en un hogar, podrá hablarse de distintos niveles de *domotización* del mismo. El objetivo final de los sistemas domóticos más ambiciosos, y del de *Ergodomus* en concreto, será ofrecer el mayor nivel de *domotización*, esto es, una solución de domótica integral.

Un hogar tradicional, o con un *nivel 0* de domotización, se podría caracterizar por:

- Servicio de alimentación eléctrica con una cierta **potencia contratada**.
- **Iluminación manual**. En la gran mayoría de los casos consistente en interruptores de apagar/encender.
- **Soluciones de seguridad aisladas entre sí** (como *parches*): esto es, pueden instalarse sistemas de seguridad, de alimentación ininterrumpida, etc. pero no interactuarán con el resto de aparatos de la casa

Un hogar domótico integral, o de máximo nivel de domotización, estaría por el contrario caracterizado por:

- **Simplificación de la vida diaria**, mediante automatismos tan relativamente sencillos como realizar la compra desde casa eligiendo para cada gama de productos las tiendas más baratas, o sistemas de limpieza robotizados que sean capaces de usar la información proporcionada por los sensores del hogar para realizar una limpieza óptima, etc.
- Iluminación (en particular) y **potencia eléctrica** (en general) tanto manual como **automática** (a la medida), variable según múltiples parámetros (como presencia o no de gente, condiciones climatológicas exteriores, grado de ahorro energético deseado por los dueños del hogar, riego automático según lo necesite el terreno...).
- **Sistemas de seguridad globales e interconectados** entre sí y con todos los sistemas del hogar. Esto abarca cualquier tipo de sistema de seguridad imaginable: contra intrusos (avisos a centralitas), incendios (posibilitando incluso sistemas para su sofocamiento automático), escapes de gas (cierre de la entrada de gas general y apagado del sistema eléctrico), emergencias médicas...

En todos estos casos también sería posible avisar al propietario del hogar a través de un mensaje al móvil, por correo electrónico, etc.

- **Control remoto de la vivienda**. Cualquier sistema podría ser revisado de forma remota a través de un panel de control central o incluso a través de internet. Esto sería posible mediante el envío de la señal de vídeo y audio de las cámaras instaladas, del estado de los distintos tipos de sensores presentes (luz, humedad, humo, gas, temperatura...).

De igual manera, sería posible modificar el estado de los sistemas de la casa de forma remota: cerrar las ventanas si se quedaron abiertas al salir, preparar un baño de agua caliente para que esté listo al llegar a casa, etc.

- **Gestión eficaz automática de los propios recursos de la vivienda**. Esto es, la toma en consideración de las condiciones del exterior y del interior de la vivienda a la hora de ejecutar cualquier acción: por ejemplo, en invierno subir las persianas en las horas de mayor calor del día para poder bajar el consumo en calefacción,

Los objetivos de este hogar altamente domotizado pueden lograrse mediante la adición de sistemas no sólo pasivos, como ocurría en los hogares tradicionales, sino también activos e incluso *proactivos* (esto es, que tomen decisiones no sólo cuando ocurre un evento en el sistema, sino que actúen constantemente regulando el conjunto del hogar domótico para conseguir un estado óptimo durante todo el tiempo).



Como es obvio, esto requiere un cierto nivel de *inteligencia* y de capacidad de computación por parte del sistema domótico. La solución puede implementarse de dos maneras opuestas:

- **Mediante un centro de control.** Esta puede ser en un principio la solución más obvia y más sencilla, ya que puede implementarse mediante un simple ordenador personal de potencia media, muy baratos en la actualidad.
- **Mediante un control distribuido.** Esta solución se basaría en contar con módulos controladores distribuidos por la casa y comunicados entre sí.

Como cualquier solución distribuida, puede resultar en principio más complicada de implementar, pero resulta mucho más escalable y sostenible, además de ser invulnerable a fallos en alguno de los sistemas (en la solución anterior, un problema en el centro de control podría afectar a toda la instalación).

Por otra parte, la comunicación entre los buses puede ser tanto **cableada** como **inalámbrica**. En ambos casos existen varios estándares suficientemente probados como para poder ser tenidos en cuenta (por ejemplo, *EIB* y *LonWorks* por los primeros, o *Wifi* y *Bluetooth* por los segundos), por lo que la elección de una solución u otra dependerá de los requerimientos concretos de cada instalación.

2. TECNOLOGÍAS EMPLEADAS

Para la realización de este Proyecto Fin de Carrera se ha utilizado principalmente la tecnología **Java**, y dado el amplio espectro de posibilidades que ésta abarca, dentro de la misma se han empleado distintas utilidades existentes en la actualidad aunque con distinto grado de difusión.

En particular, uno de los criterios de utilización que se han seguido ha sido su carácter libre (no vinculado a empresas concretas) y gratuito.

1. La historia de Java

El lenguaje Java se publicó por primera vez en 1991 con el nombre *Oak* (posteriormente se cambiaría de nombre, al existir ya un lenguaje con ese nombre).

Para entonces ya era el resultado de décadas de esfuerzo por parte de Sun Microsystems para conseguir un lenguaje adecuado para programar las nuevas generaciones de dispositivos inteligentes de consumo, como vídeos, televisores, etc.

Este lenguaje debía ser sencillo, es decir, debía ser fácil de aprender y fácil de programar, y ésto implicaba que tuviera que ser parecido hasta cierto punto a los lenguajes más exitosos ya existentes por entonces, como C o C++.

Entre 1991 y 1992 el equipo de desarrollo se centró en la ampliación del lenguaje, el desarrollo de aplicaciones básicas y la construcción de algunos prototipos hardware que hicieran uso de las mismas. Con la base ya creada, en 1993 y la primera mitad de 1994 el equipo se centró en buscar nichos de mercado en los que Java pudiera competir con éxito; sin embargo, sus principales ofertas fueron rechazadas y el equipo responsable decidió que el sector del mercado en el que se estaban centrando no era el adecuado.

Sin embargo, en 1994 la *WWW* estaba en plena expansión; esto no pasa desapercibido para los responsables del proyecto, que deciden que los mismos requisitos que cumplía el lenguaje originalmente (sobretudo, operatividad entre distintos sistemas, con hardware y software también distinto), son los que se necesitan en la Web de ese momento.

De este modo, el equipo de desarrollo se centró en conseguir un lenguaje que, a través de un navegador de Internet, pudiera hacer funcionar una aplicación independientemente del sistema operativo del usuario. Fruto de este trabajo, a finales de

septiembre de ese mismo año se terminó el desarrollo del prototipo *HotJava*, que convenció a la directiva de Sun.

A partir de ese momento Sun puso todo su empeño en el desarrollo del prototipo, que se alargó durante todo 1995 hasta que en enero de 1996 apareció la versión 1.0 del JDK de Java. El hecho de que el navegador estrella del momento, Netscape, soportara en su versión 2.0 los *applets* de Java y que Sun permitiera a otras compañías desarrollar JVM para diferentes sistemas operativos, herramientas de programación y acceso al código fuente fue la mezcla de factores que permitió al lenguaje expandirse muy rápidamente y obtener apoyos de las grandes firmas internacionales.

En la actualidad, Java es el lenguaje de programación más extendido del mundo y dispone de multitud de entornos de programación de gran calidad, tanto gratuitos como de pago, además de contar con una enorme comunidad de desarrolladores que continuamente extienden su funcionalidad con nuevas librerías y utilidades.

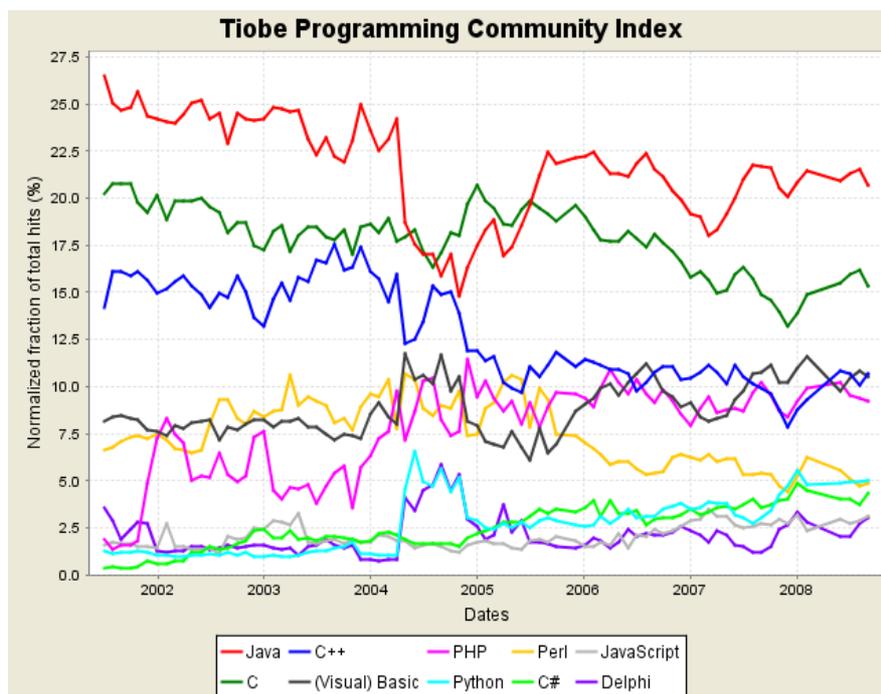


Ilustración 2: Evolución en los últimos años del uso de Java [I-2]

2. Características de Java

Como se ha comentado en el punto anterior, la plataforma Java es una tecnología que en la actualidad cuenta con una larga trayectoria y una amplia implantación.

Las características que le han permitido llegar hasta esta situación han sido las siguientes:

- Es multiplataforma: un programa Java realizado en Windows funcionará igual que si se ejecuta en Linux, con tal de que en ambos sistemas la máquina virtual de java (*JVM*) se encuentre correctamente instalada y sean de las mismas versiones.
- Es multisistema: por los mismos motivos, una aplicación Java también funcionará perfectamente ya se ejecute en un Pc o en un Mac, siempre que esté instalada la correspondiente *JVM*.
- Es un lenguaje de alto nivel orientado a objetos: La orientación a objetos (POO) es un paradigma de programación que utiliza el concepto de objeto para la construcción de aplicaciones. Los conceptos básicos de la orientación a objetos son:
 - **Objeto**: sería una entidad con un conjunto de propiedades y métodos que definen su comportamiento.
 - **Clase**: definición de las propiedades y los métodos de un objeto. Los objetos se crean (*se instancian*) a partir de ellas.
 - **Mensaje**: la comunicación entre un objeto y otro. Ésta se consigue cuando un objeto ejecuta un método del otro.
 - **Propiedad**: representa el estado del objeto, que puede ser alterado a través de sus métodos.
 - **Método**: descripción de una acción que el objeto realizará cuando reciba un mensaje.

Este tipo de paradigma de programación permite características avanzadas que facilitan la labor del programador al mismo tiempo que ofrecen una gran potencia (como recolección de basura, tratamiento de errores, etc.) siguiendo la forma de programar de lenguajes como C y C++, con lo que su aprendizaje y manejo son relativamente sencillos.

- Tiene una amplia librería de clases: los programadores Java tienen a su disposición una enorme variedad de clases y utilidades proporcionadas tanto por *Sun* como por desarrolladores independientes. Java es el lenguaje que mayor cantidad de librerías ofrece a sus desarrolladores en todo el mundo.
- Está extensamente documentada: cualquier clase que se desee utilizar contará, casi con total seguridad, con la correspondiente documentación *Javadoc* explicando su funcionamiento y forma de uso.

Los dos primeros aspectos (clave para la elección de Java en este proyecto, ya que deberá funcionar tanto en Windows como en Linux) son debidos a su naturaleza de lenguaje *interpretado*, esto es, la compilación de su código fuente no genera código máquina directamente, sino un pseudocódigo interpretable por la *JVM*, que será quien lo ejecute como si fuera un “ordenador” que a su vez se ejecuta dentro de nuestro ordenador.

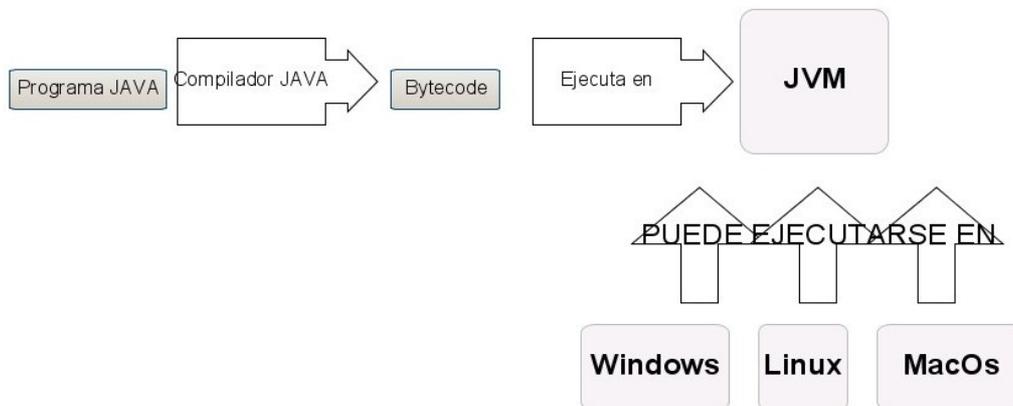


Ilustración 3: Pasos de la ejecución de un programa Java

La complejidad de esta plataforma puede apreciarse mediante el siguiente gráfico de capas (Ilustración 4):

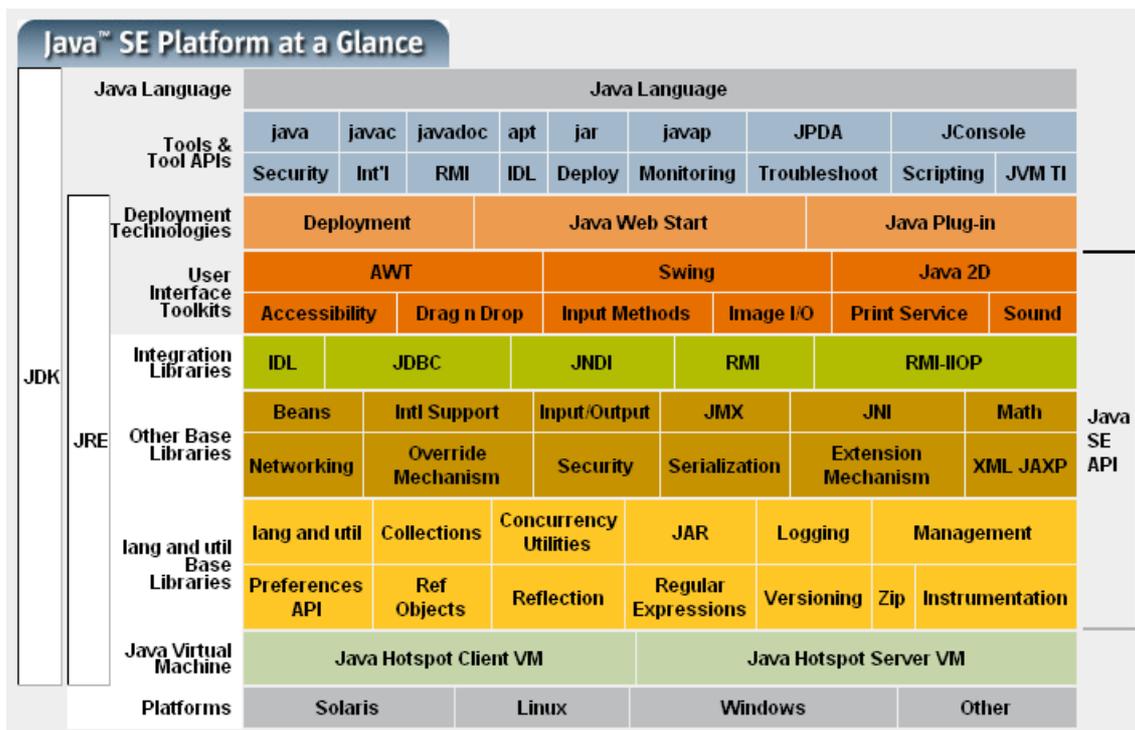


Ilustración 4: Estructura de la Plataforma Java

3. Herramientas y librerías Java utilizadas

1. SwiXML

SwiXML es una librería gracias a la cual es posible plasmar la estructura de una interfaz gráfica *Swing* en un archivo XML aparte, con lo que se “limpia” enormemente el código Java y queda mucho más accesible y clara la estructura interna de la interfaz.



Ilustración 5: Logotipo de SwiXML

El tamaño de la librería es inferior a los 50kb, lo que da una idea de la pequeña carga que supone para el sistema.

2. RXTX

RXTX es una librería de comunicaciones similar a la *JavaComm* proporcionada por Sun, pero a diferencia de ésta está disponible en su última versión para una multitud de sistemas operativos (Windows, Linux, Solaris, FreeBSD) y, en un tiempo, de plataformas (como Mac).

La presente aplicación utiliza esta librería para establecer comunicación con los dispositivos domóticos mediante puerto serie (RS-232), si bien también da la posibilidad (no implementada en el Proyecto) de usar puerto paralelo (IEEE 1284).

3. Netbeans IDE

El *IDE* Netbeans es la plataforma de desarrollo bajo la cual se ha desarrollado la aplicación del presente Proyecto Fin de Carrera. Como en el caso del OpenOffice, se caracteriza por ser *libre* (bajo licencia *GNU/GPL*) y gratuito, aunque en este caso ofrece una enorme funcionalidad que no tiene nada que envidiar a otras aplicaciones comerciales similares.

Netbeans permite escribir, depurar, compilar y ejecutar aplicaciones java e incluso orientadas a la web. Está escrito en Java, pero permite programar en múltiples lenguajes y aumentar la funcionalidad de la plataforma mediante extensiones.

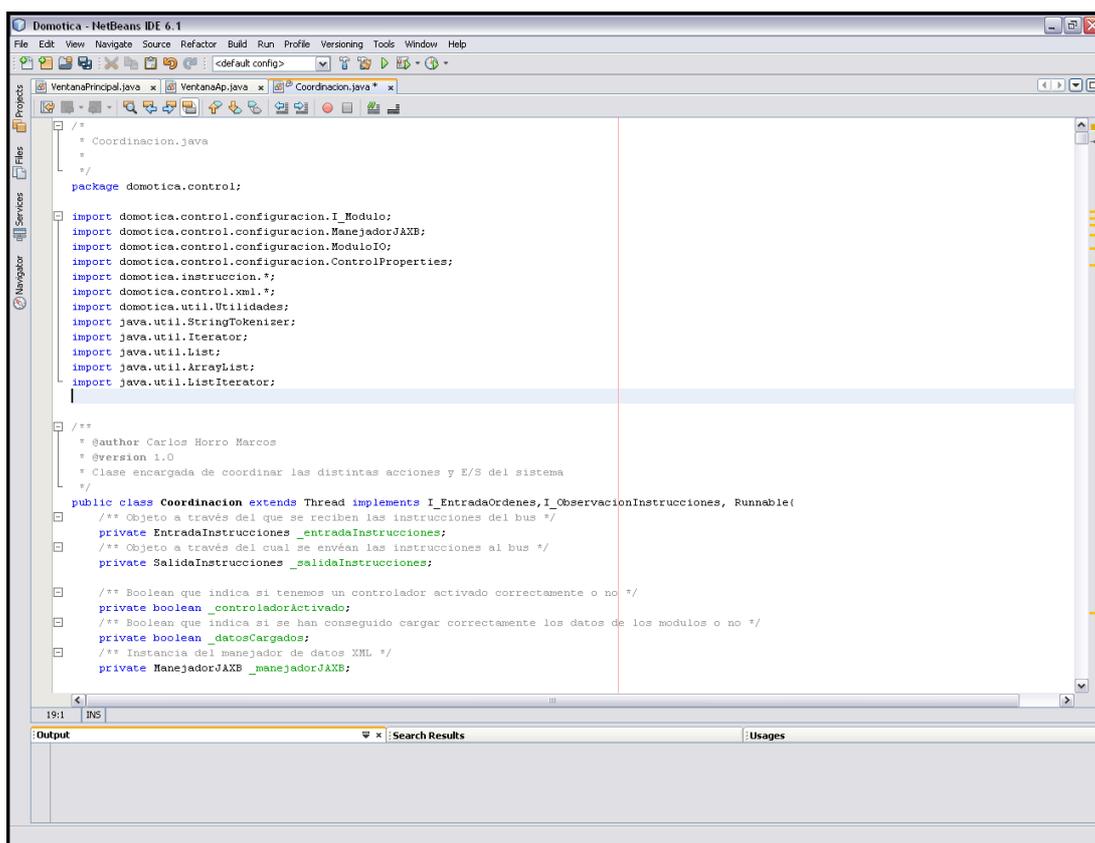


Ilustración 6: Código fuente del presente proyecto en Netbeans 6.1

3. OpenOffice

Se ha utilizado la versión 2.4 de OpenOffice para la realización de esta documentación principalmente por su faceta de *Suite* abierta y gratuita, además de ofrecer una funcionalidad muy amplia que se acerca mucho a la de la principal aplicación (propietaria) del sector, el *Office* de Microsoft.

Además, posee versiones para cualquier sistema, plataforma e idioma, lo que representa una facilidad enorme para la elaboración de documentos en cualquier ordenador que se disponga.

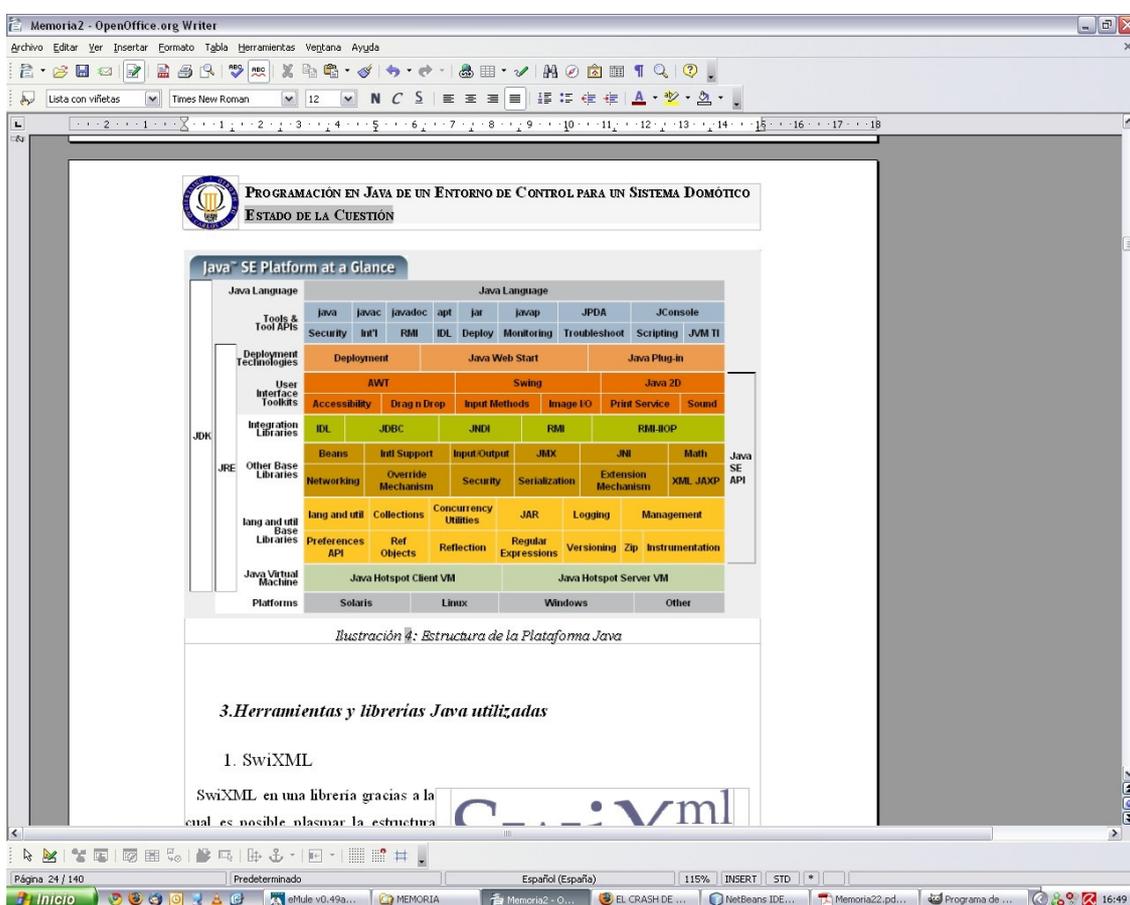


Ilustración 7: Openoffice writer editando la presente memoria

Capítulo 3

ESTUDIO DEL SISTEMA DE ERGODOMUS

1. LA EMPRESA

Este PFC es el resultado de una oferta de proyectos realizada por Ergodomus S.L, empresa en fase de desarrollo del Vivero de Empresas de la Universidad Carlos III de Madrid, donde se realiza el presente proyecto.



Ilustración 8: Logotipo de Ergodomus

Ergodomus, como empresa, es una idea de Enrique Verdú Calatayud, antiguo estudiante de la Universidad Carlos III de Madrid. Ésta resultó premiada en el *Primer concurso de creación de empresas de base tecnológica* promovido por dicha universidad en 2003. Además, participó en el *I Congreso de ideas y proyectos empresariales: Reemprende 04*, y aunque no resultó premiada, recibió muy buenas críticas del Grupo Recoletos.

El sector de mercado de Ergodomus es el sector de la domótica, y sus objetivos empresariales a corto plazo son:

- Desarrollar e introducir en el mercado un sistema domótico competitivo de calidad, y de bajo coste.
- Implantarse en el mercado español como desarrollador e instalador para viviendas de clase media.

2. DESCRIPCIÓN DEL SISTEMA

1. Una visión general

El principal objetivo de *Ergodomus* ha sido realizar un sistema domótico flexible y de bajo coste basado en unos módulos diseñados y programados por la propia empresa que pueden añadirse fácilmente a una instalación ya existente y a los que se pueden conectar directamente cualquier tipo de dispositivo que se deseen controlar mediante el sistema domótico. Esto es, no es necesario que sensores y actuadores sean del mismo fabricante ni que usen los mismos protocolos de comunicación.

Por ejemplo, esto permite que en la iluminación se puedan utilizar pulsadores a 3,3 V de corriente continua en vez ir a tensión de red, reduciéndose el riesgo de electrocución.

En resumen, las características más importantes del sistema domótico desarrollado por *Ergodomus* (disponibles en su totalidad en la página de la empresa, [D-1]) son las siguientes:

- Es **descentralizado**, por lo que son posibles desde instalaciones pequeñas hasta grandes edificios.
- La **interconexión de módulos** se realiza mediante un bus de datos serie a alta velocidad (115200 baudios), permitiendo una sencilla instalación y un tránsito de datos fluido.
- **Comunicación principal** (y **alimentación** del sistema) por medio un único **cable** de seis hilos, un bus que permite simplificar mucho la conexión de los distintos componentes del sistema domótico.
- **Reprogramable** en marcha y sin obra mediante un PC o mediante un módulo de control y programación.
- **Conexión remota** que permite el control del hogar desde cualquier parte y recepción de alarmas técnicas y de robo.
- **Escalabilidad**: pueden añadirse nuevos módulos fácilmente para mejorar la instalación o añadir funcionalidades.
- **Sistema de alimentación ininterrumpido** en prevención de cortes de luz.
- **Pulsadores de baja tensión**, con lo que el cableado resulta más barato y seguro.

- **2 modos de pulsación:** un simple pulsador puede utilizarse para 2 funciones diferentes y ser reprogramado (Ej.: pulsación sencilla = encender/apagar luz central de la habitación; pulsación prolongada = apagar todas las luces de la habitación).
- **Integra** bajo un mismo sistema los diferentes automatismos del hogar (electrodomésticos, iluminación, calefacción, aire acondicionado, alarmas...).
- **Versátil:** se adapta a las necesidades y a los sistemas preinstalados.
- Riego automático y todo tipo de **temporizaciones**.

Ergodomus barajó la posibilidad de basar la comunicación entre módulos por radiofrecuencia (como WiFi, una solución adoptada por un cierto número de soluciones domóticas). Esta se desechó para el caso general (podría usarse para soluciones concretas) no por problemas de privacidad de datos (que pueden resolverse mediante sistemas de encriptación de las comunicaciones), sino porque un inhibidor de frecuencia ajeno al sistema podría incomunicar los módulos entre sí, constituyendo un grave riesgo de seguridad.

Con estos principios, los componentes utilizados son de un coste bastante inferior a la media de sistemas domóticos existentes en la actualidad.

2. El bus ED

El **bus ED** es un desarrollo de *Ergodomus* que se basa en el estándar RS-485 (o EIA-485), protocolo de comunicaciones diferencial en bus de la capa física del modelo OSI.

Este estándar es ampliamente usado en la industria por varios motivos, como la larga longitud de cableado que permite (así como la sencillez del mismo), su buena tolerancia a ruidos (al ser un bus diferencial), etc. Permite una velocidad máxima de 10 Mbps y conexión multipunto de tipo half-duplex.

Sin embargo, el que la especificación RS-485 sólo defina la capa física obliga a que sea necesario definir un protocolo de transporte que



Ilustración 9: Pila de protocolos OSI [1-3]



permita una comunicación eficaz, libre de errores y capaz de prever y resolver las colisiones entre mensajes de distintos módulos, entre un emisor y un receptor.

A continuación detallo una lista de las técnicas implementadas para superar estos problemas:

- El bus ED se basa en un **sistema de direcciones** consistente en que cada mensaje contiene la dirección del remitente y la dirección del destinatario. Estas direcciones son únicas e identifican inequívocamente a cada sistema de la instalación.
- Para evitar la pérdida de mensajes en la instalación (ya sea por colisión, ruidos, caída de un sistema intermedio, etc.) la estrategia que sigue el bus es: **un mensaje-una respuesta**. De esta forma, si un dispositivo ordena a una persiana subirse, la persiana responderá al dispositivo de origen confirmando el éxito o el fracaso de la operación.
- Por otra parte, para asegurar que una respuesta responde a un mensaje en concreto y no a otro, se usa un **byte de control**, que deberá ser el mismo para ambos mensajes. Este byte se va incrementando en el sistema, de forma que pase un tiempo considerable hasta que pueda volver a repetirse.
- El problema de las colisiones se resuelve mediante un sistema CSMA, que las evita en su mayor parte y detecta en caso de haber ocurrido, reenviando la información de nuevo cuando parezca estar libre el canal.
- Por último, el bus ED también incluye un sistema para verificar que los mensajes intercambiados no han sufrido ninguna modificación o degradación desde su origen hasta su destino mediante un **byte de CRC**.



3. Consideraciones finales

Ya que el objetivo comercial de *Ergodomus* es prestar un servicio domótico integral, su tarea no sólo se limita a la creación de los módulos y buses de control:

- Los sistemas domóticos se diseñan e instalan personalmente según las necesidades del cliente.
- *Ergodomus* garantiza la asistencia técnica en cuanto mantenimiento y ampliación de los sistemas domóticos instalados durante su vida útil.
- El sistema puede ser personalizado para optimizar su consumo y adaptarse a las condiciones naturales de la zona donde vaya a ser instalado. Esto puede permitir obtener un considerable ahorro económico.
- Los sistemas que no puedan ser desarrollados por *Ergodomus* (como, por ejemplo, sistemas de atención de alarmas), pueden ser contratados a terceros e integrados en el sistema domótico general sin ningún problema.



3. ESTUDIO DE LA SOLICITUD

La oferta de Proyecto de Fin de Carrera presentada por Ergodomus consiste en:

Programación en Java de un entorno de control para el sistema domótico de Ergodomus.

Tareas asignadas:

1. Recepción y envío de datos con el bus domótico.
2. Representación gráfica del estado de entradas y salidas del sistema en entorno Java 2D.
3. Recepción y almacenaje de errores y colisión de datos.

Tareas opcionales:

4. Representación mediante Java 3D.

El sistema a desarrollar cuenta con 3 tipos de restricciones principales:

- **Restricciones económicas:** La empresa cuenta con un presupuesto limitado.
- **Restricciones técnicas:** Las restricciones técnicas principales son los 3 ordenadores con los que cuenta la empresa, y las aplicaciones instaladas en ellos.
- **Restricciones operativas:** El trabajo desarrollado en este proyecto se basa en el ya desarrollado por *Ergodomus* (de bajo nivel), esto es, la fabricación de los módulos, el adaptador a PC, su programación, y el diseño del sistema en general. Estas partes están a cargo de 2 personas de la empresa.

Los objetivos generales de viabilidad teniendo en cuenta las restricciones anteriores son:

- Analizar el conjunto concreto de necesidades.
- Identificar los requisitos que deben ser cubiertos.
- Analizar la situación actual de la organización en materia de sistemas de información existentes.
- Identificar sistemas de información existentes (en la organización o en el mercado) que puedan facilitar el desarrollo del proyecto.



4. CONSIDERACIONES AMBIENTALES

La aplicación está diseñada para funcionar sobre cualquier hardware cuyo sistema operativo tenga disponible una JVM que soporte la versión 6 de Java. Esto asegura su portabilidad así como su integración por parte de los principales fabricantes.

Los entornos de uso habituales de la aplicación son tanto entornos de oficina como entornos caseros (en los hogares donde estén instalados los sistemas domóticos). Las condiciones del entorno requeridas para la aplicación vienen dadas por las del hardware usado, que será normalmente un equipo informático. En principio la única restricción ambiental destacable es la humedad, siendo aconsejable que sea menor del 50%. La temperatura idónea debe estar comprendida entre los 20 y los 35 grados centígrados.

5. REQUISITOS DE USUARIO

El este capítulo recoge una descripción general del producto a desarrollar a nivel de capacidades, restricciones, características del usuario, entorno operacional y dependencias a las que esté sujeto el sistema.

Los requisitos de usuario reflejan las necesidades que debe cubrir el sistema desde el punto de vista del cliente, que en este caso es *Ergodomus S.L.* A partir de estos requisitos se podrá entender lo que espera el cliente del sistema a construir y llevarlo a cabo con éxito.

Cada requisito tendrá asociado un identificador único que lo diferencie del resto de requisitos. El formato de este identificador es el siguiente:

XX-[YY]nnn

- **Requisito (XX):** Dos letras que indiquen si es un requisito de usuario (RU), software (RS) o hardware (RH).
- **Tipo de requisito ([YY]):** En caso de que fuera necesario, se pueden utilizar una o dos letras más para indicar el tipo de requisito (capacidad, restricción, operacional, de interfaz, etc.)
- **Número de requisito:** número identificador del requisito dentro de la clasificación de su tipo.

Siguiendo esta nomenclatura, los requisitos de usuario presentarán la siguiente estructura:

REQUISITO DE USUARIO – TIPO DE REQUISITO Y NÚMERO = RU – YYNN

Ejemplo: **RU-C01:** Requisito de Usuario de capacidad 1.

Los tipos de requisitos posibles entre los requisitos de usuario son los siguientes (entre paréntesis, la abreviatura utilizada en los identificadores):

- **Requisitos de capacidad (C):** Funciones y operaciones necesitadas por los usuarios para resolver un problema o lograr un objetivo determinado.
- **Requisitos de restricción (R):** Limitaciones impuestas por los usuarios en la forma en la que se puede lograr el objetivo propuesto. Restringen la forma en la que el sistema se construye y opera.

Los requisitos de usuario se presentarán en un formato de tabla que presente toda la información de forma compacta:

IDENTIFICADOR: REQUISITO DE EJEMPLO			
Descripción del requisito			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> Ergodomus	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Las características que están asociadas a cada requisito son:

- **Identificador:** Incluye un identificador esquemático y un título corto orientativo.
- **Descripción:** Explicación breve de un requisito.
- **Fuente:** Origen del requisito (el cliente o el analista).
- **Prioridad:** orden en la implementación del requisito. Si el requisito debe ser de los primeros en ser implementado el valor será *alta*, si es conveniente realizarlo pronto, *media*, y si es indiferente realizarlo en cualquier momento, *baja*.
- **Necesidad:** Indica si un requisito es imprescindible (alta), conveniente su inclusión (media), o no sería estrictamente necesario incluirlo (baja).
- **Estabilidad:** Informa de la probabilidad de que el requisito vaya a cambiar en el transcurso del desarrollo del proyecto. Si es *alta*, es muy difícil que cambie el requisito, si es *media*, indica que no se espera un cambio en el requisito, y si es *baja*, hay una gran probabilidad de que el requisito vaya a cambiar.

A continuación se especifican los requisitos de usuario que tendrá que cumplir el sistema propuesto.

1. Requisitos de Capacidad

RU-C01: ADMINISTRACIÓN DE MÓDULOS			
El sistema permitirá administrar los módulos de E/S a los que esté conectado.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C02: ADMINISTRACIÓN DE APARATOS			
El sistema permitirá administrar los aparatos conectados a los módulos de E/S que tenga configurados.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C03: COMPONENTES FÍSICOS DE LOS MÓDULOS			
Los módulos de E/S del sistema poseen 3 entradas analógicas, 6 entradas digitales, 1 salida analógica y 6 salidas digitales.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C04: PROPIEDADES DE UNA ENTRADA DIGITAL			
Una entrada digital presenta dos posibles comportamientos distintos, según se realice sobre ella una acción simple o doble.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja



RU-C05: PROPIEDADES DE UNA SALIDA ANALÓGICA			
Una salida analógica presenta un determinado estado inicial.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C06: PROPIEDADES DE UNA SALIDA DIGITAL			
Una salida digital presenta un determinado estado inicial (ya sea después de un reseteo o en condiciones normales) y unas determinadas condiciones para su apagado o encendido.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C07: COMPONENTES LÓGICOS DE LOS MÓDULOS			
Los módulos de E/S del sistema tienen programado un identificador y admiten hasta un máximo de 5 grupos de acciones programadas y 3 persianas lógicas.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C08: PROPIEDADES DE UN GRUPO DE ACCIONES			
Un grupo de acciones del módulo de E/S consiste en un máximo de diez acciones distintas sobre objetivos distintos, con una determinada prioridad.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja



RU-C09: PROPIEDADES DE UNA PERSIANA LÓGICA			
Una persiana lógica del módulo de E/S tiene unos determinados tiempos de subida y de bajada.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C10: SITUACIÓN DE LOS MÓDULOS			
Los módulos de E/S del sistema estarán caracterizados por una posición y una habitación.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C11: PROPIEDADES DE LOS APARATOS			
Los aparatos del sistema tienen un nombre, tipo, identificador de su conexión a un módulo de E/S, posición y una habitación.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C12: INTERACCIÓN CON EL SISTEMA DOMÓTICO			
La aplicación permitirá interactuar con el sistema domótico de forma sencilla, rápida y al mismo tiempo completa.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja



RU-C13: REPROGRAMACIÓN DE LOS MÓDULOS			
La aplicación permitirá modificar la programación interna de los módulos de E/S.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C14: <i>BACKUP</i> DE LA PROGRAMACIÓN DE LOS MÓDULOS			
La aplicación permitirá guardar la programación interna de los módulos de E/S.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-C15: REGISTRO DE ACTIVIDAD			
La aplicación mantendrá un registro de la actividad del sistema.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> <i>Ergodomus</i>	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

2.Requisitos de Restricción

RU-R01: COMUNICACIÓN POR PUERTO SERIE			
La aplicación se comunicará por defecto con los módulos de E/S del sistema a través del puerto serie del ordenador.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> Ergodomus	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-R02: FUNCIONAMIENTO DE LA APLICACIÓN			
La aplicación se desarrollará en Java y funcionará de forma estable en un ordenador con puerto serie que tenga instalada una máquina virtual Java en su versión 6 como mínimo, con una resolución mínima de 800x600.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> Ergodomus	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-R03: TIPO DE DISPOSITIVOS CONECTADOS A UN MÓDULO			
A cada conexión física de un módulo de E/S sólo se podrá conectar un dispositivo compatible con el tipo de conexión.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> Ergodomus	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RU-R04: OPCIONES DE LA APLICACIÓN			
La aplicación permitirá elegir entre el inglés y el español para su manejo, además de las opciones que se consideren necesarias para un mejor uso de la misma.			
Fuente:	<input type="checkbox"/> Analista	<input type="checkbox"/> Ergodomus	
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja



RU-R05: MANTENIMIENTO DE LA APLICACIÓN						
La aplicación estará diseñada y documentada de tal forma que permitirá un sencillo mantenimiento y ampliación de funcionalidades.						
Fuente:	<input type="checkbox"/>	Analista	<input type="checkbox"/>	<i>Ergodomus</i>		
Prioridad:	<input type="checkbox"/>	Alta	<input type="checkbox"/>	Media	<input type="checkbox"/>	Baja
Necesidad:	<input type="checkbox"/>	Esencial	<input type="checkbox"/>	Deseable	<input type="checkbox"/>	Opcional
Estabilidad:	<input type="checkbox"/>	Alta	<input type="checkbox"/>	Media	<input type="checkbox"/>	Baja

Capítulo 4

ANÁLISIS DEL SISTEMA

1. ESPECIFICACIÓN DE CASOS DE USO

Para la especificación de los casos de uso del sistema, se utilizará el diagrama de casos de uso junto con la descripción de cada uno de ellos.

1. Diagrama de Casos de Uso

En este diagrama se representan las funcionalidades y comportamientos del sistema ante su interacción con el usuario.

Este diagrama sólo cuenta con un actor, el usuario, que es al mismo tiempo el cliente al que va dirigida la aplicación.

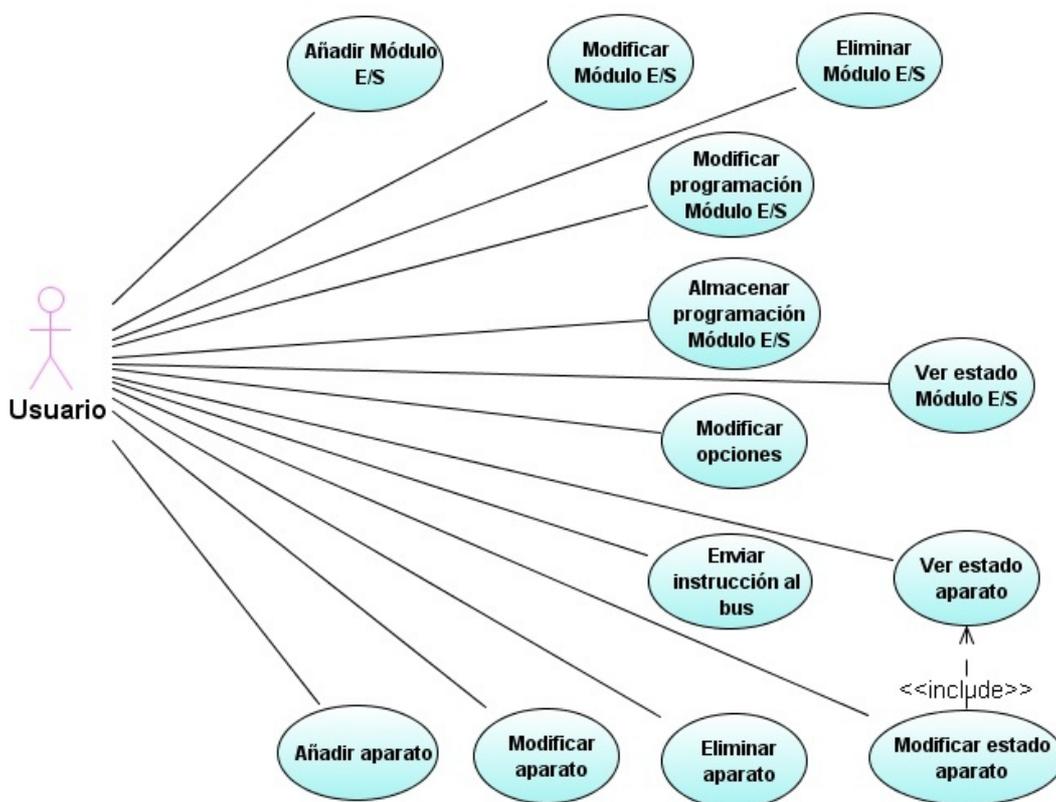


Ilustración 10: Diagrama de Casos de Uso

2.Descripción de los Casos de Uso

Para describir adecuadamente los casos de uso de la aplicación, se utilizarán tablas como las siguientes:

IDENTIFICADOR ÚNICO DEL CASO DE USO	
Descripción corta	
Actores	Roles posibles que pueden tomar los usuarios a la hora de interactuar con el sistema.
Objetivo	Servicio que el actor busca conseguir.
Precondiciones	Descripción del conjunto de estados del sistema anteriores a la ejecución del caso de uso.
PostCondiciones	Descripción del conjunto de estados del sistema posteriores a la ejecución del caso de uso.
Escenario básico	Secuencia de acciones principales (información intercambiada) en la interacción del escenario básico.

Tabla I. Caso de uso de ejemplo: descripción

A continuación se pasa a listar el conjunto de casos de uso según la tabla anterior:

CU-01	
Añadir Módulo E/S	
Actores	Usuario.
Objetivo	Incluir en la aplicación un módulo de E/S existente físicamente.
Precondiciones	El módulo de E/S todavía no está incluido.
PostCondiciones	El módulo de E/S aparece en la interfaz gráfica.
Escenario básico	<ul style="list-style-type: none"> ➢ El usuario pulsa sobre el botón de “añadir módulo”. ➢ La aplicación muestra una ventana con los campos a introducir. ➢ El usuario introduce el identificador del nuevo módulo. ➢ El usuario pulsa el botón de “aceptar”.

Tabla II. Caso de uso CU-01: Añadir Módulo E/S

CU-02	
Modificar Módulo E/S	
Actores	Usuario.
Objetivo	Cambiar el comportamiento del módulo de E/S.
Precondiciones	El módulo de E/S aparece en la interfaz gráfica.
PostCondiciones	La nueva especificación se guarda en la interfaz.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario selecciona el módulo de E/S cuya información desea modificar. ➤ La aplicación muestra un menú de opciones. ➤ El usuario selecciona “cambiar”. ➤ La aplicación muestra una ventana con la información del módulo. ➤ El usuario pulsa sobre el botón de “entradas”. ➤ La aplicación muestra una ventana con todas las entradas del módulo. ➤ El usuario modifica en una de las entradas digitales el destino de la acción simple. ➤ El usuario pulsa el botón de “aceptar” en las dos ventanas.

Tabla III. Caso de uso CU-02: Modificar Módulo E/S

CU-03	
Eliminar Módulo E/S	
Actores	Usuario.
Objetivo	Eliminar de la aplicación un módulo de E/S existente en ella.
Precondiciones	El módulo de E/S aparece en la interfaz gráfica.
PostCondiciones	El módulo de E/S ya no aparece en la interfaz gráfica.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario selecciona el módulo de E/S que desea eliminar. ➤ La aplicación muestra un menú de opciones. ➤ El usuario selecciona “eliminar”.

Tabla IV. Caso de uso CU-03: Eliminar Módulo E/S

CU-04	
Ver estado módulo E/S	
Actores	Usuario.
Objetivo	Ver en pantalla el estado de un determinado módulo de E/S.
Precondiciones	El módulo de E/S aparece en la interfaz gráfica
PostCondiciones	El estado del módulo de E/S aparece en pantalla.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario pulsa sobre el módulo de E/S del que desea conocer su estado. ➤ El estado del módulo de E/S aparece en un panel lateral.

Tabla V. Caso de uso CU-04: Ver estado módulo E/S

CU-05	
Modificar programación módulo E/S	
Actores	Usuario.
Objetivo	Modificar el comportamiento de un módulo de E/S o alguno de sus aparatos.
Precondiciones	El módulo de E/S aparece en la interfaz gráfica. El módulo físico de E/S opera normalmente.
PostCondiciones	El módulo físico de E/S pasa a comportarse según la nueva especificación.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario selecciona el módulo de E/S cuya programación desea modificar. ➤ La aplicación muestra un menú de opciones. ➤ El usuario selecciona “cambiar”. ➤ La aplicación muestra una ventana con la información del módulo. ➤ El usuario selecciona “entradas”. ➤ La aplicación muestra una ventana con todas las entradas del módulo. ➤ El usuario modifica en una de las entradas digitales el destino de la acción simple. ➤ El usuario elige “reprogramar entrada”. ➤ El usuario elige “aceptar” en las dos ventanas.

Tabla VI. Caso de uso CU-05: Modificar programación módulo E/S

CU-06	
Almacenar programación módulo E/S	
Actores	Usuario.
Objetivo	Cargar en la interfaz la programación interna del módulo de E/S para consultarla o poder guardarla para hacer <i>backup</i> de sus datos.
Precondiciones	El módulo de E/S aparece en la interfaz gráfica. El módulo físico de E/S opera normalmente.
PostCondiciones	La información de la programación del módulo físico de E/S aparece en sus ventanas de conexiones.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario selecciona el módulo de E/S cuya programación desea almacenar. ➤ La aplicación muestra un menú de opciones. ➤ El usuario selecciona “cambiar”. ➤ La aplicación muestra una ventana con la información del módulo. ➤ El usuario pulsa en “Actualizar todo”. ➤ El usuario pulsa en “aceptar”.

Tabla VII. Caso de uso CU-06: Almacenar programación módulo E/S

CU-07	
Añadir aparato	
Actores	Usuario.
Objetivo	Incluir en la aplicación un aparato existente físicamente.
Precondiciones	El aparato no aparece en la interfaz gráfica.
PostCondiciones	El aparato aparece en la interfaz gráfica.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario pulsa sobre el botón “añadir aparato”. ➤ La aplicación muestra una ventana con los campos a introducir. ➤ El usuario selecciona el módulo de E/S al que está conectado el aparato físico. ➤ La aplicación muestra los tipos de conexiones soportados por el módulo de E/S. ➤ El usuario selecciona el tipo de conexión del módulo al que está conectado el aparato (entrada, salida...). ➤ La aplicación muestra las conexiones disponibles de ese tipo en el módulo. ➤ El usuario selecciona el número de conexión al que está conectado el aparato. ➤ El usuario introduce el nombre y tipo del aparato en cuestión. ➤ El usuario pulsa en “aceptar”.

Tabla VIII. Caso de uso CU-07: Añadir aparato

CU-08	
Modificar aparato	
Actores	Usuario.
Objetivo	Modificar en la aplicación las características de un aparato.
Precondiciones	El aparato aparece en la interfaz gráfica.
PostCondiciones	El aparato aparece en la interfaz gráfica con las nuevas características.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario selecciona el aparato cuyas características desea modificar. ➤ La aplicación muestra un menú con distintas opciones. ➤ El usuario selecciona “cambiar”. ➤ La aplicación muestra una ventana con la información del aparato. ➤ El usuario cambia el tipo del aparato por otro. ➤ El usuario pulsa “aceptar”.

Tabla IX. Caso de uso CU-08: Modificar aparato

CU-09	
Eliminar aparato	
Actores	Usuario.
Objetivo	Elimina de la aplicación un aparato.
Precondiciones	El aparato aparece en la interfaz gráfica.
PostCondiciones	El aparato ya no aparece en la interfaz gráfica.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario selecciona el aparato que desea eliminar. ➤ La aplicación muestra un menú con distintas opciones. ➤ El usuario selecciona “eliminar”.

Tabla X. Caso de uso CU-09: Eliminar aparato

CU-10	
Ver estado aparato	
Actores	Usuario.
Objetivo	Ver en pantalla el estado de un determinado aparato.
Precondiciones	El aparato aparece en la interfaz gráfica
PostCondiciones	El estado del aparato aparece en pantalla.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario pulsa sobre un aparato. ➤ El estado del aparato aparece en un panel lateral

Tabla XI. Caso de uso CU-10: Ver estado aparato

IDENTIFICADOR: CU-11	
Modificar estado aparato	
Actores	Usuario.
Objetivo	Ver en pantalla el estado de un determinado dispositivo.
Precondiciones	El aparato debe estar conectado a una salida para poder modificar su estado.
PostCondiciones	El estado del aparato se ha modificado según la petición del usuario.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario pulsa sobre un aparato. ➤ El estado del mismo aparato aparece en un panel lateral, junto con los botones para cambiarlo. ➤ El usuario pulsa en los controles de cambio de estado. ➤ La aplicación muestra el nuevo estado del aparato tal cual solicitó el usuario.

Tabla XII. Caso de uso CU-11: Modificar estado aparato

IDENTIFICADOR: CU-12	
Modificar opciones	
Actores	Usuario.
Objetivo	Modificar el comportamiento de la interfaz a través de parámetros configurables.
Precondiciones	
PostCondiciones	La interfaz gráfica pasa a comportarse según los parámetros introducidos.
Escenario básico	<ul style="list-style-type: none"> ➤ El usuario pulsa sobre el botón “opciones”. ➤ La aplicación muestra una ventana con los parámetros configurables que hay disponibles. ➤ El usuario modifica la frecuencia de actualización de la pantalla. ➤ El usuario pulsa “aceptar”.

Tabla XIII. Caso de uso CU-12: Modificar opciones



IDENTIFICADOR: CU-13	
Enviar instrucción al bus	
Actores	Usuario.
Objetivo	Comunicarse a bajo nivel con algún dispositivo del sistema domótico, sin pasar por los elementos gráficos.
Precondiciones	
PostCondiciones	El dispositivo al que se ha enviado la instrucción responde a la misma.
Escenario básico	<ul style="list-style-type: none">➤ El usuario selecciona “consola”.➤ La aplicación muestra una ventana con las distintas posibilidades existentes de envío de instrucciones.➤ El usuario elige “pedir configuración”.➤ El usuario introduce el identificador del elemento del que desea una respuesta con su configuración.➤ El usuario pulsa en “enviar” para realizar su petición.

Tabla XIV. Caso de uso CU-13: Enviar instrucción al bus

2. MODELO CONCEPTUAL

En este apartado se presenta el diagrama de clases conceptual. Este diagrama está pensado para facilitar la comprensión de cómo se realizará el acercamiento a la estructura definitiva de la aplicación.

1. Diagrama de clases conceptual

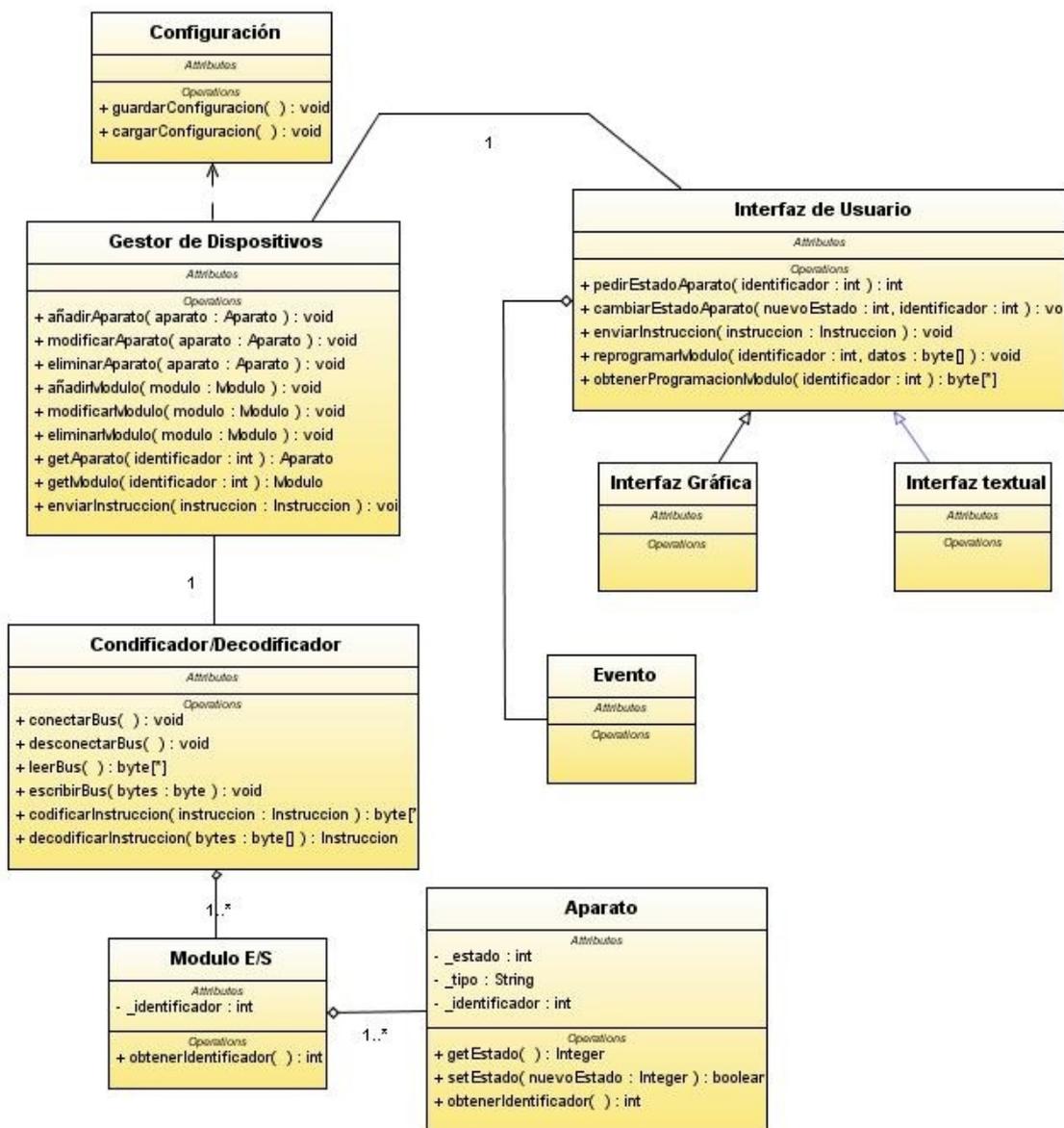


Ilustración 11: Diagrama de clases conceptual

2. Descripción del diagrama

A continuación paso a realizar una descripción de las clases presentes en el diagrama conceptual:

- **Gestor de Dispositivos:** representa el *corazón* de la aplicación, la parte encargada de hacer que las acciones del usuario en la interfaz se traduzcan en información transportada por el bus de datos hasta los módulos de E/S y sus aparatos respectivos, así como de hacer que el estado de todos estos pueda ser visto por el propio usuario. Debe presentar una amplia gama de funcionalidades, disponibles a todas las interfaces que ofrezca el sistema al usuario, así como también gestionar la información del sistema de forma fiable.
- **Interfaz de Usuario:** representa el compendio de funcionalidades que debe ofrecer al usuario cualquier interfaz de usuario que esté presente en el sistema. Las interfaces de usuario deberán ser intercambiables, de forma que el sistema pueda usarse ya sea a través de la interfaz gráfica, de una interfaz textual, a distancia a través de una web, etc.
- **Interfaz Gráfica:** es la interfaz de usuario central de la aplicación. Debe ser capaz de presentar de forma vistosa el estado del sistema domótico al que está conectado el sistema, facilitar con pocas pulsaciones del ratón la modificación del estado de sus aparatos, agregar nuevos aparatos o módulos ya existentes físicamente, y proporcionar una vía alternativa para permitir además la comunicación directa con los dispositivos del bus.
- **Interfaz Textual:** representa una (tal vez la principal) de las muchas interfaces de usuario alternativas que podrá soportar la aplicación en un futuro.
- **Evento:** representa un suceso del sistema recogido por la interfaz.
- **Configuración:** la parte del sistema encargada de manejar la información sobre el sistema domótico que posea la aplicación. Ésta se llevará a cabo a través de un archivo XML desde el que se cargará (y en el que se guardarán) la información existente sobre módulos y aparatos, gracias a la tecnología JAXB.
- **Codificador/Decodificador:** representa la parte del sistema encargada de transformar las órdenes de la aplicación en cadenas de bytes entendibles por el sistema domótico, y viceversa. Es el nexo de unión entre la parte física (la infraestructura del sistema domótico) y la parte lógica (el software de la aplicación) que hace posible la interacción inmediata entre ellas.



- **Módulo E/S:** Representa los módulos de E/S fabricados por *Ergodomus S.L* que contienen la lógica interna del sistema domótico. Son la base del mismo, ya que se comunican entre ellos y con la aplicación domótica, además de permitir a través de ellos el manejo de los aparatos que tengan conectados.
- **Aparato:** Esta clase representa todos y cada uno de los posibles aparatos que pueden conectarse al sistema domótico, que son prácticamente ilimitados. La conexión a los módulos de E/S permite abstraernos en buena medida de sus complejidades particulares y nos ofrece un manejo sencillo de los mismos al crear un identificador para cada uno de ellos.

3. OBTENCIÓN DE REQUISITOS SOFTWARE

En el presente apartado se va a dar una visión del proyecto desde el punto de vista del desarrollo. Partiendo del capítulo anterior, se realiza un análisis orientado al software, del que se hace una descripción general, un modelado conceptual y un diagrama de casos de uso, así como los requisitos software de la aplicación.

Posteriormente se pasará a realizar un análisis por componentes del sistema, mucho más detallado.

Siguiendo la nomenclatura explicada en el capítulo anterior, los requisitos software presentarán la siguiente estructura:

REQUISITO SOFTWARE – TIPO DE REQUISITO Y NÚMERO = RS – XX

Los tipos de requisitos posibles en el análisis son los siguientes (entre paréntesis, la abreviatura utilizada en los identificadores):

- **Requisitos funcionales (F)**: Especifican la funcionalidad o servicios que el sistema o un componente del sistema debe proporcionar.
- **Requisitos de rendimiento (R)**: Especifican valores numéricos para variables medibles (por ejemplo, frecuencia, velocidad)... Las restricciones temporales deben realizarse mediante valores numéricos (cuantitativos), no cualitativos.
- **Requisitos de interfaz (I)**: Describen la forma en que el sistema se comunica con otros sistemas externos (hardware, software, o interfaces de comunicaciones).
- **Requisitos operacionales (O)**: Especifican cómo funciona el sistema y cómo se comunica cuando es usado por los usuarios. Esto incluye requisitos de interfaz de usuario, interacción persona-máquina y usabilidad, pero también requisitos logísticos y de organización.
- **Requisitos de recursos (U)**: Especifican los máximos recursos físicos disponibles para el sistema, tales como capacidad de procesamiento, de memoria, de disco duro...

- **Requisitos de verificación (V):** Definen cómo el software debe ser verificado, es decir, especifican las medidas a tomar para comprobar que el sistema proporciona las funcionalidades descritas en los requisitos especificados.
- **Requisitos de pruebas de aceptación (A):** Definen cómo el software debe ser validado, es decir, especifican las medidas a tomar para asegurarse de que el sistema satisface los requisitos de uso previstos.
- **Requisitos de documentación (D):** Especifican requisitos específicos para documentos de este proyecto, como puede ser el manual de usuario.
- **Requisitos de seguridad ante amenazas (SA):** Especifican las medidas propuestas para asegurar al sistema ante amenazas contra la integridad, disponibilidad o confidencialidad de su información.
- **Requisitos de portabilidad (P):** Especifican las modificaciones necesarias a realizar en el software para que éste pueda funcionar en otros ordenadores o sistemas operativos.
- **Requisitos de calidad (C):** Especifican las propiedades del sistema (cualitativa o cuantitativamente según convenga) que asegurarán su idoneidad para cumplir con el objetivo marcado.
- **Requisitos de fiabilidad (FB):** Especifican un tiempo medio entre fallos aceptable en el sistema o uno de sus componentes.
- **Requisitos de mantenimiento (M):** Recogen el grado en que es posible resolver problemas del sistema o añadir nuevas funcionalidades al mismo, de forma cuantitativa siempre que sea posible.
- **Requisitos de seguridad ante daños físicos (SF):** Describen procedimientos para reducir las posibilidades de daños a personas o propiedades en caso de fallo del sistema.

Los requisitos software se presentarán en un formato de tabla que presente toda la información de forma compacta:

IDENTIFICADOR: REQUISITO SOFTWARE DE EJEMPLO			
Descripción del requisito			
Fuente:			
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Las características que están asociadas a cada requisito son:

- **Descripción:** Descripción corta del requisito.
- **Fuente:** Origen del requisito.
- **Prioridad:** orden en la implementación del requisito. Si el requisito debe implementarse de los primeros el valor será “alta”, si es conveniente realizarlo pronto, “media”, y si es indiferente realizarlo en cualquier momento, “baja”.
- **Necesidad:** Indica si un requisito es imprescindible (alta), conveniente su inclusión (media), o no sería estrictamente necesario incluirlo (baja).
- **Estabilidad:** Informa de la probabilidad de que el requisito vaya a cambiar en el transcurso del desarrollo del proyecto. Si es alta, es muy difícil que cambie el requisito, si es media, indica que no se espera un cambio en el requisito, y si es baja, hay una gran probabilidad de que el requisito vaya a cambiar.
- **Claridad:** Grado en el que el requisito no resulta ambiguo o equívoco.
- **Verificabilidad:** Grado en el que puede medirse el éxito al llevar a cabo el requisito.

4.Requisitos Funcionales

RS-F01: REPRESENTACIÓN INTERNA DEL SISTEMA			
La aplicación mantendrá una representación interna propia del sistema domótico, que podrá coincidir o no con los módulos de E/S y los aparatos instalados físicamente en él.			
Fuente:	RU-C01, RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F02: ADMINISTRACIÓN DE MÓDULOS			
La aplicación permitirá añadir, modificar o eliminar módulos de E/S a su representación interna, existan estos físicamente o no.			
Fuente:	RU-C01		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F03: OBTENCIÓN DEL ESTADO DE LOS MÓDULOS			
La aplicación permitirá conocer el estado de funcionamiento de los módulos de E/S físicos a través de su representación virtual en la aplicación.			
Un módulo de E/S puede presentar dos estados de funcionamiento: operativo o inoperativo.			
Fuente:	RU-C01		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F04: ADMINISTRACIÓN DE APARATOS			
La aplicación permitirá añadir, modificar o eliminar aparatos a su representación interna, existan estos físicamente o no.			
Fuente:	RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F05: OBTENCIÓN DEL ESTADO DE LOS APARATOS			
La aplicación permitirá conocer el estado de funcionamiento de los aparatos físicos a través de su representación virtual en la aplicación.			
Un aparato puede presentar un máximo de tres estados de funcionamiento: encendido, apagado o inoperativo.			
Fuente:	RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F06: MODIFICACIÓN DEL ESTADO DE LOS APARATOS			
La aplicación permitirá modificar el estado de funcionamiento de los aparatos físicos que estén conectados a alguna de las salidas de su módulo de E/S correspondiente, a través de su representación virtual en la aplicación.			
Fuente:	RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F07: ELIMINACIÓN DE APARATOS EN CASCADA			
En la aplicación, los aparatos conectados a un módulo de E/S serán eliminados cuando se elimine dicho módulo.			
Fuente:	RU-C01, RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F08: CARACTERÍSTICAS FÍSICAS DE LOS MÓDULOS DE E/S			
Los módulos de E/S físicos utilizados poseen 3 entradas analógicas, 6 entradas digitales, 1 salida analógica y 6 salidas digitales.			
Fuente:	RU-C03		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F09: CARACTERÍSTICAS DE UNA ENTRADA ANALÓGICA			
Una entrada analógica física no realiza ningún tratamiento o modificación de la información recibida. La entrada analógica <i>virtual</i> correspondiente sólo estará caracterizada por tanto por la información relativa al aparato que tenga conectado, en caso de tenerlo.			
Fuente:	RU-C03		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F10: CARACTERÍSTICAS DE UNA ENTRADA DIGITAL

Una entrada digital presenta dos posibles reacciones distintas (con una acción, prioridad y destino determinado) según reciba una pulsación simple o doble.

Además, se pueden dar 4 tipos de sistemas de entrada de información en una entrada digital: tipo interruptor, tipo pulsador, tipo interruptor incremental y tipo interruptor de persiana.

Por último, la entrada digital estará caracterizada por los datos del aparato que tenga conectado, en caso de tenerlo.

Fuente:	RU-C04		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F11: CARACTERÍSTICAS DE UNA SALIDA ANALÓGICA

Una salida analógica está caracterizada por un determinado estado inicial, además de los datos del aparato que tenga conectado, en caso de tenerlo.

Fuente:	RU-C05		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F12: CARACTERÍSTICAS DE UNA SALIDA DIGITAL

Una salida digital está caracterizada por un determinado estado inicial en el encendido y otro estado inicial tras un reinicio, además de los datos del aparato que tenga conectado, en caso de tenerlo.

Fuente:	RU-C06		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F13: CONDICIONES DE UNA SALIDA DIGITAL

Una salida digital está caracterizada por dos condiciones, una para el apagado y otra para el encendido. Cada una es el resultado (negado o no) de una operación (que puede ser de suma o de resta, en caso de ser analógica, o *AND* y *OR* de ser digital) entre, a su vez, el resultado de las operaciones internas en dos grupos de un máximo de 5 elementos cada uno.

Cada elemento de un grupo puede consistir en un valor numérico que represente un dato, o en una dirección.

Asimismo, cada operación entre elementos puede ser del mismo modo analógica o digital, y su resultado negado o no.

Fuente:	RU-C06		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F14: CARACTERÍSTICAS LÓGICAS DE LOS MÓDULOS DE E/S

Los módulos de E/S del sistema tienen programado un identificador que debe ser único en toda la infraestructura del sistema domótico, ya que les identifica inequívocamente en él y permite que se comuniquen entre ellos.

Además, los módulos de E/S presentan hasta un máximo de 5 direcciones para grupos de acciones programadas y 3 direcciones con comportamiento especial para manejo de persianas.

Fuente:	RU-C07		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F15: CARACTERÍSTICAS DE UN GRUPO DE ACCIONES PROGRAMADAS

Un grupo de acciones programadas en un módulo de E/S consiste en una dirección del mismo ante la que la llegada de una orden de ejecución del grupo lanza un máximo de diez acciones distintas sobre objetivos distintos con una determinada prioridad.

Un grupo de acciones programadas también puede recibir una orden de ejecución inversa, en cuyo caso las acciones ejecutadas serán las contrarias a las lanzadas normalmente (las acciones serán las contrarias a criterio de la programación del módulo: por ejemplo, apagar en vez de encender).

Cada grupo de acciones se corresponderá a un aparato *virtual* más, con todas sus características asociadas, y se ofrecerá una descripción del objetivo de cada acción.

Fuente:	RU-C08		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F16: CARACTERÍSTICAS DE UNA PERSIANA LÓGICA

Una persiana lógica consiste en una dirección del módulo de E/S que se corresponde con la misma salida física que una de sus direcciones de salidas normales, con la peculiaridad de que su comportamiento ante las órdenes es especial para manejar persianas.

De este modo, una persiana lógica estará caracterizada por unos determinados tiempos programados para subir y bajar la persiana, y se corresponderá a un aparato *virtual* normal.

Fuente:	RU-C09		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F17: CARACTERÍSTICAS DE SITUACIÓN DE LOS MÓDULOS

Los módulos de E/S del sistema estarán situados en unas coordenadas determinadas, que tendrán tres componentes (x,y,z) si se usa un plano en 3 dimensiones o sólo 2 (x e y; con la z con valor 0) si se usa el plano normal de dos dimensiones.

Estas coordenadas serán relativas (en tanto por uno) al tamaño de la casa y, al mismo tiempo, por tanto también al del mapa de la casa en la aplicación, de forma que se podrá hacer una correspondencia directa entre la situación de un módulo en la aplicación y su situación en la casa física.

Además, los módulos de E/S estarán situados en una habitación en particular, que se identificará a través de un número.

Fuente:	RU-C10		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F18: CARACTERÍSTICAS DE LOS APARATOS

Los aparatos del sistema domótico estarán caracterizados por un nombre, su tipo de aparato, el identificador de su conexión a un módulo de E/S, y por su situación y habitación en la casa.

Sólo se considerará que un aparato está conectado a una de las entradas o salidas de un módulo de E/S cuando en las equivalentes de su representante *virtual* conste un nombre de aparato válido (esto es, que tenga al menos un carácter del alfabeto latino).

El sistema de posicionamiento y de habitaciones de los aparatos funcionará de igual manera que el explicado para los módulos en el requisito RS-F15.

Fuente:	RU-C11		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F19: MÚLTIPLES INTERFACES DE USUARIO

La aplicación permitirá la inclusión sencilla de nuevas interfaces de usuario con las que el usuario podrá interactuar con el sistema domótico, como interfaces gráficas o interfaces textuales. La interfaz gráfica será la interfaz implementada por defecto en la aplicación.

Fuente:	RU-C12, RU-R05		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F20: MÚLTIPLES BUSES DE COMUNICACIÓN

La aplicación permitirá la inclusión sencilla en la aplicación de nuevos buses de comunicación, de forma que permita interactuar con el sistema domótico a través de los diferentes buses implementados, como el puerto serie o el USB. El bus de comunicación implementado por defecto por la aplicación será el puerto serie.

Fuente:	RU-C12, RU-R05		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F21: INTERFAZ GRÁFICA COMPLETA

La interfaz gráfica permitirá al usuario no sólo interactuar con el sistema domótico de forma visual e intuitiva, sino también proporcionar una vía de acceso para el uso de instrucciones de bajo nivel que permitan la comunicación con el sistema domótico a nivel de bytes.

Fuente:	RU-C12		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F22: VISUALIZACIÓN SELECTIVA

La interfaz gráfica permitirá al usuario mostrar todos los dispositivos presentes en la representación interna del sistema, o sólo los elementos que cumplan una o varias características simultáneamente. Estas características pueden ser:

- Conexión a un determinado módulo de E/S.
- Pertenencia a un determinado tipo de aparato.
- Situación en una habitación determinada.

Fuente:	RU-C12		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F23: REPROGRAMACIÓN DE LOS MÓDULOS DE E/S

La aplicación permitirá modificar la programación interna de los módulos de E/S en base a las características de sus representaciones *virtuales*.

Fuente:	RU-C13		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-F24: ACTUALIZACIÓN DE LA REPRESENTACIÓN INTERNA DE LOS MÓDULOS DE E/S

La aplicación permitirá sincronizar las características de los módulos de E/S *virtuales* en base a la programación interna de sus correspondientes módulos de E/S físicos del sistema domótico, de forma que esta representación interna actúe como un *backup* de dicha programación.

Fuente:	RU-C14		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja



RS-F25: HISTORIAL DE EVENTOS

La interfaz gráfica mostrará un resumen de los eventos más importantes ocurridos en el sistema domótico. El tipo de eventos recogido será: cambios de estado en aparatos y módulos de E/S (ya sea por parte del usuario o no) y añadido o borrado de dispositivos.

Fuente:	RU-C15		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

5.Requisitos de Rendimiento

RS-R01: TIEMPO DE RESPUESTA DE ESTADO			
El tiempo de respuesta de la aplicación a una petición de estado de un módulo de E/S o aparato no superará 1 segundo, sea cual sea éste.			
Fuente:	RU-C01, RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-R02: TIEMPO DE RESPUESTA A CAMBIO DE ESTADO			
El tiempo de respuesta de la aplicación a una petición de cambio de estado de un aparato no superará 1 segundo			
Fuente:	RU-C01, RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-R03: MÓDULOS Y APARATOS ILIMITADOS			
No habrá límite en el número de módulos controlados por la aplicación; el límite en el número de aparatos vendrá dado por las conexiones disponibles en los módulos de E/S.			
Fuente:	RU-C01, RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja



RS-R04: VALORES INICIALES EN LOS PARÁMETROS CONFIGURABLES

Los valores iniciales de los parámetros configurables son:

- Idioma: **español**.
- Frecuencia de actualización: **3 segundos**.
- Tamaño de los iconos: **64x64**.
- Número de sucesos registrados: **100**.
- Tipo del bus a usar: **serie**.
- Nombre del puerto serie: **COM1**.
- Velocidad del puerto serie: **115200**.
- Tipo de control de flujo de entrada al puerto serie: **ninguno**.
- Tipo de control de flujo de salida al puerto serie: **ninguno**.
- Bits de datos del puerto serie: **8**.
- Bits de stop del puerto serie: **1**.
- Paridad de datos del puerto serie: **ninguna**.

Fuente:	RU-C01, RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-R05: TIPOS DE APARATOS VÁLIDOS

Cada tipo de aparato presenta unas determinadas características:

- **Nº** : Código numérico que identifica el tipo.
- **Nombre**: Descripción textual corta del tipo de aparato.
- **Tipo**: Puede ser binario o gradual, según funcione en base a valores todo/nada o más amplios.
- **Conexiones**: Pueden ser de tipo entrada, salida, grupo o persiana. Según sea ésta, un mismo aparato se puede comportar de formas distintas.
- **Rango**: La gama de estados (siempre enteros) que contempla el aparato.
- **Estado apagado**: El estado (siempre dentro de su rango) en el que se considera que el aparato está apagado. Si es nulo, significa que el aparato no se puede considerar apagado nunca.

Los tipos de aparatos que contemplará el sistema, junto con sus características de funcionamiento, serán los siguientes:

Nº	Nombre	Tipo	Conexiones	Rango	Estado apagado
0	Interruptor	Binario	Entrada	0 : 1	Nulo
1	Pulsador	Binario	Entrada	0 : 1	Nulo
2	Pulsador incremental para luces	Gradual	Entrada	0 : 100	Nulo
3	Pulsador de persianas	Binario	Entrada	0 : 1	Nulo
9	Desconocido	Gradual	Entrada Salida Grupo Persiana	0 : 100	Nulo
10	Luz binaria	Binario	Salida	0 : 1	0
11	Bombilla incremental	Gradual	Salida	0 : 100	0
15	Persiana	Gradual	Persiana	0 : 100	Nulo
16	Grupo de persianas	Gradual	Grupo	0 : 100	Nulo
20	Enchufe binario	Binario	Entrada	0 : 100	Nulo
21	Enchufe regulado	Gradual	Entrada	0 : 100	Nulo
30	Escena	Binario	Grupo	0 : 1	Nulo
40	Termómetro	Gradual	Entrada	-60 : 120	Nulo
50	Sensor lumínico	Gradual	Entrada	0 : 100	0

52	Sensor de viento binario	Binario	Entrada	0 : 1	0
53	Sensor de viento analógico	Gradual	Entrada	0 : 100	0
60	Sensor volumétrico	Binario	Entrada	0 : 1	0
62	Sensor de apertura	Binario	Entrada	0 : 1	0
64	Sensor acústico	Binario	Entrada	0 : 1	0
65	Sensor de fuego	Binario	Entrada	0 : 1	0
66	Sensor de inundación	Binario	Entrada	0 : 1	0
68	Sensor de humedad	Gradual	Entrada	0 : 100	0
69	Sensor genérico	Binario	Entrada	0 : 1	0
70	Electroválvula de agua	Gradual	Entrada	0 : 100	0
71	Electroválvula de gas	Gradual	Entrada	0 : 100	0
71	Sirena acústica	Binaria	Salida	0 : 1	0
76	Strobe (avisador luminoso)	Binario	Salida	0 : 1	0
80	Aire acondicionado	Binario	Salida	0 : 1	0
85	Estufa	Binario	Salida	0 : 1	0
90	Aire acondicionado con bomba de calor	Binario	Salida	0 : 1	0
100	Alarma	Binario	Salida	0 : 1	0
102	Simulador de presencia	Binario	Entrada	0 : 1	Nulo
110	Climatización	Binario	Salida	0 : 1	Nulo
120	Programación	Binario	Salida	0 : 1	Nulo
130	Temporización	Binario	Salida	0 : 1	Nulo

Tabla XV. Tipos de aparatos soportados por la aplicación

Fuente:	RU-R03		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

6. Requisitos de interfaz

RS-I01: ELEMENTOS PARA LA CONEXIÓN			
Para la conexión con el sistema domótico físico se necesitará una conexión de puerto serie que cumpla el estándar RS-485 y un adaptador que convierta esta señal a la utilizada en el bus <i>ED</i> .			
Fuente:	RU-R01		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-I02: COMUNICACIÓN DE LA APLICACIÓN CON EL SISTEMA DOMÓTICO			
La aplicación se comunicará con la infraestructura del sistema domótico (esto es, con los módulos de E/S proporcionados por <i>Ergodomus S.L</i> y con los aparatos de ésta o cualquier otra empresa que se conecten a dichos módulos) mediante cadenas de bytes que siguen un protocolo patentado de dicha compañía.			
Fuente:	RU-R01		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-I03: ALMACENAMIENTO DE LA INFORMACIÓN			
La aplicación almacenará la información de configuración en archivos de propiedades de Java, mientras que la información relativa a los módulos de E/S y los aparatos se manejará a través de archivos XML que cumplan el esquema adecuado.			
Fuente:	RU-C13, RU-C14, RU-R05		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

7. Requisitos operacionales

RS-O01: INTERFAZ GRÁFICA			
<p>La principal interfaz de usuario de la aplicación será una interfaz gráfica a través de la que el usuario podrá acceder a toda la funcionalidad de la aplicación y visualizar y modificar el estado de los módulos y aparatos que se hayan añadido.</p> <p>Concretamente, a través de la ventana principal de la interfaz gráfica el usuario podrá:</p> <ul style="list-style-type: none"> ➤ Monitorizar el estado del sistema. ➤ Acceder a la consola del sistema. ➤ Configurar las opciones principales de la interfaz. ➤ Añadir un nuevo módulo, visualizar y modificar las propiedades de uno ya existente, o eliminarlo. ➤ Añadir un nuevo aparato, visualizar y modificar las propiedades de uno ya existente, o eliminarlo. ➤ Modificar la programación interna de un módulo de E/S físico. ➤ Almacenar su programación. ➤ Acceder al manual de usuario. ➤ Salir de la aplicación. 			
Fuente:	RU-C12		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-O02: VISUALIZACIÓN DE LA INFORMACIÓN DE UN MÓDULO			
<p>Al seleccionar un módulo de E/S, se mostrará su identificador, estado y habitación.</p>			
Fuente:	RU-C01		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-O03: VISUALIZACIÓN RÁPIDA DE LA INFORMACIÓN DE UN MÓDULO			
En la interfaz gráfica, al pasar el ratón sobre la imagen de un módulo se visualizarán sobre ella su identificador, su tipo (módulo de E/S) y su estado de funcionamiento (operativo o inoperativo).			
Fuente:	RU-C01		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-O04: VISUALIZACIÓN DE LA INFORMACIÓN DE UN APARATO			
Al seleccionar un aparato con el botón izquierdo del ratón, se mostrará en el panel de estado su nombre, tipo de aparato, habitación que ocupa, lugar de conexión y estado.			
Fuente:	RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-O05: VISUALIZACIÓN RÁPIDA DE LA INFORMACIÓN DE UN APARATO			
En la interfaz gráfica, al pasar el ratón sobre la imagen de un aparato se mostrará sobre ella su nombre, tipo de aparato (textual y numérico) y estado de funcionamiento (encendido, apagado o inoperativo).			
Fuente:	RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-O06: MODIFICACIÓN DEL ESTADO DE UN APARATO

Al seleccionar un aparato con el botón izquierdo del ratón se mostrarán en el panel de estado las opciones apropiadas (según su tipo y lugar de conexión) en cada caso para modificar su estado.

Fuente:	RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-O07: SISTEMA DE COLORES EN LA INTERFAZ GRÁFICA

Las imágenes de los módulos de E/S y de los aparatos seguirán un sistema de colores indicativo de su estado. Este sistema es el siguiente:

- **Amarillo:** Para módulos de E/S que estén funcionando con normalidad y aparatos que estén encendidos o en funcionamiento (ej: una bombilla encendida, un aire acondicionado funcionando, un termómetro, etc.)
- **Azul:** Para aparatos que estén apagados o no estén en funcionamiento (ej: un sensor que no está detectando nada o una alarma que no esté sonando).
- **Gris:** Para aparatos y módulos de E/S que no estén funcionando adecuadamente o cuyos representantes *virtuales* no tengan una correspondencia física.

Fuente:	RU-C01, RU-C02, RU-C12		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-O08: MENÚS CONTEXTUALES EN LA INTERFAZ GRÁFICA

Al pulsar con el botón derecho del ratón sobre un módulo de E/S o un aparato, aparecerá un menú contextual que permitirá de forma rápida modificar sus características o eliminarlo de la representación interna del sistema.

Fuente:	RU-C01, RU-C02, RU-C12		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-O09: MOVIMIENTO DE DISPOSITIVOS EN LA INTERFAZ GRÁFICA			
Al pulsar con el botón izquierdo del ratón sobre un módulo de E/S o un aparato y mantenerlo pulsado mientras se mueve el ratón (movimiento comúnmente llamado <i>drag&drop</i>), se podrá situar el dispositivo en cualquier lugar del mapa de dispositivos, modificando sus coordenadas en el momento de soltarlo sobre su lugar del destino.			
Fuente:	RU-C12		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

8.Requisitos de recursos

RS-U01: REQUISITOS DEL PC			
La aplicación funcionará en un ordenador con una velocidad de procesamiento mínima de 1,5Ghz, 512 megas de memoria RAM y soporte una resolución gráfica de 800x600 pixels. La aplicación necesita menos de 10 megas de disco duro para instalarse, pero a efectos de rendimiento se recomienda un mínimo de 500 megas libres.			
Fuente:	RU-R02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

9.Requisitos de verificación

RS-V01: COMPROBACIÓN DEL PARADIGMA			
Se comprobará que la aplicación se ha diseñado y desarrollado siguiendo el paradigma de la orientación a objetos.			
Fuente:	RU-R02, RU-R05		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

10. Requisitos de pruebas de aceptación

RS-A01: COMPROBACIÓN DE ESTADOS			
Se comprobará que los estados de los aparatos que aparecen en la aplicación se corresponden con los estados físicos reales de los mismos.			
Fuente:	RU-C01, RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-A02: COMPROBACIÓN DE MODIFICACIÓN DE ESTADOS			
Se comprobará que al ejecutar una acción de cambio de estado sobre un aparato que aparece en la aplicación, el aparato físico reproduce el mismo cambio tal cual se pidió.			
Fuente:	RU-C01, RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-A03: COMPROBACIÓN DE OPCIONES			
Se comprobará que al modificar el idioma, la frecuencia de actualización o el tamaño de los iconos, la interfaz reacciona como se espera.			
Fuente:	RU-R04		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-A04: COMPROBACIÓN DEL FUNCIONAMIENTO DE LA CONSOLA

Se comprobará que:

- al enviar una instrucción mediante la consola de órdenes, ésta realmente se envía.
- si ha sido enviada a una dirección de un módulo, se comprobará que se recibe la respuesta correctamente.
- se recogen todas las instrucciones que circulan por el bus y las enviadas a él desde la aplicación.

Fuente:	RU-C12		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-A05: COMPROBACIÓN DE LA REPROGRAMACIÓN DE UN MÓDULO

Se comprobará que tras una reprogramación de un módulo de E/S, éste pasa a funcionar exactamente como se especificó en su nueva programación.

Fuente:	RU-C13		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-A06: COMPROBACIÓN DEL REGISTRO DE ACTIVIDAD

Se comprobará que ante un encendido, apagado, nueva conexión, malfuncionamiento o eliminación de un aparato, y ante la creación, malfuncionamiento o eliminación de un módulo de E/S, el sistema registra lo ocurrido de forma que el usuario pueda enterarse de ello con posterioridad.

Fuente:	RU-C15		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-A07: COMPROBACIÓN DE INFORMACIÓN VÁLIDA			
Se comprobará que al introducir información en la consola, ventana de opciones, de aparatos o de módulos, estas ventanas no provocan inestabilidad en la aplicación ante datos de tipo incorrecto (cadenas de caracteres en campos numéricos, introducción de caracteres extraños, etc.).			
Fuente:	Desarrollo		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

11. Requisitos de documentación

RS-D01: MANUAL DE USUARIO			
A través de la aplicación se podrá acceder a un manual de usuario que permita conocer de forma sencilla las posibilidades de la aplicación.			
Fuente:	RU-R05		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-D02: DOCUMENTACIÓN JAVADOC			
Junto con la aplicación se entregará su documentación de código asociada Javadoc, de forma que permita a cualquier desarrollador o analista posterior conocer de forma rápida la estructura y el objetivo de cualquier parte del código.			
Fuente:	RU-R05		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

12. Requisitos de seguridad ante amenazas

RS-SA01: COPIAS DE SEGURIDAD			
La aplicación permite realizar copias de la programación interna de los módulos, de forma que si alguno se estropea por algún motivo, puede instalarse otro e introducirle la programación guardada. Deben realizarse copias de seguridad periódicas durante la instalación de un sistema domótico y una copia final al terminar ésta.			
Fuente:	RU-C14		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-SA02: PROTECCIÓN DE LAS COPIAS DE SEGURIDAD			
Las copias de seguridad se almacenarán en lugares separados de la instalación y protegidos adecuadamente, sólo al alcance de personal autorizado de la empresa.			
Fuente:	RU-C14		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

13. Requisitos de portabilidad

RS-P01: ENTORNO JAVA 6 Y RXTX			
La aplicación funcionará en cualquier máquina que tenga instalada el entorno J2SE versión 6 y cumpla los requisitos de recursos especificados. La única modificación a realizar podrá ser la sustitución de la librería de comunicaciones precompilada según el entorno de ejecución (por ejemplo, en entornos unix de 64 bits).			
Fuente:	RU-R02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

14. Requisitos de calidad

No aplicable.

15. Requisitos de fiabilidad

RS-FB01: FIABILIDAD DEL SISTEMA			
La fiabilidad del sistema queda sujeta a la fiabilidad de la infraestructura del sistema domótico y a la apropiada alimentación eléctrica. En condiciones de funcionamiento normales, la fiabilidad de los datos almacenados en la aplicación, así como la ejecución de las diferentes operaciones será del 100%.			
Fuente:	RU-R02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

16. Requisitos de mantenimiento

RS-M01: CORRECCIÓN DE ERRORES Y NUEVAS FUNCIONALIDADES			
La utilización de patrones de diseño y la documentación tanto interna (comentarios de código) como externa (javadoc) del código hacen posible la ampliación sencilla de las funcionalidades de la aplicación, así como la corrección de errores.			
Fuente:	RU-R05		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

17. Requisitos de seguridad ante daños físicos

RS-SF01: CONTROL DE DISPOSITIVOS INOPERATIVOS			
Deberá revisarse el estado de funcionamiento cualquier aparato o módulo de E/S físicos cuyo estado en su equivalente virtual sea <i>inoperativo</i> .			
Fuente:	RU-C01, RU-C02		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

RS-SF02: DESCONEXIÓN Y APAGADO DE APARATOS ANTE RIESGOS			
Para prevenir accidentes o daños en la infraestructura del sistema domótico se ofrecen dos funcionalidades importantes:			
<ul style="list-style-type: none"> ➤ Visualización del estado y cambio de estado de todos los aparatos del sistema. ➤ Posibilidad de apagar al instante desde la aplicación cualquier aparato de la instalación en caso de que se detecte un funcionamiento anómalo o unas condiciones de funcionamiento peligrosas proporciona una herramienta muy importante para. 			
Deberá apagarse cualquier aparato eléctrico que esté sometido a condiciones de humedad excesivas, y desconectarse del sistema domótico si su funcionamiento físico es erróneo.			
Fuente:	Desarrollo		
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Claridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja



6. MATRIZ DE TRAZABILIDAD

	RU-C01	RU-C02	RU-C03	RU-C04	RU-C05	RU-C06	RU-C07	RU-C08	RU-C09	RU-C10	RU-C11	RU-C12	RU-C13	RU-C14	RU-C15	RU-R01	RU-R02	RU-R03	RU-R04	RU-R05
RS-F01	✓	✓																		
RS-F02	✓																			
RS-F03	✓																			
RS-F04		✓																		
RS-F05		✓																		
RS-F06		✓																		
RS-F07	✓	✓																		
RS-F08			✓																	
RS-F09			✓																	
RS-F10				✓																
RS-F11					✓															
RS-F12						✓														
RS-F13						✓														
RS-F14							✓													
RS-F15								✓												
RS-F16									✓											
RS-F17										✓										
RS-F18											✓									



	RU-C01	RU-C02	RU-C03	RU-C04	RU-C05	RU-C06	RU-C07	RU-C08	RU-C09	RU-C10	RU-C11	RU-C12	RU-C13	RU-C14	RU-C15	RU-R01	RU-R02	RU-R03	RU-R04	RU-R05
RS-F19												✓								✓
RS-F20												✓								✓
RS-F21												✓								
RS-F22												✓								
RS-F23													✓							
RS-F24														✓						
RS-F25															✓					
RS-R01												✓								
RS-R02												✓								
RS-R03	✓	✓																		
RS-R04	✓	✓																		
RS-R05																		✓		
RS-I01																✓				
RS-I02																✓				
RS-I03														✓			✓			✓
RS-O01												✓								
RS-O02	✓																			
RS-O03	✓																			
RS-O04		✓																		



	RU-C01	RU-C02	RU-C03	RU-C04	RU-C05	RU-C06	RU-C07	RU-C08	RU-C09	RU-C10	RU-C11	RU-C12	RU-C13	RU-C14	RU-C15	RU-R01	RU-R02	RU-R03	RU-R04	RU-R05
RS-O05		✓																		
RS-O06		✓																		
RS-O07	✓	✓										✓								
RS-O08	✓	✓										✓								
RS-O09												✓								
RS-U01																	✓			
RS-V01																	✓			✓
RS-A01	✓	✓																		
RS-A02	✓	✓																		
RS-A03				✓																
RS-A04												✓								
RS-A05													✓							
RS-A06															✓					
RS-A07	Desarrollo																			
RS-D01																				✓
RS-D02																				✓
RS-SA01														✓						
RS-SA02														✓						
RS-PA01																	✓			



	RU-C01	RU-C02	RU-C03	RU-C04	RU-C05	RU-C06	RU-C07	RU-C08	RU-C09	RU-C10	RU-C11	RU-C12	RU-C13	RU-C14	RU-C15	RU-R01	RU-R02	RU-R03	RU-R04	RU-R05
RS-FB01																	✓			
RS-M01																				✓
RS-SF01	✓	✓																		
RS-SF02	Desarrollo																			

Tabla XVI. Matriz de Trazabilidad



7. DIAGRAMAS DE SECUENCIA

Para una comprensión del sistema más completa, puede resultar muy clarificadora la utilización de diagramas de secuencia que ilustren la dinámica de las interacciones entre las distintas partes del sistema.

A continuación se ofrecerán los diagramas de secuencia de las interacciones más importantes del usuario con el sistema. Los componentes implicados existen en el nivel conceptual y no en el de implementación, ya que lo que se busca es ofrecer una explicación clara y abstraída de las particularidades de implementación.

1. Diagrama de Secuencia de petición de estado por el usuario

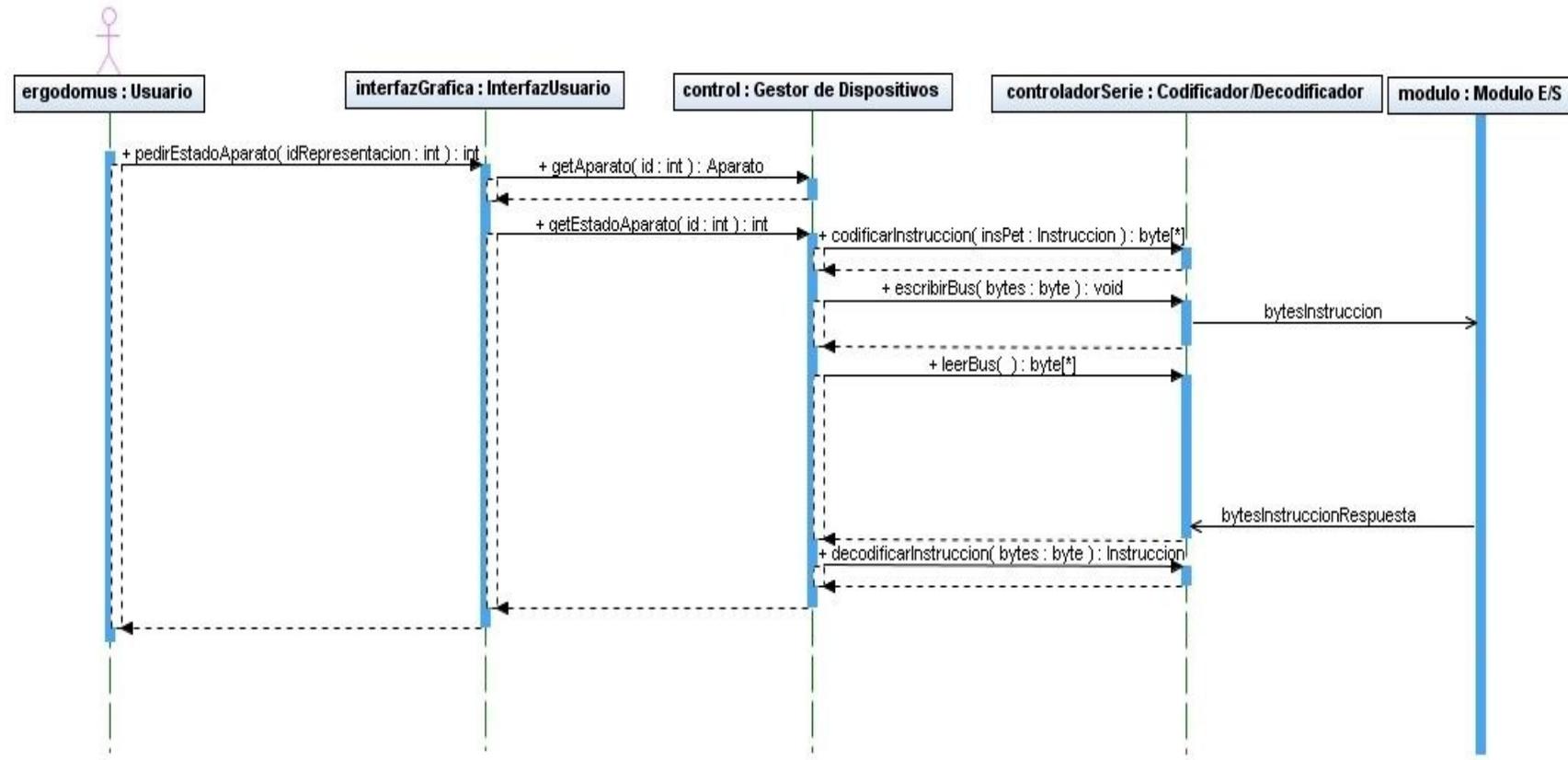


Ilustración 12: Diagrama de Secuencia: petición de estado por el usuario

2. Diagrama de Secuencia de la actualización del estado de los dispositivos del sistema

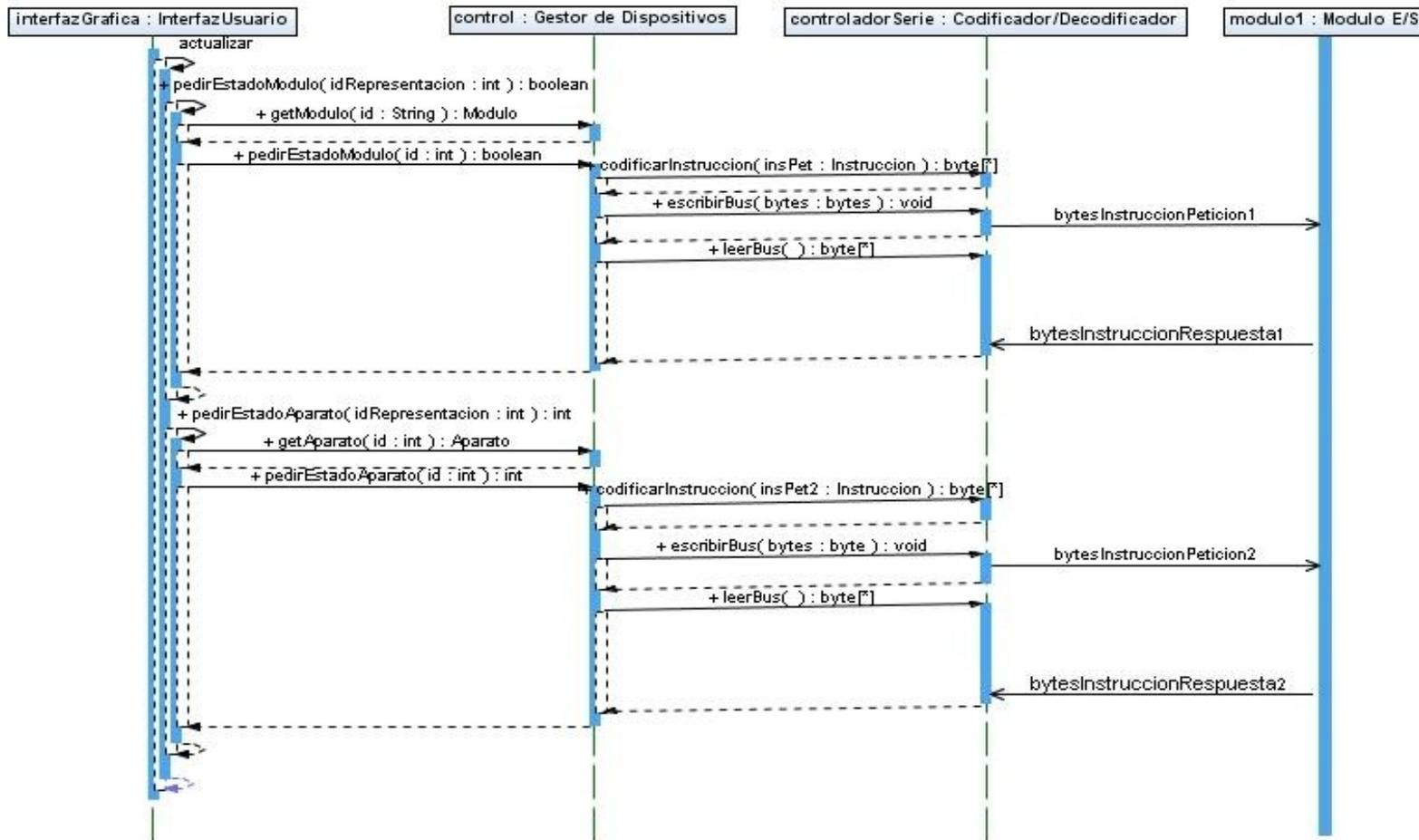


Ilustración 13: Diagrama de Secuencia: actualización del estado de los dispositivos del sistema

Capítulo 5

DESCRIPCIÓN DEL SISTEMA

1. INSTALACIÓN

La aplicación necesita para funcionar un entorno de ejecución java versión 6 como mínimo. Puede comprobarse la versión del entorno java instalado mediante el comando

```
java -version
```

En caso de necesitar instalar el entorno, éste puede descargarse desde la *web* de sun: <http://java.sun.com>

La aplicación domótica no necesita de ningún programa instalador, siendo sólo necesario copiar los archivos requeridos al disco duro con la jerarquía de directorios tal y como ha sido proporcionada.

La aplicación ha sido programada mediante el entorno Netbeans 6, por lo que presenta una serie de carpetas utilizadas por este.

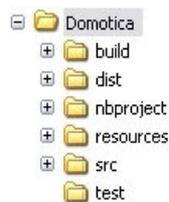


Ilustración 14: Estructura de directorios de Netbeans

Sin embargo, para hacer funcionar la aplicación sólo es necesaria una de las subcarpetas de la carpeta *build*, concretamente la carpeta **domotica**.



Ilustración 15: Estructura de directorios necesaria

Basta con copiar esta carpeta y todo su contenido al lugar desde el que se quiera ejecutar la aplicación, por ejemplo, el directorio raíz (en windows; en linux se recomienda situarla tras el directorio */home* del usuario) C:, y ejecutar desde allí:

```
C:\> java domotica.Main
```

2. CONFIGURACIÓN DE LA APLICACIÓN

La aplicación tiene una considerable cantidad de parámetros que pueden ser configurados mediante la edición de archivos de texto (en algunos casos la edición de un parámetro tendrá efecto en tiempo de ejecución; en otros casos será necesario reiniciar la aplicación) y que cambian el comportamiento de la misma de distintas maneras.

A partir de la carpeta **domotica**, pueden verse varias carpetas con los paquetes java de la aplicación (*control*, *controlador*, *instruccion*, *interfaz* y *util*). Sin embargo, otro directorio contiene todos los archivos de la aplicación que no son las clases java de la misma: el directorio de recursos *resources*.

En este directorio podemos encontrar distintos archivos (listados en cursiva) y subdirectorios, cada uno con una función determinada.

A continuación se muestra una captura de pantalla (Ilustración 16) del administrador de archivos mostrando el contenido de este directorio.

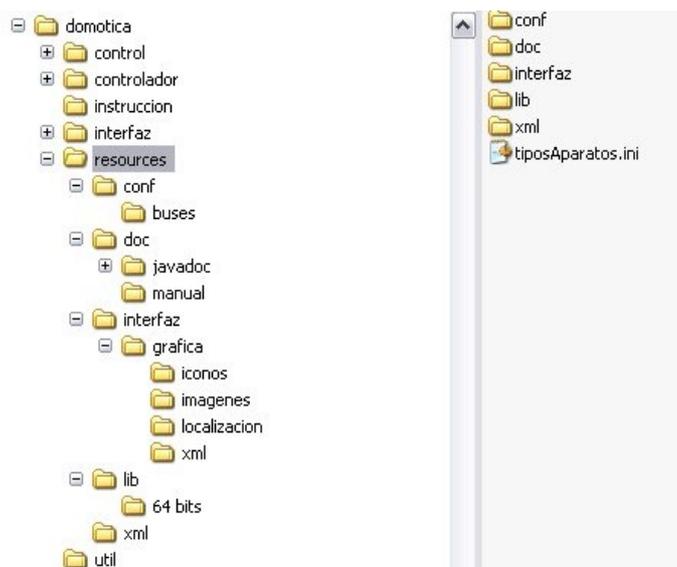


Ilustración 16: Estructura del directorio de recursos

- *tiposAparatos.ini*: este archivo es fundamental, ya que contiene todos los tipos de aparatos, y sus características, que soporta la aplicación. Se carga con el inicio de la aplicación y cada vez que se requiere la información de tipos de aparatos, y puede modificarse en tiempo real. Para más información ver la tabla XV: *Tipos de aparatos soportados por la aplicación*.
- **conf**: contiene distintos ajustes de bajo nivel que pueden realizarse en la aplicación. A su vez, en él se pueden encontrar:
 - *configuracion.properties*: archivo básico de configuración, con los parámetros más importantes de la aplicación.

configuracion.properties		
Parámetro	Valor por defecto	Descripción
nombreArchivoDatos	xml/ergodomus_moduloIO.xml	Localización del archivo XML que contiene toda la información de los dispositivos del sistema.
nombreEsquemaDatos	xml/ergodomus_schema.xsd	Localización del esquema de los datos.
nombreArchivoTipoAparatos	tiposAparatos.ini	Localización del archivo de tipos de aparatos.
tipoBus	serie	El bus que se usará en la comunicación con la infraestructura domótica. En principio sólo existe el serie, pero en un futuro podrá haber USB, paralelo, etc.
tipoInterfaz	grafica	La interfaz a usar. En principio sólo existe la gráfica, pero en un futuro podrá haber textual, web, etc.

configuracion.properties		
Parámetro	Valor por defecto	Descripción
idZonaAplicacion	1	Identificador de la zona que identificará a la aplicación en el sistema domótico. Puede modificarse en tiempo real.
idAplicacion	10	Identificador interno a la zona que identificará a la aplicación en el sistema domótico. Puede modificarse en tiempo real.

Tabla XVII. Archivo de configuración *configuracion.properties*

- **buses:** directorio con las configuraciones específicas de todas las interfaces de comunicación soportadas por las aplicación, cada una contenida en un archivo. En principio sólo estará el del bus serie, *puertoSerie.properties*.

puertoSerie.properties		
Parámetro	Valor por defecto	Descripción
nombreSerie	COM1	Nombre del puerto serie en el sistema. En windows, por defecto es <i>COM1</i> , en linux suele ser <i>/dev/tty0</i> .
velocidadSerie	115200	Velocidad a la que trabajará el puerto serie, en baudios. Hay que tener cuidado con que el puerto realmente soporte la velocidad especificada.
controlFlujoEntradaSerie	ninguno	El tipo de control de flujo aplicado a los datos entrantes. Puede ser sólo <i>ninguno</i> , <i>RTSCTS</i> o <i>XONXOFF</i> .
controlFlujoSalidaSerie	ninguno	El tipo de control de flujo aplicado a los datos salientes. Puede ser sólo <i>ninguno</i> , <i>RTSCTS</i> o <i>XONXOFF</i> .
bitsDatosSerie	8	Los bits de datos totales usados en la comunicación. Pueden ser 5, 6, 7 u 8.
BitsStopSerie	1	Los bits de parada usados en la comunicación. Pueden ser 1, 1.5 o 2.
ParidadSerie	ninguna	El sistema de paridad de datos usado en el bus serie. Puede ser <i>ninguna</i> , <i>even</i> , <i>mark</i> y <i>odd</i> .

Tabla XVIII. Archivo de configuración *puertoSerie.properties*

Mencionar que si en el arranque del sistema, se produce algún problema con la carga de una configuración del *puertoSerie.properties*, se procederá a intentar la carga de los valores por defecto del sistema. Sólo si éstos tampoco funcionan se descartará continuar con el arranque del sistema.

- **doc:** Directorio en el que se almacena toda la documentación asociada al sistema.
 - **javadoc:** directorio con la documentación de código generada por la herramienta de java *javadoc*, almacenada en formato html.
 - **manual:** directorio con el presente manual del usuario en formato pdf.
- **interfaz:** En este directorio podremos encontrar, separadas en subdirectorios, las configuraciones y archivos relativos a cada una de las interfaces de usuario de la aplicación. En un principio, dado que sólo tenemos la interfaz gráfica, sólo estará el subdirectorio *grafica*.
 - **grafica:** Contiene todos los archivos de configuración, imágenes y datos usados por la interfaz gráfica. Presenta varios directorios y un archivo de configuración básico, el *interfaz.properties*.
 - *interfaz.properties:* Archivo con la configuración básica de la interfaz gráfica.

interfaz.properties		
Parámetro	Valor por defecto	Descripción
frecuencia	3	Tiempo que tarda en actualizarse el estado de los dispositivos en la pantalla. Los cambios tienen lugar en tiempo real.
idioma	es	Lenguaje con el que se mostrarán los diálogos y toda la información del sistema al usuario. Puede ser <i>es</i> (español) o <i>en</i> (inglés). Para que el cambio tenga efecto totalmente es necesario reiniciar la aplicación.
tamanoIconos	2	Multiplicador del tamaño básico (32x32 pixels) con el que se mostrarán los iconos de los dispositivos al usuario. Puede ser 1, 2 o 3. Los cambios tienen efecto en tiempo real.
numEventosPanel	100	Número de eventos que podrá albergar el panel de historial de eventos antes de llenarse y empezar a eliminar los más antiguos. Los cambios tienen efecto en tiempo real.

Tabla XIX. Archivo de configuración *interfaz.properties*

- **iconos:** directorio que contiene los iconos usados en los diálogos, botones y menú de la interfaz gráfica.
- **imagenes:** directorio con las imágenes que representan a los aparatos y a los módulos en la interfaz gráfica. Cada tipo de aparato, según su estado

(operativo y activado: amarillo; desactivado: azul ; inoperativo: gris), tiene una imagen determinada.

La relación entre los tipos de aparatos, su estado y la imagen concreta a utilizar por la interfaz gráfica en su representación está almacenada en el archivo *imagenes.properties*, con el formato:

```
img_numeroTipoAparato_estadoAparato = ubicacionImagen
```

- **localizacion**: este directorio contiene los archivos de propiedades con los diferentes idiomas soportados por la interfaz gráfica. En principio sólo están disponibles el español y el inglés.
 - **xml**: directorio que contiene los archivos xml con la descripción de la interfaz gráfica según el formato de SwiXML. Hay un archivo xml por cada ventana a renderizar por ésta tecnología.
- **lib**: Este directorio contiene todas las librerías necesarias para el funcionamiento de la aplicación, desde la librería de SwiXML (*swixml.jar*) hasta la librería de comunicaciones rxtx (*RXTXcomm.jar*).

Es importante tener en cuenta que puede ser necesario tener versiones distintas de las librerías precompiladas (esto es, librerías no implementadas en java) para cada sistema operativo y para cada versión del mismo. Este es el caso de la librería base de rxtx, *rxtxSerial.dll* para windows y *librxtxSerial.so* para linux, además de tener otra versión alternativa de ésta última para las versiones de 64 bits.

- **xml**: Este directorio tal vez es el más importante de la aplicación, ya que contiene toda la información de los módulos de E/S y de los aparatos del sistema domótico. Dicha información está almacenada en un archivo en formato xml (por defecto, de nombre *ergodomus_xmlmoduloIO.xml*), que sigue un esquema en formato xsd (por defecto, *ergodomus_schema.xsd*).

3. EJECUCIÓN DE LA APLICACIÓN

Como se comentó en el primer apartado, para iniciar la aplicación basta con situarse en el directorio padre del directorio **domotica** y ejecutar:

```
java domotica.Main
```

Después de una serie de mensajes en la terminal indicando el progreso de la carga de la interfaz gráfica, se abre la ventana principal de la aplicación.

1. La ventana principal de la aplicación

Las funcionalidades de la interfaz gráfica giran en torno a la ventana principal, que en una primera ejecución tiene esta apariencia (Ilustración 17):

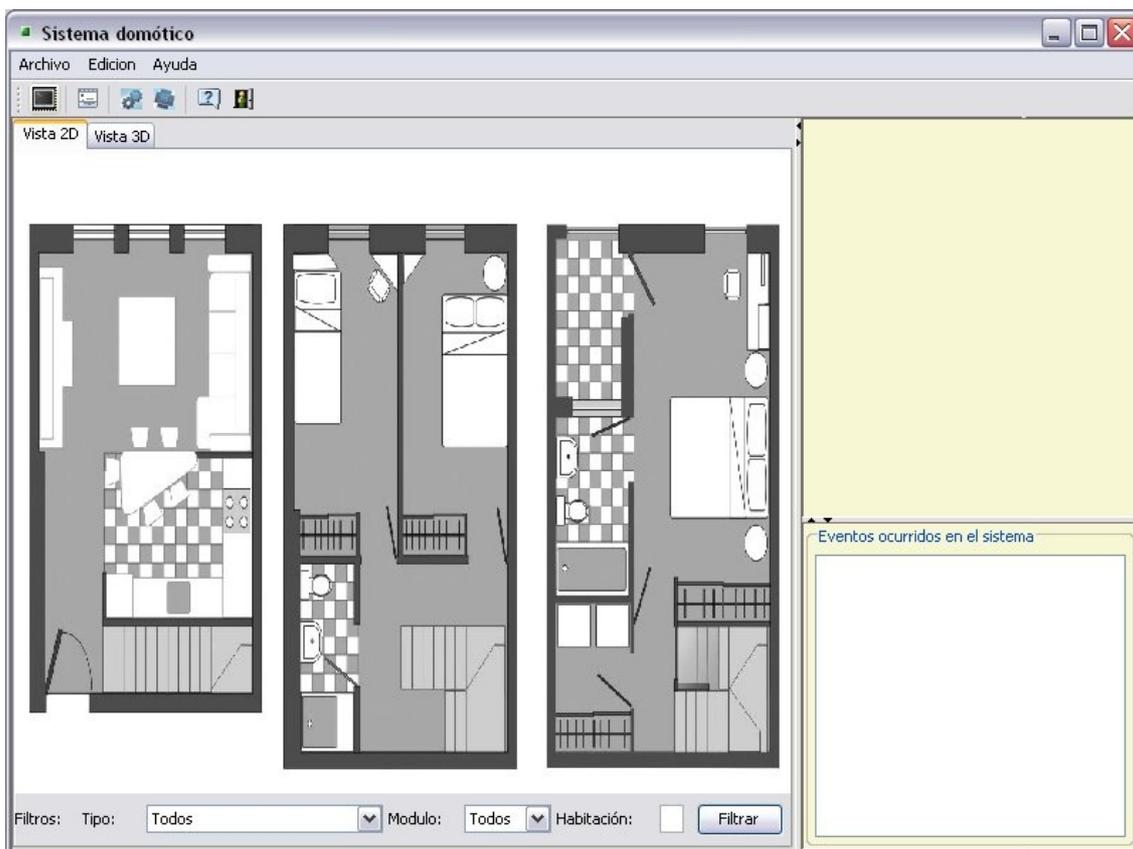


Ilustración 17: Ventana principal al iniciarse por primera vez

Esta ventana está dividida en 4 secciones separadas según su funcionalidad y objetivos:

1. Menú de herramientas y barra de botones
2. Mapa de dispositivos
3. Panel de estado
4. Panel de eventos

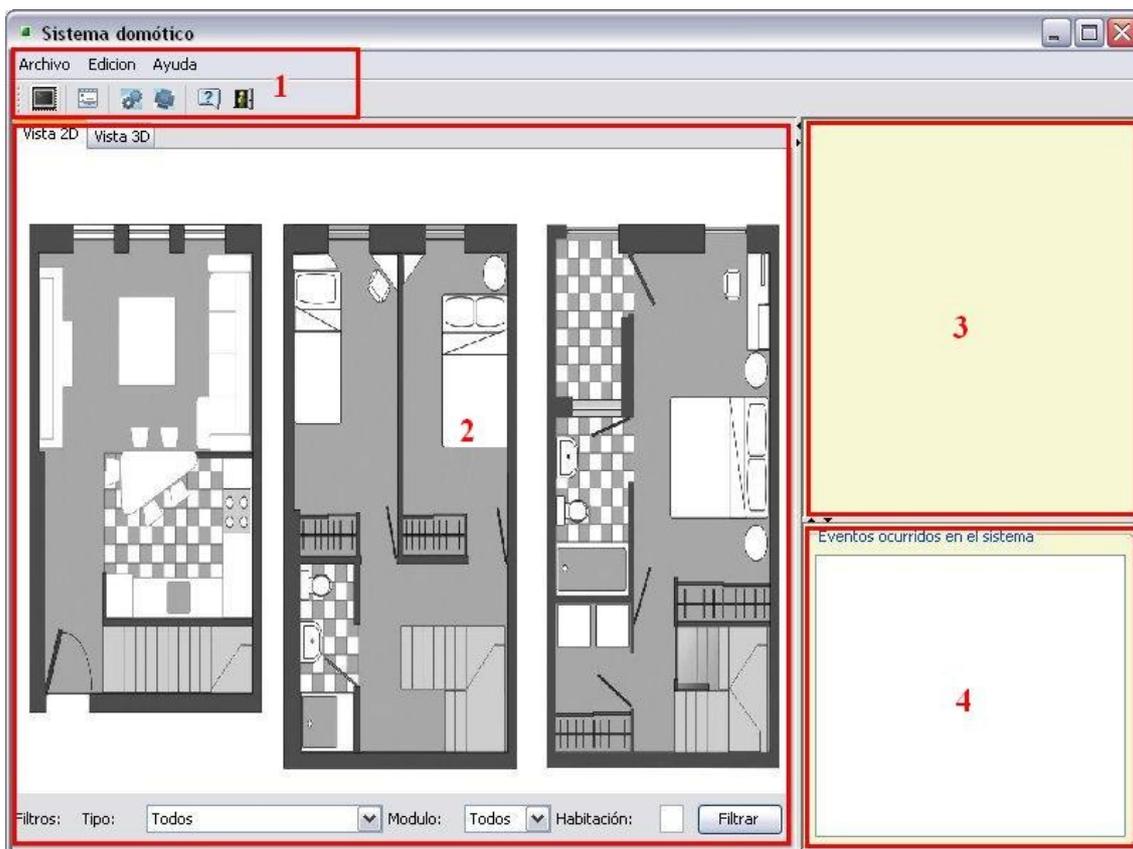


Ilustración 18: Las diferentes secciones de la ventana principal

Desde algunos de estos apartados se puede acceder a otras ventanas y funcionalidades importantes de la aplicación. De este modo, a continuación se describirán tanto las distintas secciones de la ventana principal como las *subventanas* de todas ellas.

1. Menú de herramientas y barra de botones

Representa el punto de entrada del usuario a las diferentes funcionalidades de la aplicación. Tanto el **menú de herramientas** como la **barra de botones** ofrecen acceso prácticamente a las mismas funcionalidades, con diferencias mínimas.

El **menú de herramientas** está dividido en tres submenús: *Archivo*, *Edición* y *Ayuda*.

El submenú *Archivo* ofrece las siguientes posibilidades:

- **Consola:** Lanza una instancia de la consola de usuario.
- **Opciones:** Presenta una ventana con las opciones principales de configuración de la interfaz gráfica.
- **Salir:** Cierra la aplicación, guardando toda la información de los dispositivos en el archivo de datos XML.



Ilustración 19: Menú de herramientas con el submenú 'Archivo' desplegado.

El submenú *Edición* ofrece las siguientes:

- **Añadir aparato:** Abre una ventana que posibilita la adición de un aparato a un módulo de E/S.
- **Añadir modulo:** Abre una ventana que posibilita la adición de un módulo de E/S a la interfaz gráfica.

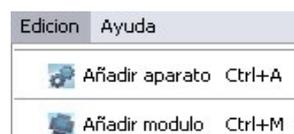


Ilustración 20: Menú de herramientas con el submenú 'Edición' desplegado.

Por último, el submenú Ayuda ofrece las opciones:

- **Ayuda:** Abre el manual de usuario.
- **Acerca de:** Abre una ventana con información general sobre la aplicación, destacando la versión de la misma. Esta es la única ventana cuyo único punto de acceso es el menú de herramientas, no pudiéndose acceder desde la barra de botones.



Ilustración 21: Ventana “Acerca de” el Sistema domótico.



Ilustración 22: Menú de herramientas con el submenú 'Ayuda' desplegado.

Por otra parte, la **barra de botones** ofrece el siguiente aspecto (Ilustración 23):



Ilustración 23: Barra de botones de la ventana principal

De izquierda a derecha, ofrece acceso a las siguientes funcionalidades (explicadas anteriormente para el menú de herramientas): consola de usuario, ventana de opciones, ventana de adición de aparatos, ventana de adición de módulos, manual de usuario y salida de la aplicación.

Todas estas características, junto con las restantes partes de la ventana principal, serán descritas ampliamente a continuación.

2. Consola de usuario

Esta consola permite la interacción de forma directa con la infraestructura del sistema domótico, esto es, la comunicación a nivel de bytes y sin pasar a través de los componentes gráficos de la interfaz. De esta manera se ofrece al usuario una forma alternativa de utilizar de la aplicación (de bajo nivel) que puede ser útil en numerosas circunstancias, en contraste con el manejo de la aplicación a través de imágenes y botones.

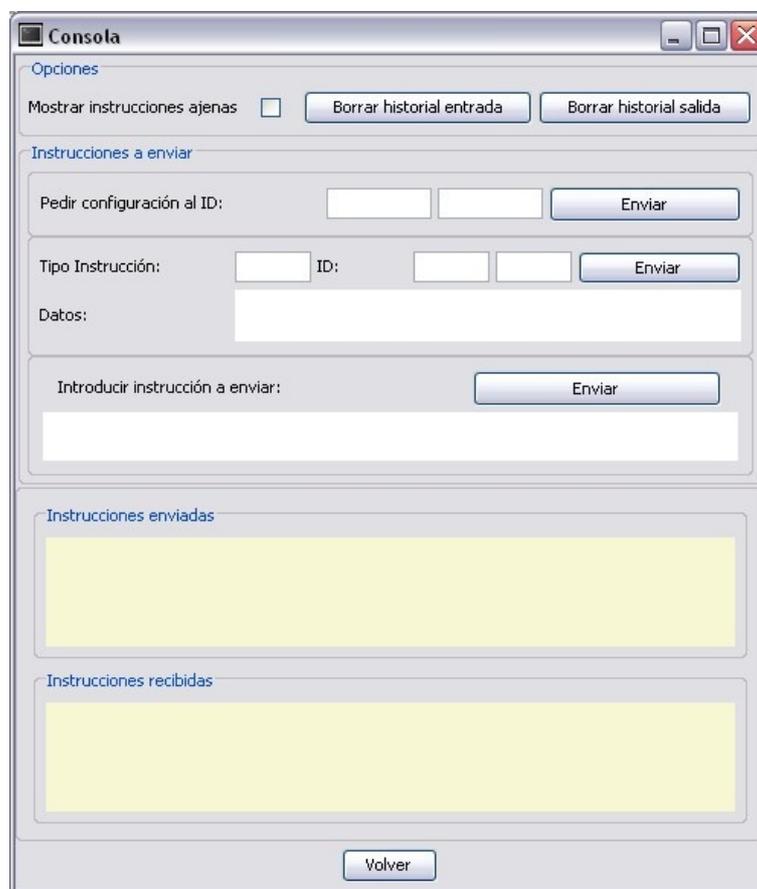


Ilustración 24: Consola de usuario

Esta consola de usuario está dividida en tres partes. De arriba a abajo, son las siguientes:

- **Opciones.** Permite modificar el comportamiento de la consola para una mayor comodidad en su uso. Las distintas posibilidades que soporta son:
 - *Mostrar instrucciones ajenas.* Cuando está seleccionada, en el historial de instrucciones recibidas se muestran las instrucciones recibidas que no estaban destinadas a la aplicación. Esto es útil cuando se quiere poner la aplicación en



modo promiscuo, esto es, que nos permita ver todos los intercambios de información que ocurren en la infraestructura domótica.

Si, por el contrario, lo que se desea es ver sólo la información que sale y entra de la aplicación, debe estar desactivada.

- *Borrar historial entrada*. Como su nombre indica, elimina la información del historial de entrada de la consola. Útil cuando está demasiado lleno y se quiere hacer un seguimiento en detalle de una nueva información por recibir.
- *Borrar historial salida*. Realiza la misma función que el botón anterior pero sobre el historial de salida de la consola.

➤ **Instrucciones a enviar**. Este bloque de operaciones permite al usuario enviar instrucciones a nivel de bytes con 3 distintos niveles de detalle. De arriba a abajo y de menor a mayor detalle, estos niveles son:

- Petición de configuración: envía una instrucción de petición de configuración al identificador introducido por el usuario. Ese es el único dato que el usuario tendrá que introducir en este nivel.
- Por tipo de instrucción: en este nivel, el usuario tendrá que especificar el tipo de instrucción (numérico) que desea enviar, el identificador de la localización al que quiere enviar la instrucción, y los datos adecuados al tipo de instrucción a enviar.

Para enviar una instrucción en este nivel, el usuario debe conocer en profundidad el protocolo de comunicación. La única facilidad que se le ofrece es el cálculo automático del CRC de la instrucción al ser enviada, y la generación del byte de control adecuado.

- A medida: En este último nivel de detalle, el usuario deberá introducir todos los bytes de la instrucción a enviar, incluyendo el CRC adecuado.

➤ **Historiales**. Esta sección, situada en la parte inferior de la ventana, ofrece dos registros distintos de actividad:

- Instrucciones enviadas: almacena y presenta las instrucciones que se han enviado desde la aplicación, ya sea desde la presente consola o desde cualquier otro componente de la interfaz gráfica. Junto con las instrucciones enviadas se indica además la hora en la que se realizó el envío.
- Instrucciones recibidas: funciona igual que el registro anterior, excepto porque almacena y presenta las instrucciones que han llegado hasta la aplicación (si está activado el *flag de mostrar instrucciones ajenas*) o que han sido enviadas exclusivamente hacia ésta (si el *flag* está desactivado).

3. Ventana de opciones

La ventana de opciones permite modificar características importantes del comportamiento de la interfaz gráfica. Su aspecto es el siguiente (Ilustración 25):



Ilustración 25: Ventana de opciones

Actualmente, ofrece 4 opciones:

- **Idioma.** Permite seleccionar el idioma con el que se desea que se presenten todos los mensajes, botones y menús de la interfaz gráfica. Para que los cambios surjan efecto, es necesario reiniciar la aplicación.
- **Frecuencia de actualización.** Permite modificar el tiempo que tarda la aplicación en muestrear el estado de los aparatos y módulos de E/S que tenga registrados, adecuando su estado visual al real. Si se especifica 0, la interfaz gráfica no actualiza los estados de sus dispositivos a no ser que se seleccione alguno en concreto, cuyo estado sí sería actualizado.
- **Tamaño de los iconos.** Permite modificar el tamaño con el que se presentan las imágenes de los aparatos y de los módulos de E/S. Un tamaño de 32x32 puede ser útil cuando hay muchos dispositivos en pantalla y se superponen unos a otros. El habitual de trabajo debería ser de 64x64, mientras que 128x128 sólo debería ser utilizado en momentos puntuales en los que se necesite ver claramente la imagen del aparato o módulo en cuestión, se tenga alguna deficiencia visual o haya poca visibilidad.
- **Número de eventos en panel.** Permite modificar el número de registros que podrá contener simultáneamente el panel de historial de eventos.

4. Ventana de adición de aparatos

La ventana de adición de aparatos presenta el siguiente aspecto (Ilustración 26):

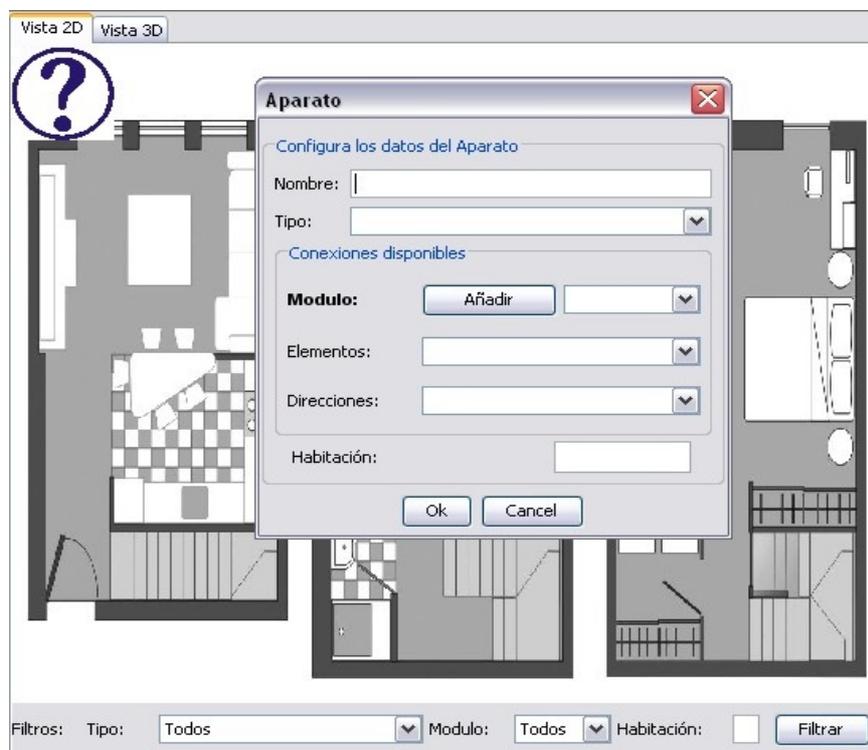


Ilustración 26: Ventana de adición de aparatos

Mientras la ventana permanezca abierta, en la ventana principal de la aplicación se mostrará un icono de interrogación indicando que todavía no se conoce el tipo de aparato que se está intentando añadir.

Para añadir un aparato a la aplicación, deben realizarse los siguientes pasos:

1. Elección del módulo al que estará conectado el aparato.

Dentro de la sección “Conexiones disponibles”, en la pestaña *módulo* se elegirá el identificador del módulo de E/S al que está conectado físicamente el aparato.

Si todavía no se añadió el módulo de E/S en cuestión, puede hacerse en este mismo momento pulsando sobre el botón *añadir* (el proceso de adición de un nuevo módulo está descrito en el apartado siguiente). El nuevo módulo podrá ser elegido entonces en el desplegable.

2. Selección del tipo de dirección a usar.

En el desplegable *elementos*, se seleccionará el tipo de dirección en el que estará conectado el aparato. El tipo de dirección puede corresponderse tanto con conexiones físicas (entradas analógicas, entradas digitales, salidas analógicas y salidas digitales) como lógicas (grupos y persianas). Las conexiones lógicas, por supuesto, se corresponden con una determinada conexión física en el módulo de E/S, pero su comportamiento es diferente al de cualquier aparato que se conecte de forma normal a esa conexión física.

A continuación puede verse una imagen (Ilustración 27) de la lista de tipos de conexiones desplegada para un determinado módulo seleccionado (en este caso, para el ejemplo se ha seleccionado el 10:160).



Ilustración 27: Ventana de adición de aparatos con los elementos disponibles de un módulo

3. Selección de la conexión exacta.

Una vez seleccionado el tipo de conexión (en el ejemplo, se selecciona “salida digital”), en el desplegable “direcciones” aparecerán todas las conexiones de ese tipo que hay disponibles para la conexión del aparato y en las que se especificará “libre”.



Ilustración 28: Ventana de adición de aparatos con las direcciones disponibles para conectarse

4. Tipos de aparatos.

Una vez seleccionada la conexión a utilizar por el aparato (en el ejemplo, se ha seleccionado la salida digital número 1 del módulo de E/S), se pueden completar el resto de campos en cualquier orden (Nombre del aparato, su tipo y habitación).

El desplegable de tipos de aparatos permite elegir de qué clase de aparato se trata; éste debería ser lo más parecido posible al aparato real que está conectado al módulo de E/S físico.

Hay que tener en cuenta que según el tipo de conexión seleccionado, los tipos de aparatos permitidos varían (por ejemplo, no se puede conectar una bombilla a una entrada, o un pulsador a una salida).



Ilustración 29: Ventana de adición de aparatos con los tipos de aparatos permitidos

5. Adición del aparato.

Tras haber rellenado todos los campos (en el ejemplo, se ha llamado al aparato “Bombilla”, de tipo “Luz binaria” y en la habitación 1), al pulsar *Ok* el nuevo aparato aparece en la ventana principal reflejando el estado que tenga el aparato físico (en el ejemplo -Ilustración 30-, la bombilla de la salida digital número 1 del módulo 10:160 está apagada, por lo que en la ventana aparece con un azul oscuro) y su tipo (en el ejemplo, la imagen que representa al aparato añadido es una bombilla).

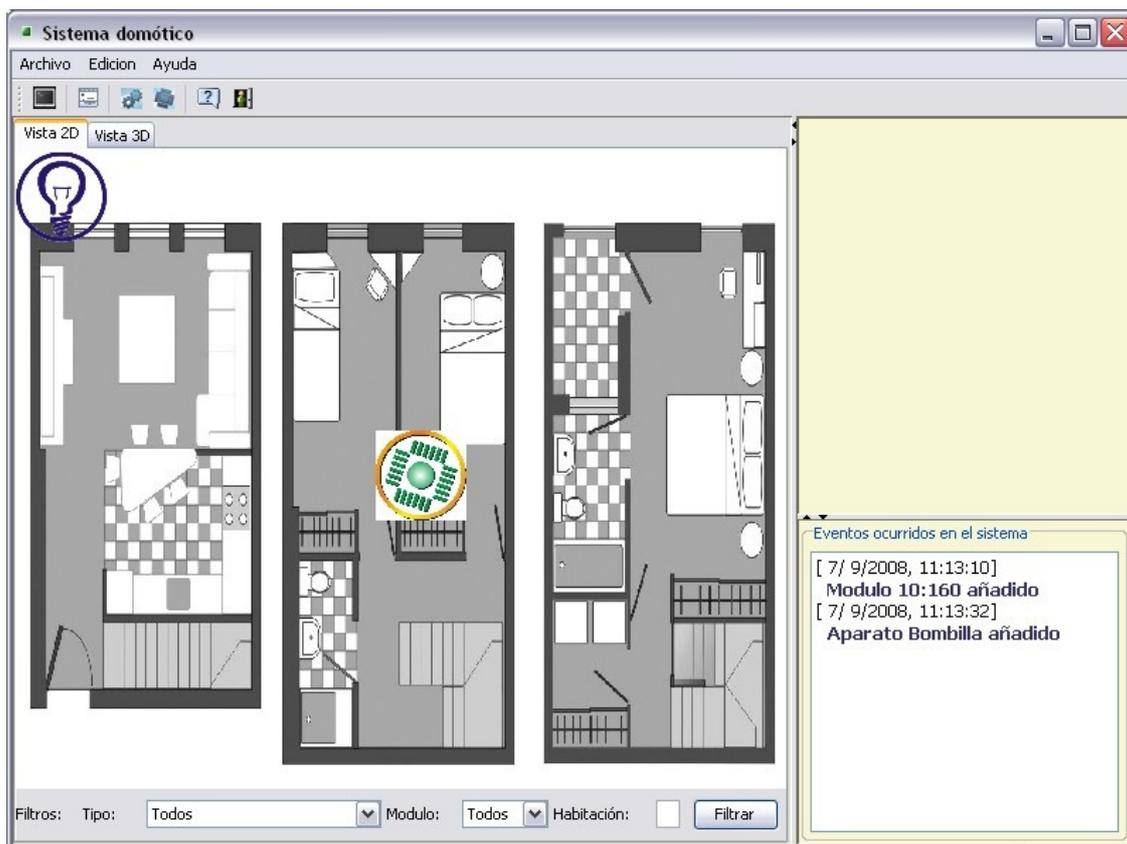


Ilustración 30: Ventana principal con un módulo y un aparato (bombilla) añadido

6. A partir de este momento, la actividad del aparato físico podrá controlarse y manipularse a través del aparato *virtual* recién añadido y del módulo de E/S *virtual* al que éste está conectado, además de poder conocer su estado y características en tiempo real.

5. Ventana de adición de módulos

Esta ventana puede alcanzarse tanto desde el menú de herramientas (submenú *edición*) y la barra de botones, como desde la ventana de adición de aparatos (explicada en el apartado anterior), en el apartado Módulo: *añadir*.

La ventana tiene el siguiente aspecto (Ilustración 31):



Ilustración 31: Ventana de adición de módulos

Cuando se desea añadir un nuevo módulo de E/S a la aplicación, las únicas variables que se pueden introducir son su identificador completo, su posición en la interfaz gráfica y la habitación donde se encuentra.

El identificador es un valor clave, ya que debe corresponderse exáctamente con el del módulo de E/S físico al que se desea que el módulo de E/S virtual represente. Si no coincide, no se podrá comunicar con él.

Los identificadores de los módulos de E/S son números enteros positivos. La primera parte del identificador puede ser cualquier número entero, mientras que la segunda debe ser múltiplo de 32. Así, serían valores válidos 3:96, 5:128 o 10:224. En caso de introducir un valor incorrecto de identificador, la ventana mostrará un mensaje de error y no permitirá añadir el módulo hasta que se haya introducido un valor correcto. El resto de parámetros no son obligatorios.

6. El Mapa de Dispositivos

Es la parte principal de la aplicación, ya que proporciona acceso a todos los dispositivos introducidos en la aplicación. Se divide en tres zonas bien diferenciadas (Ilustración 32):



Ilustración 32: Diferentes partes del mapa de dispositivos

1. Pestañas de vistas.

La aplicación está preparada para ser ampliada en un futuro de modo que pueda reflejar la infraestructura del sistema domótico de múltiples formas.

Actualmente permite una interacción con él en forma de vista en dos dimensiones del plano de la casa en cuestión y de iconos representativos de los aparatos y de los módulos de E/S. Sin embargo, ya está previsto añadir soporte a vistas en 3 dimensiones de la casa y sus dispositivos

A través de las pestañas de vistas se podrá cambiar en cualquier momento entre las distintas vistas del sistema, si bien en este momento la vista por defecto (*Vista 2D*) es la única disponible.

2. Contenedor de dispositivos.

Constituye la parte central y principal del Mapa de Dispositivos. Aquí se presentan las imágenes de los aparatos y los módulos de E/S *virtuales* a través de los que controlar sus equivalentes físicos, y desde aquí se pueden modificar sus propiedades y acceder a las herramientas que permiten modificar sus estados (ver el apartado 8, *Panel de Estado*).



Ilustración 33: Mapa de Dispositivos con varios módulos y aparatos cargados

Cada imagen de un dispositivo, presenta dos propiedades principales:

- El **tipo de dispositivo** representado: cada tipo de aparato (además de los módulos de E/S, que tienen su imagen propia) se representará aquí con una imagen distintiva e intuitiva que permita a primera vista hacerse una idea de la disposición de los aparatos en la infraestructura domótica.
- El **estado del dispositivo** representado: cada aparato puede tener 3 estados distintos (los módulos sólo pueden tener 2), cada uno de ellos representado con un color diferente:
 - Inoperativo: se representa con el color gris. Un aparato virtual en este estado puede indicar que el aparato físico no se está comunicando correctamente con la aplicación, que está funcionando mal en algún

sentido o que simplemente no existe un aparato en la conexión que se ha especificado.

- Apagado: se representa con color azul oscuro. Representa un aparato que se encuentra en estado de reposo (por ejemplo, una bombilla apagada o un sensor que no esté detectando nada).

Los módulos de E/S y algunos tipos de aparatos nunca presentarán este estado (por ejemplo, un termómetro nunca está apagado).

- Encendido: se representa con el color amarillo brillante. Un aparato en este estado representa que su equivalente físico está en funcionamiento (por ejemplo, una bombilla encendida, un aire acondicionado ventilando o una sirena acústica sonando).

Este sistema de tipos y colores permite de un vistazo hacerse una idea de las características más interesantes de cada dispositivo del sistema y del estado del sistema domótico en general.

Se puede conseguir una visualización textual un poco más detallada de la información de un dispositivo a través de los *tooltips* de los mismos.



Ilustración 34: Mapa de Dispositivos mostrando un 'tooltip' con información de un aparato

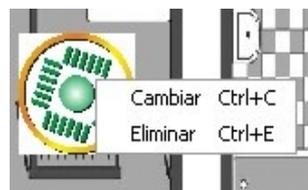
Estos *tooltips* pueden visualizarse situando el puntero del ratón sobre un aparato o módulo de E/S. Siempre presentarán la misma estructura visual, ya se trate de un aparato o de un módulo:

- **Nombre del aparato/identificador completo del módulo de E/S**
- Tipo textual (y *numérico*, si es un aparato) del dispositivo
- Estado del dispositivo

Como puede apreciarse en la imagen del ejemplo anterior, algunos tipos de aparatos no sólo informarán de que están encendidos, sino **en qué grado lo están**. Este puede ser el caso de una bombilla conectada a una salida analógica, un sensor de luz bajo cierta luminosidad o de un termómetro.

Sin embargo, la información de los dispositivos de la que se ha hablado hasta ahora resulta práctica pero no es totalmente completa. Tanto para obtener una información completa de la información de los dispositivos, como para poder modificar su estado en caso de tratarse de un aparato, se debe acceder al **Panel de Estado** correspondiente del dispositivo. Esto se consigue pulsando y soltando con el botón izquierdo del ratón sobre el dispositivo que se desee. Para más información se recomienda la sección “Panel de Estado”.

La manipulación de las imágenes de los dispositivos también permite controlar otras características suyas. Al pulsar con el botón derecho del ratón sobre la imagen de un dispositivo cualquiera (ya sea un aparato o un módulo de E/S) aparecerá un **menú contextual** como el siguiente (Ilustración 35):



*Ilustración 35: Menú contextual
de un dispositivo*

Este menú permite realizar dos acciones distintas:

- Modificar las características del dispositivo.
- Eliminar el dispositivo.

Si se desea modificar las características del dispositivo, la ventana emergente dependerá de si el dispositivo era un aparato o un módulo de E/S.

Si se trataba de un aparato, la ventana resultante será idéntica a la de adición de un nuevo aparato, con la diferencia de que los campos contendrán la información del aparato en cuestión.

A través de esta ventana el usuario podrá cambiar tanto el nombre del aparato como su tipo o el lugar al que está conectado, actualizándose la imagen en la interfaz según sus nuevas características.

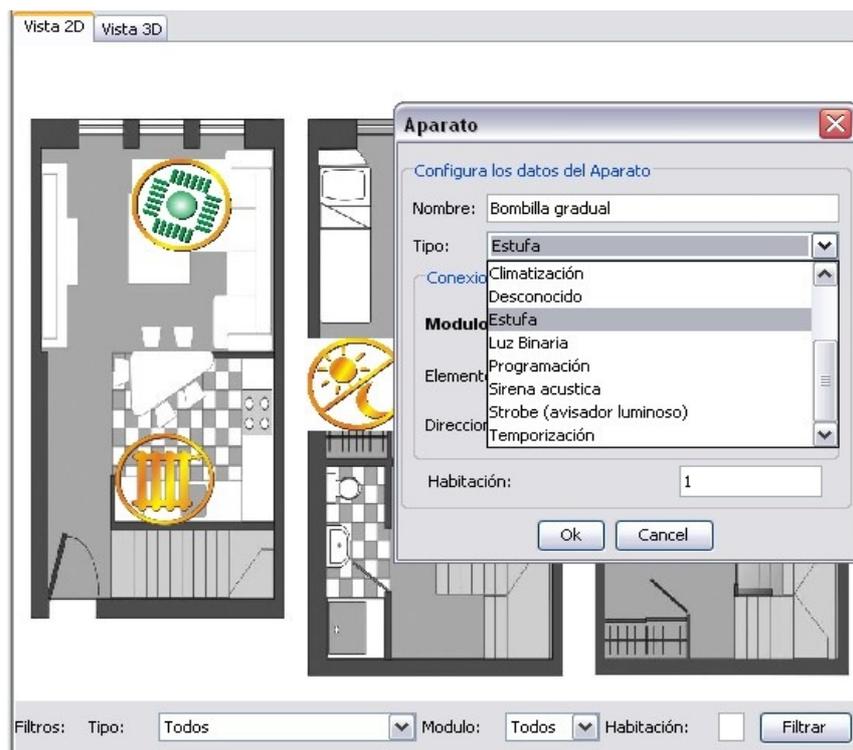


Ilustración 36: Cambio del tipo de un aparato

Si por el contrario se deseaba modificar las características de un módulo de E/S, la ventana emergente permite modificar numerosos aspectos de la configuración interna del módulo, por lo que serán explicados más en detalle en el apartado 7 *Ventana de modificación de módulos*.

3. Filtrado de dispositivos.

Situado en la parte inferior del Mapa, permite visualizar y/o ocultar dispositivos en la interfaz según diferentes criterios de selección. Esto es muy útil cuando el número de dispositivos en pantalla es grande y queremos centrarnos en los dispositivos que cumplan una determinada condición.

Hay que hacer incapié en que se trata exclusivamente de mostrar y/o ocultar dispositivos en el contenedor de dispositivos, y que la información de éstos sigue permaneciendo intacta en el sistema.

Los filtros de visualización que se pueden aplicar son:

- **Por tipo de aparato:** por defecto se muestran todos los aparatos sea cual sea su tipo. Sin embargo, puede seleccionarse cualquiera de los tipos de aparatos existentes en el sistema, en cuyo caso al pulsar sobre *filtrar* sólo se mostrarán los aparatos del tipo seleccionado.

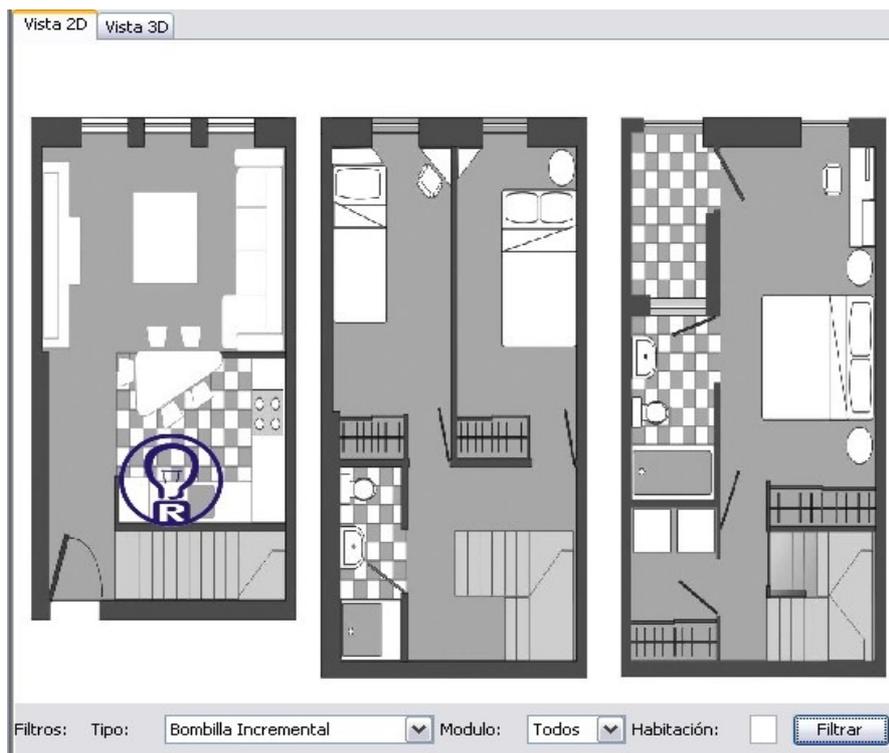


Ilustración 37: Mapa de Dispositivos con un filtrado por tipo de aparato

- **Por módulo de E/S:** Este filtro permite seleccionar para su visualización sólo los aparatos que estén conectados a un módulo de E/S concreto del sistema, junto con el propio módulo.
Por defecto se visualizan todos los aparatos y módulos del sistema.



Ilustración 38: Mapa de Dispositivos con un filtrado por módulo de conexión

- **Por habitación:** el último filtro de visualización consiste en mostrar y/o ocultar los dispositivos en función de la habitación que ocupen.



Ilustración 39: Mapa de Dispositivos con un filtrado por habitación

Hay que destacar que es posible el uso de varios filtros a la vez, de modo que se posibilita la creación de condiciones complejas de visualización.

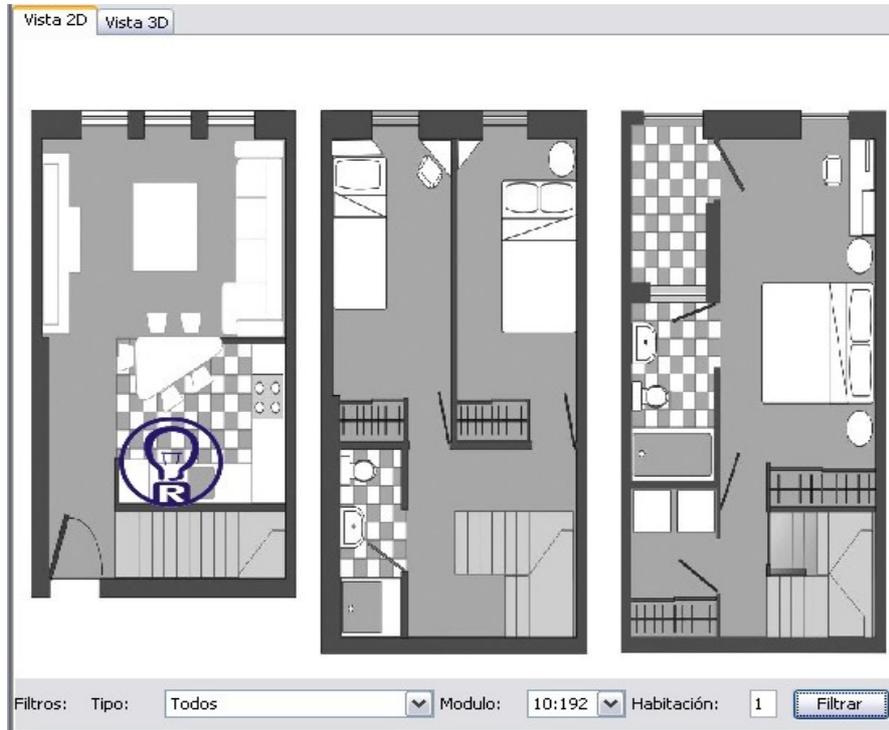


Ilustración 40: Mapa de Dispositivos con dos filtrados a la vez: por módulo y por habitación

7. Ventana de modificación de módulos

A esta ventana se puede acceder a través de la opción *cambiar* del menú contextual del módulo cuyas características se deseen modificar.

Como ejemplo, la ventana de modificación de módulos para un módulo 10:192 sería la siguiente (Ilustración 41):



Ilustración 41: Ventana de modificación de módulos de E/S

En la parte central y al igual que la ventana de adición de módulos, esta ventana permite modificar el identificador del módulo de E/S, además de su posición y la habitación que ocupa.

Hay que tener en cuenta algo muy importante: si sólo se modifica el identificador del módulo y se pulsa en “Ok”, el módulo de E/S *virtual* pasará a tener un identificador distinto al del módulo *físico* de E/S que representaba antes, lo que puede tener dos consecuencias:

1. El módulo de E/S *virtual* no se corresponde con ningún módulo *físico* de E/S existente en la infraestructura domótica, con lo que el estado del módulo *virtual* pasará a ser inoperativo, al igual que todos los aparatos que dependan de él.
2. El módulo de E/S *virtual* pasa a corresponderse con otro módulo *físico* de E/S, con lo que su estado seguirá siendo operativo pero las configuraciones internas de ambos y los aparatos que tienen conectados pueden ser totalmente distintos, por lo que es probable que en la representación *virtual* de los aparatos conectados haya un malfuncionamiento generalizado.

Ahora es el momento de hablar, por tanto, de las **reprogramaciones** y de las **actualizaciones de configuración**.

Para evitar este problema que se ha señalado anteriormente (entre otras funcionalidades que se señalarán a continuación), se ofrece la posibilidad de modificar no sólo la configuración de la aplicación sino también la del propio módulo de E/S físico.

En este caso de ejemplo, se permite extender la modificación del identificador del módulo *virtual* 10:192 al módulo *físico* 10:192, de modo que ambos pasen a tener el identificador 10:224. Para hacer esto hay que realizar dos pasos:

1. Se pulsa en el botón “Reprogramar ID” que está a la derecha del identificador: este primer paso modifica el identificador programado en el módulo *físico* de E/S, que pasa a ser el 10:224. En este paso el módulo *virtual* de E/S sigue siendo el 10:192, de modo que no estaría totalmente actualizado.



Ilustración 42: Reprogramación del identificador de un módulo

2. Se pulsa en el botón “Ok”. En este paso el nuevo identificador se guarda también en la información interna de la aplicación, con lo que a partir de este momento se podrá manejar el módulo de E/S igual que se manejaba en un principio pero bajo el identificador 10:224.

La otra (realmente, la principal) gran funcionalidad que ofrece este tipo de operaciones va mucho más allá de modificar el identificador de un módulo de E/S: **puede modificarse cualquier aspecto de la configuración interna del módulo, así como también obtener esta configuración** para volcarla en la información interna del sistema sobre su módulo de E/S *virtual* equivalente. Esto en la práctica permite almacenar en el archivo de datos XML la configuración interna de toda la infraestructura del sistema domótico que esté reflejada en su equivalente *virtual*.

Estas operaciones de actualización y reprogramación de toda la configuración de un módulo de E/S están disponibles en la parte inferior derecha de la ventana, en los botones “Actualizar Todo” y “Reprogramar Todo”.



Ilustración 43: Opciones de actualización y reprogramación masiva de un módulo

No hay que olvidar que tras una operación de actualización total de un módulo, para que los nuevos datos extraídos del módulo físico de E/S se guarden se debe pulsar en “Ok”. Pulsando en “Cancelar” todos los datos nuevos serán descartados.

La ventana de modificación de módulos permite, como ya se ha dicho, modificar cualquier aspecto de la configuración interna del módulo de E/S. Estas configuraciones están agrupadas en 4 grupos, a los que puede accederse a través de otros tantos botones: entradas, salidas, grupos y persianas.



Ilustración 44: Configuración detallada de un módulo

Entradas

Esta nueva ventana presenta dos pestañas separadas, una para las entradas analógicas (que aparece en primer plano) y otra para las entradas digitales.

Para las entradas analógicas, se permite elegir a través de un desplegable entre las 3 entradas analógicas disponibles en los módulos actuales de E/S.

Una vez seleccionada una de las entradas, se carga la configuración que exista de ella en el sistema. En la imagen del ejemplo puede verse que en la segunda entrada analógica aparece el sensor de luz que podía verse en los ejemplos anteriores.

De hecho, esta es una forma alternativa de modificar las propiedades de un aparato e, incluso, de eliminarlo, ya que si en el campo del nombre del dispositivo se introduce un nombre no válido o se elimina el nombre por completo y se guardan todos los cambios, el aparato se eliminará de la aplicación.



Ilustración 45: Configuración de las entradas analógicas de un módulo

Es de destacar que las entradas analógicas no contienen ninguna configuración especial que pueda ser reprogramada o actualizada a través del módulo físico de E/S.

Este no es el caso de las entradas digitales, que contienen diferencias en su programación interna de modo que un mismo aparato conectado a una entrada digital puede comportarse de forma distinta que si se conecta a otra.

Estas diferencias radican en el modo en que se interpretan las pulsaciones en la entrada digital (como si fueran un interruptor, un pulsador, un pulsador incremental o uno de persiana) y las acciones a realizar ante ellas.

Puede apreciarse en la imagen del ejemplo (Ilustración 46) que a través de los botones situados a la derecha del número de entrada digital seleccionada, puede actualizarse la configuración exclusiva de esa entrada o por el contrario incorporar al módulo *físico* de E/S los cambios realizados en la interfaz.



Ilustración 46: Configuración de las entradas digitales de un módulo

Salidas

Al abrir la ventana correspondiente a las salidas del módulo, aparece una ventana con dos pestañas que permiten acceder a la información de la salida analógica (cada módulo sólo tiene una) o a la de las salidas digitales del módulo.

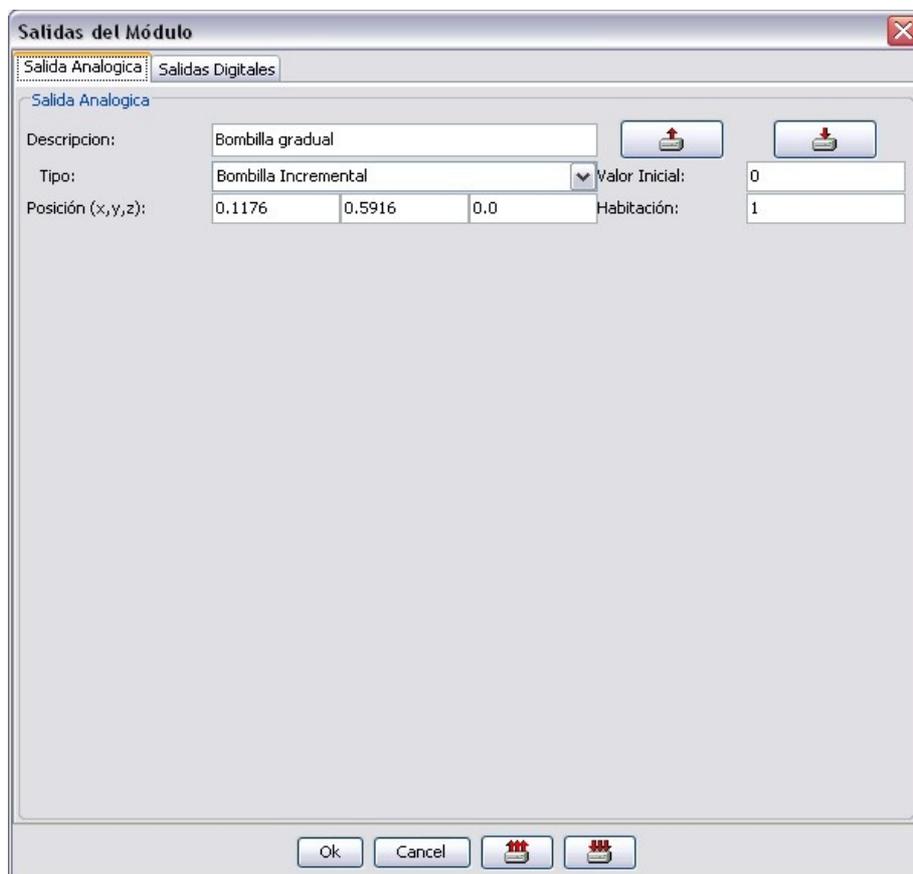


Ilustración 47: Configuración de la salida analógica de un módulo

La pestaña pulsada por defecto es la que contiene la información de la salida analógica. En la imagen del ejemplo (Ilustración 47) puede observarse cómo en la salida analógica del módulo seleccionado se encuentra la bombilla gradual, junto con su información asociada, que se ha visto en otras imágenes anteriores de la ventana principal.

El único valor configurable únicamente dependiente del módulo de E/S es el valor inicial con el que comienza la salida analógica.

La otra pestaña contiene la información de las seis salidas digitales que presenta un módulo de E/S, y constituye la ventana más compleja de la aplicación.

Ilustración 48: Configuración de la salidas digitales de un módulo

Aparte del nombre (*descripción*) del aparato que se pueda conectar a la salida (en la imagen de ejemplo, para la salida 1 no hay ningún nombre, con lo que se deduce que no hay conectada a esa salida ningún aparato), su posición y su tipo, los demás parámetros son enteramente parte de la configuración del módulo *físico* de E/S para esa salida digital. Constituyen un conjunto de condiciones para el encendido o apagado de la salida, calculadas en un primer paso mediante operaciones entre los elementos de dos grupos de elementos, para a continuación operar sobre el resultado de los dos conjuntos de operaciones.

Grupos

Los grupos de un módulo de E/S permiten la ejecución de varias acciones (en los módulos actuales, hasta un máximo de 10) en distintas direcciones con el envío de una sola orden de ejecución a una dirección determinada.

Ilustración 49: Configuración de los grupos de un módulo

Los módulos actuales permiten un máximo de 5 grupos por módulo, y para cada acción de cada grupo puede especificarse el destino de la acción, la instrucción concreta que se enviará y con qué prioridad.

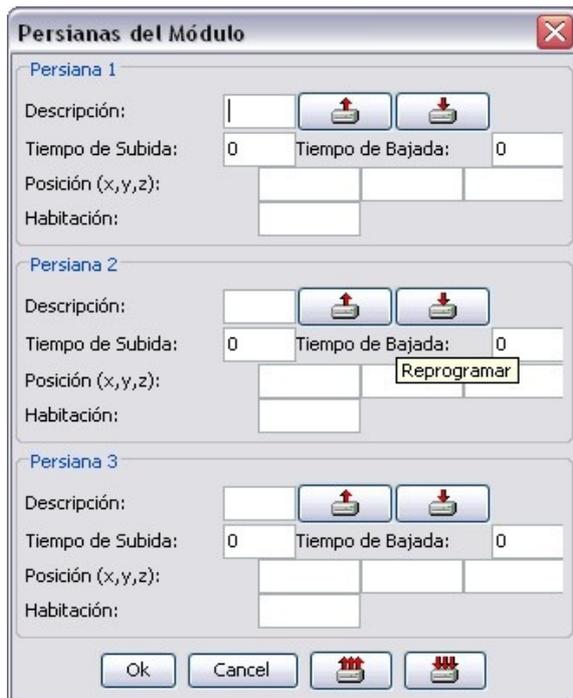
Con el objetivo de hacer más claro el propósito de cada acción, por cada una de ellas se permite realizar una descripción del mismo, que por supuesto no afectará en ningún momento a la configuración interna del módulo.

Persianas

Los módulos de E/S admiten hasta 3 direcciones con modo de operación tipo persiana.

Los valores configurables de una persiana son sólo dos, su tiempo de subida y su tiempo de bajada (en segundos).

A continuación (Ilustración 50) se muestra la ventana de configuración de las persianas de un módulo de E/S.



The screenshot shows a software window titled "Persianas del Módulo" with a close button in the top right corner. The window is divided into three sections, one for each blind (Persiana 1, Persiana 2, and Persiana 3). Each section contains the following fields and controls:

- Descripción:** A text input field followed by two buttons with upward and downward arrows.
- Tiempo de Subida:** A numeric input field with the value "0".
- Tiempo de Bajada:** A numeric input field with the value "0".
- Posición (x,y,z):** Three separate numeric input fields.
- Habitación:** A text input field.

At the bottom of the window, there are four buttons: "Ok", "Cancel", and two buttons with upward and downward arrows.

Ilustración 50: Configuración de las persianas de un módulo

8. Panel de Estado

El panel de estado de la ventana principal constituye una herramienta fundamental a la hora de manipular los dispositivos del sistema domótico. A través de él se visualiza toda la información principal de los aparatos y de los módulos de E/S, así como también se puede modificar su estado de forma sencilla.

En principio el panel de estado aparece vacío. Para ver la información asociada a un dispositivo sólo es necesario pulsar sobre él en el Mapa de Dispositivos, y su información aparecerá en el panel de estado.

En el ejemplo siguiente (Ilustración 51) se muestra la información y el estado de una bombilla conectada a la salida analógica del módulo de E/S 10:192:

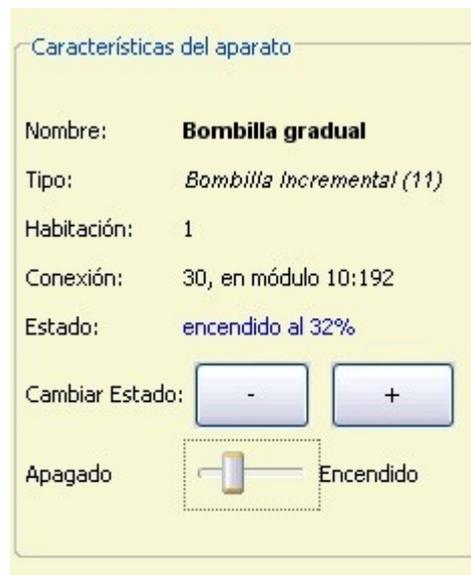


Ilustración 51: Panel con la información y opciones de cambio de estado de una bombilla conectada a una salida analógica

Es muy importante destacar que la información mostrada por el panel de estado variará en función del tipo de dispositivo de que se trate y a donde esté conectado.

Para cualquier aparato, se mostrará siempre su nombre, tipo de aparato, habitación ocupada, lugar de conexión y estado.

Si se trata de una salida, además se permitirá modificar su estado a través de unos botones de encender/apagar (si es una salida binaria) o aumentar/descender (si es gradual), y de una barra deslizante con la que se podrá ajustar exáctamente el nivel de encendido para los aparatos graduales.

Si la salida es digital, se podrá activar/desactivar además un parámetro extra, la comprobación del estado de su salida viva, que es una condición añadida al apagado/encendido de la salida digital correspondiente.

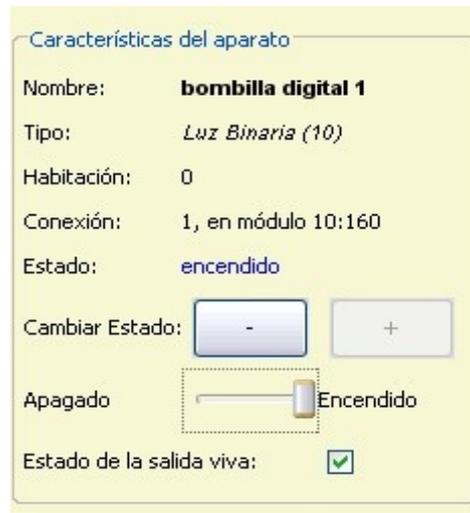


Ilustración 52: Panel con la información y opciones de cambio de estado de una bombilla conectada a una salida digital

Si el dispositivo *virtual* está emplazado en una conexión perteneciente al rango de conexiones de persianas, las funciones de cambio de estado son muy similares a las de la salida analógica, cambiando los conceptos de apagado/encendido por el de bajada/subida.

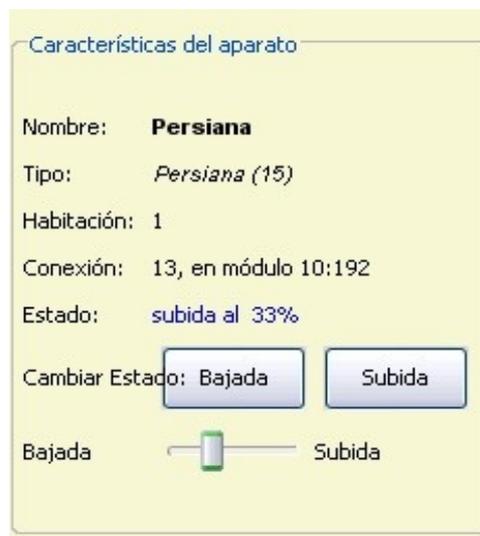


Ilustración 53: Panel con la información y opciones de cambio de estado de un aparato de tipo persiana

Si el dispositivo *virtual* está emplazado en una conexión perteneciente a un grupo de acciones, las funciones de cambio de estado anteriores desaparecen para dejar paso a las de ejecutar las acciones asociadas al grupo (*ejecutar*), o ejecutar justo las contrarias (*invertir*).

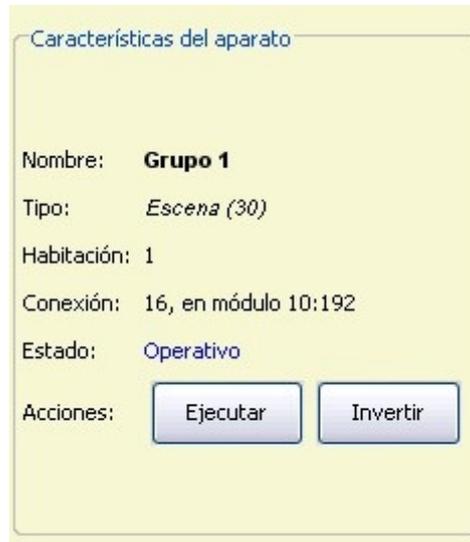


Ilustración 54: Panel con la información y opciones de cambio de estado de un grupo de acciones

Por último, si el aparato está conectado a una entrada, ya sea analógica o digital, el panel de estado sólo muestra su información, no mostrando lógicamente ninguna herramienta de cambio de estado.

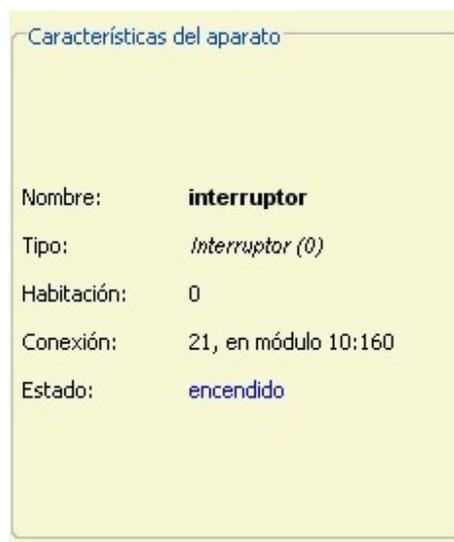


Ilustración 55: Panel con la información de un aparato de entrada de datos

9. Panel de historial de eventos

El panel de historial de eventos está situado en la parte inferior derecha de la ventana principal de la aplicación.

Este panel recoge toda la actividad principal realizada por el usuario sobre los dispositivos del sistema (adicción y eliminación de aparatos y módulos de E/S, cambios de estado en cualquiera de ellos), así como los propios cambios de estado de los aparatos debido a otros agentes externos.

Toda esta actividad aparece recogida junto con el momento exacto en el que ocurrió, de modo que un usuario que no haya estado pendiente de los sucesos ocurridos en el sistema domótico pueda hacerse una idea rápida de todo lo ocurrido en él (y el momento exacto en el que ocurrió) durante su ausencia.

El formato con el que se recoge la información es el siguiente:

[dd/MM/aaaa, hh:mm:ss]

nombre del dispositivo acción ocurrida

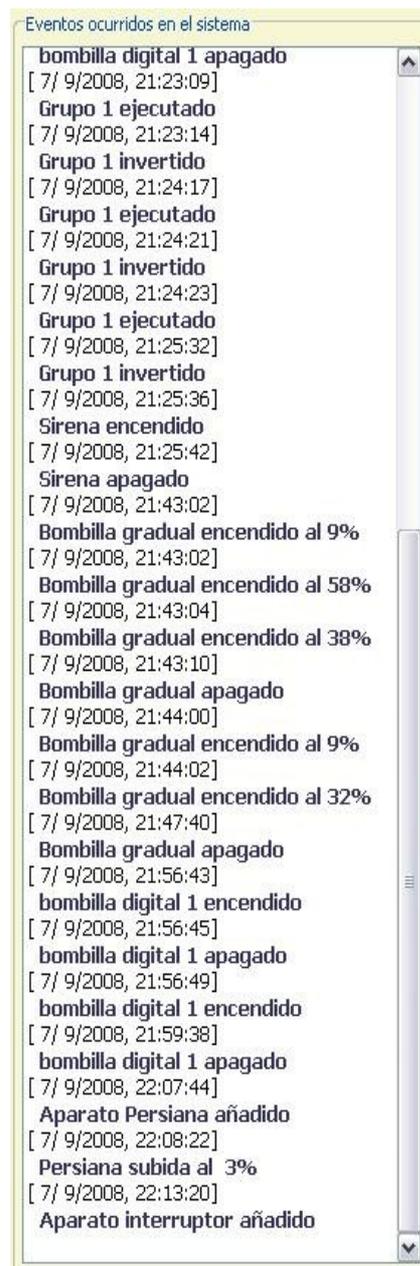


Ilustración 56: Ejemplo del estado del panel de eventos tras una actividad mediana

Hay que tener en cuenta que el número de eventos que puede almacenar el papel está limitado por el valor de la opción correspondiente de la ventana de opciones, que por defecto admite hasta 100 eventos. A partir del evento número 100, los primeros almacenados serán eliminados según lleguen nuevos eventos.

Esta cantidad es válida cuando se trata de un sistema pequeño y con una actualización estándar (3 segundos), pero puede quedarse muy pequeña si se cuenta con varios aparatos de entrada de información y su estado se refresca con esa o una mayor frecuencia.

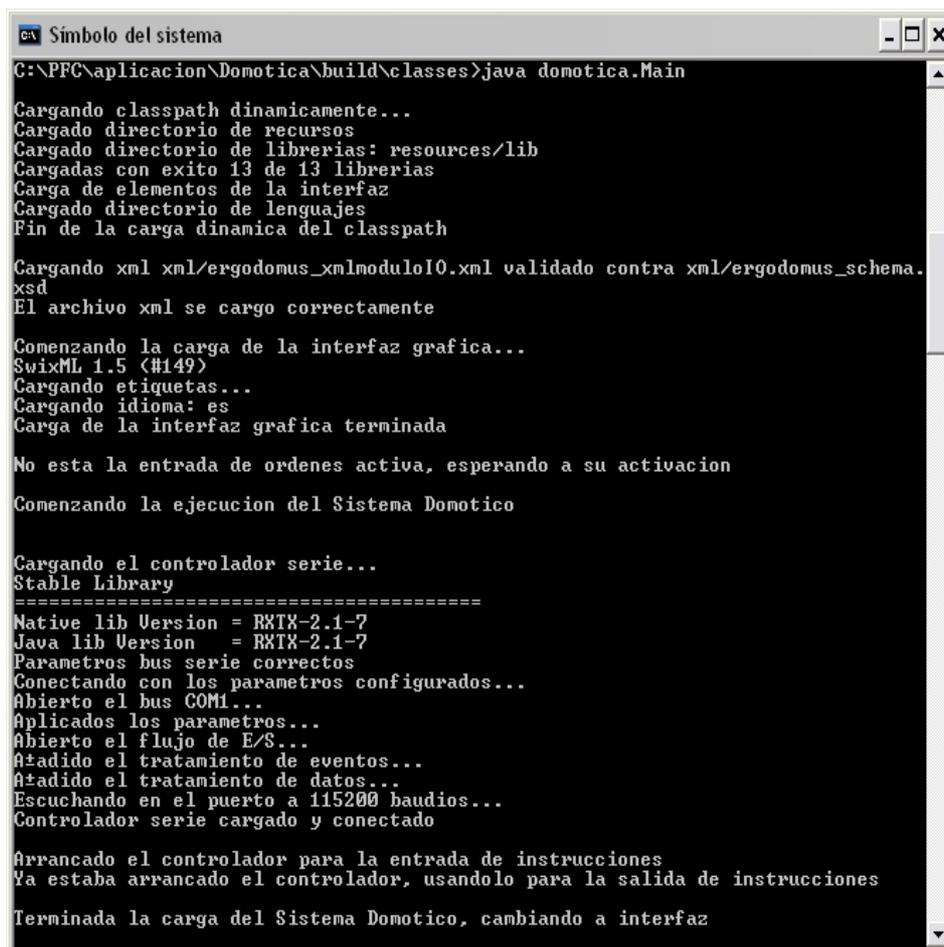
2. La terminal de comandos

Al principio de la sección 3 *Ejecución de la aplicación* se habló de la forma en que se ejecuta la aplicación domótica, a través de la terminal. De esta forma, la terminal permanece abierta durante toda la ejecución del programa, y su funcionalidad va mucho más allá que la del simple arranque de la aplicación.

La terminal ofrece información concisa de nivel técnico de lo que está ocurriendo en cada momento en la aplicación, en especial sobre comportamientos anómalos o erróneos que requieran una revisión por parte de la empresa.

1. El arranque de la aplicación

Tras introducir el comando de arranque del sistema, la aplicación muestra una serie de mensajes informativos indicando el proceso de carga de la aplicación.



```
Símbolo del sistema
C:\PFC\aplicacion\Domotica\build\classes>java domotica.Main

Cargando classpath dinamicamente...
Cargado directorio de recursos
Cargado directorio de librerias: resources/lib
Cargadas con exito 13 de 13 librerias
Carga de elementos de la interfaz
Cargado directorio de lenguajes
Fin de la carga dinamica del classpath

Cargando xml xml/ergodomus_xmlmoduloIO.xml validado contra xml/ergodomus_schema.xsd
El archivo xml se cargo correctamente

Comenzando la carga de la interfaz grafica...
SwiXML 1.5 (#149)
Cargando etiquetas...
Cargando idioma: es
Carga de la interfaz grafica terminada

No esta la entrada de ordenes activa, esperando a su activacion

Comenzando la ejecucion del Sistema Domotico

Cargando el controlador serie...
Stable Library
=====
Native lib Version = RXTX-2.1-7
Java lib Version = RXTX-2.1-7
Parametros bus serie correctos
Conectando con los parametros configurados...
Abierto el bus COM1...
Aplicados los parametros...
Abierto el flujo de E/S...
Añadido el tratamiento de eventos...
Añadido el tratamiento de datos...
Escuchando en el puerto a 115200 baudios...
Controlador serie cargado y conectado

Arrancado el controlador para la entrada de instrucciones
Ya estaba arrancado el controlador, usandolo para la salida de instrucciones
Terminada la carga del Sistema Domotico, cambiando a interfaz
```

Ilustración 57: Terminal mostrando el proceso de carga de la aplicación

En la ilustración (Ilustración 57) puede verse cómo en primer lugar se efectúa la carga de todas las librerías situadas en el directorio resources/lib. A continuación se tratan los datos de los dispositivos del sistema (de no existir el archivo xml o el xsd, se aborta la ejecución), se comienza la carga de la interfaz (en el caso de la interfaz gráfica, se tratan los archivos xml de descripción de sus elementos) y en último lugar se arranca el controlador de E/S, por defecto el controlador serie.

Sólo entonces se cede el control del sistema a la interfaz de usuario.

2. La E/S de la aplicación

A través de la terminal se puede observar en tiempo real toda la información que entra y sale de la aplicación hacia la infraestructura domótica.

```
Simbolo del sistema - java domotica.Main
<- 0 10 0 205 0 1 0 10 3 14 1 114 0 102 0 1 0 10 0 10 0 205 4 15 1 114 0 135 239 ->
<- 0 10 0 205 0 1 0 10 111 14 1 0 115 211 0 1 0 10 0 10 0 192 6 14 0 115 1 93 ->
<- 0 10 0 205 0 1 0 10 111 14 1 0 115 211 0 1 0 10 0 10 0 192 6 14 0 115 1 93 ->
<- 0 10 0 -51 0 1 0 10 3 14 1 117 0 105 ->
<- 0 10 0 205 0 1 0 10 3 14 1 117 0 105 0 1 0 10 0 10 0 205 4 15 1 117 0 117 224 ->
<- 0 10 0 -51 0 1 0 10 127 14 1 0 118 -26 ->
<- 0 10 0 205 0 1 0 10 127 14 1 0 118 230 0 1 0 10 0 10 0 192 6 14 0 118 1 96 ->
<- 0 10 0 -51 0 1 0 10 3 14 1 120 0 108 ->
<- 0 10 0 205 0 1 0 10 3 14 1 120 0 108 0 1 0 10 0 10 0 205 4 15 1 120 0 114 224 ->
<- 0 10 0 -51 0 1 0 10 124 14 1 0 121 -26 ->
<- 0 10 0 205 0 1 0 10 124 14 1 0 121 230 0 1 0 10 0 10 0 192 6 14 0 121 1 99 ->
<- 0 10 0 -51 0 1 0 10 3 14 1 123 0 111 ->
<- 0 10 0 205 0 1 0 10 3 14 1 123 0 111 0 1 0 10 0 10 0 205 4 15 1 123 0 121 234 ->
<- 0 10 0 -51 0 1 0 10 100 14 1 0 124 -47 ->
<- 0 10 0 205 0 1 0 10 100 14 1 0 124 209 0 1 0 10 0 10 0 192 6 14 0 124 1 102 ->
<- 0 10 0 -51 0 1 0 10 3 14 1 126 0 114 ->
<- 0 10 0 205 0 1 0 10 3 14 1 126 0 114 0 1 0 10 0 10 0 205 4 15 1 126 0 124 240 ->
<- 0 10 0 -34 0 1 0 10 3 14 1 -128 0 -123 ->
<- 0 10 0 222 0 1 0 10 3 14 1 128 0 133 0 1 0 10 0 10 0 222 4 15 1 128 0 100 235 ->
<- 0 10 0 -34 0 1 0 10 -56 14 1 0 -127 75 ->
<- 0 10 0 222 0 1 0 10 200 14 1 0 129 75 0 1 0 10 0 10 0 192 6 14 0 129 1 107 ->
<- 0 10 0 -34 0 1 0 10 3 14 1 -125 0 -120 ->
<- 0 10 0 222 0 1 0 10 3 14 1 131 0 136 0 1 0 10 0 10 0 222 4 15 1 131 0 200 82 ->
<- 0 10 0 -34 0 1 0 10 100 14 1 0 -124 -22 ->
<- 0 10 0 222 0 1 0 10 100 14 1 0 132 234 0 1 0 10 0 10 0 192 6 14 0 132 1 110 ->
<- 0 10 0 -34 0 1 0 10 3 14 1 -122 0 -117 ->
<- 0 10 0 222 0 1 0 10 3 14 1 134 0 139 0 1 0 10 0 10 0 222 4 15 1 134 0 100 241 ->
<- 0 10 0 -95 0 1 0 10 3 14 1 -120 0 80 ->
<- 0 10 0 161 0 1 0 10 3 14 1 136 0 80 0 1 0 10 0 10 0 161 4 15 1 136 0 0 82 ->
```

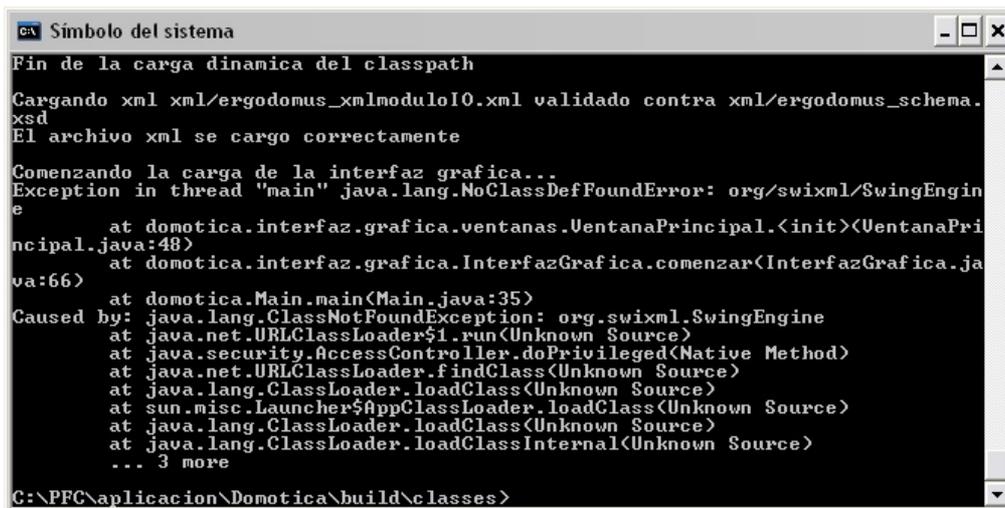
Ilustración 58: Terminal mostrando las entradas y salidas de la aplicación

Pueden verse en la imagen anterior (Ilustración 58) los bytes transmitidos en la aplicación. Las salidas de información aparecen entre los símbolos >- (comienzo) y -> (fin), mientras que las entradas lo están entre los símbolos <- (comienzo) y <- (fin).

3. Información de errores en la terminal

Como se comentaba al principio, la consola ofrece información extra valiosa sobre los errores ocurridos durante la ejecución de la aplicación. A continuación se muestran algunos ejemplos de la misma.

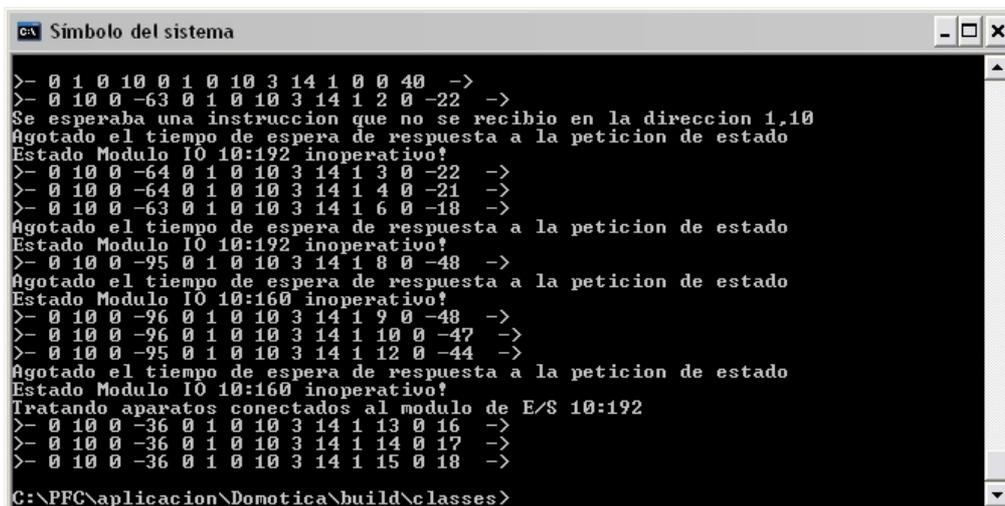
En la siguiente imagen (Ilustración 59) puede verse información interna de java ante la eliminación de una librería (concretamente, la de swixml) del directorio de recursos.



```
ca Simbolo del sistema
Fin de la carga dinamica del classpath
Cargando xml xml/ergodomus_xmlmoduloIO.xml validado contra xml/ergodomus_schema.xsd
El archivo xml se cargo correctamente
Comenzando la carga de la interfaz grafica...
Exception in thread "main" java.lang.NoClassDefFoundError: org/swixml/SwingEngine
    at domotica.interfaz.grafica.ventanas.UentanaPrincipal.<init><UentanaPrincipal.java:48>
    at domotica.interfaz.grafica.InterfazGrafica.comenzar<InterfazGrafica.java:66>
    at domotica.Main.main<Main.java:35>
Caused by: java.lang.ClassNotFoundException: org.swixml.SwingEngine
    at java.net.URLClassLoader$1.run<Unknown Source>
    at java.security.AccessController.doPrivileged<Native Method>
    at java.net.URLClassLoader.findClass<Unknown Source>
    at java.lang.ClassLoader.loadClass<Unknown Source>
    at sun.misc.Launcher$AppClassLoader.loadClass<Unknown Source>
    at java.lang.ClassLoader.loadClass<Unknown Source>
    at java.lang.ClassLoader.loadClassInternal<Unknown Source>
    ... 3 more
C:\PFC\aplicacion\Domotica\build\classes>
```

Ilustración 59: Terminal mostrando el error java ante la falta de una librería

En esta otra imagen (Ilustración 60) puede verse el proceso de carga y actualización de estado de los dispositivos almacenados en el archivo de datos de la aplicación. El puerto serie estaba desconectado del adaptador de PC, y las peticiones de estado enviadas por la aplicación no llegaban a los módulos de E/S físicos, que no podían responderlas. En estas circunstancias, para la aplicación estos módulos y sus aparatos están inoperativos hasta que respondan las peticiones, lo cual aparece reflejado en la terminal.



```
ca Simbolo del sistema
> 0 1 0 10 0 1 0 10 3 14 1 0 0 40 ->
> 0 10 0 -63 0 1 0 10 3 14 1 2 0 -22 ->
Se esperaba una instruccion que no se recibio en la direccion 1,10
Agotado el tiempo de espera de respuesta a la peticion de estado
Estado Modulo IO 10:192 inoperativo!
> 0 10 0 -64 0 1 0 10 3 14 1 3 0 -22 ->
> 0 10 0 -64 0 1 0 10 3 14 1 4 0 -21 ->
> 0 10 0 -63 0 1 0 10 3 14 1 6 0 -18 ->
Agotado el tiempo de espera de respuesta a la peticion de estado
Estado Modulo IO 10:192 inoperativo!
> 0 10 0 -95 0 1 0 10 3 14 1 8 0 -48 ->
Agotado el tiempo de espera de respuesta a la peticion de estado
Estado Modulo IO 10:160 inoperativo!
> 0 10 0 -96 0 1 0 10 3 14 1 9 0 -48 ->
> 0 10 0 -96 0 1 0 10 3 14 1 10 0 -47 ->
> 0 10 0 -95 0 1 0 10 3 14 1 12 0 -44 ->
Agotado el tiempo de espera de respuesta a la peticion de estado
Estado Modulo IO 10:160 inoperativo!
Tratando aparatos conectados al modulo de E/S 10:192
> 0 10 0 -36 0 1 0 10 3 14 1 13 0 16 ->
> 0 10 0 -36 0 1 0 10 3 14 1 14 0 17 ->
> 0 10 0 -36 0 1 0 10 3 14 1 15 0 18 ->
C:\PFC\aplicacion\Domotica\build\classes>
```

Ilustración 60: Terminal mostrando errores en la actualización de los dispositivos



Capítulo 6

CONCLUSIONES

El resultado de este Proyecto de Fin de Carrera, la aplicación de control domótico para la infraestructura del sistema domótico de *Ergodomus S.L.*, ha satisfecho las expectativas de la empresa, cumpliendo con los requisitos indispensables solicitados por ésta además de algunos otros en principio no imprescindibles.

La aplicación ha ido más allá del mismo control del sistema domótico, ya que posibilidades como almacenar un *backup* en archivos xml la configuración de una infraestructura dada permite su replicación tantas veces como sea necesaria con un coste mínimo, independizando los módulos de E/S físicos, de la programación que guardan.

Además, esto también facilita enormemente la modificación de esta programación y su análisis por el personal de la empresa.

El aspecto que ha requerido un mayor esfuerzo ha sido la especificación de requisitos y su equivalencia en cuanto a código, ya que a lo largo de todo el desarrollo del sistema los requisitos variaron y resultaron más complejos de lo inicialmente previsto.

Esta variabilidad y el amplio espectro de abstracción cubierto por la aplicación (desde el nivel más físico, con el controlador serie, hasta el nivel más abstracto, como puede ser la interfaz gráfica) ha resultado en un desafío muy completo y al mismo tiempo enriquecedor para el Proyecto de Fin de Carrera desarrollado, al ser una introducción adecuada al mundo de las aplicaciones de mediano y gran alcance.

Capítulo 7

LÍNEAS FUTURAS DE DESARROLLO

El presente proyecto de fin de carrera ha desarrollado las funcionalidades requeridas; sin embargo, su diseñado permite ser ampliado y mejorado con facilidad por otros desarrolladores.

A continuación se describen algunas de las nuevas líneas de desarrollo a seguir en el futuro.

1. NUEVAS INTERFACES DE COMUNICACIÓN

La aplicación está preparada para admitir nuevos controladores de interfaces de comunicación, como USB, bluetooth, WiFi, etc. Esta debería ser una de las primeras líneas de desarrollo a implementar dada la antigüedad del puerto serie (que actualmente puede encontrarse cada vez en menos ordenadores) y la masiva adopción de estos otros estándares.

2. NUEVOS TIPOS DE DISPOSITIVOS

El sistema domótico de *Ergodomus S.L* consta de numerosos dispositivos de distintos tipos además de los módulos de E/S y los aparatos soportados, tales como programadores, climatizadores, etc. Este tipo de dispositivos tiene un funcionamiento distinto al soportado actualmente por la aplicación que debería incluirse en un futuro cercano en la misma.

3. NUEVAS INTERFACES DE USUARIO

El sistema de control de la aplicación está preparado para que puedan ser añadidas diferentes interfaces de usuario (aparte a la gráfica existente) dada la modularización de la estructura de la aplicación y a que el paquete de control ofrece una interfaz de interacción a cualquier interfaz de usuario que quiera usarla.

4. VISTA EN 3D EN LA INTERFAZ GRÁFICA

La inclusión de una vista del sistema domótico en 3 dimensiones habría hecho el presente proyecto demasiado amplio, pero esto no ha sido impedimento para haber dejado preparada su inclusión en un futuro próximo.

El objetivo sería poder cambiar entre la vista de dos y tres dimensiones simplemente a través del panel tabulado. En la vista en 3D se ofrecería un modelo realizado en alguna aplicación de modelado 3D (ya fuera gratuita, como *Blender* o *Pov-Ray*, o de pago, como *3D Studio Max*) a través del cual el usuario podría moverse en un entorno virtual que fuera un reflejo lo más fiel posible de su propia casa, y en el que aparecieran destacados los dispositivos que el usuario podría manipular.

5. CONTROL VÍA INTERNET

Otro aspecto ampliable de la presente aplicación sería su utilización vía Internet. Esto sería muy útil para un usuario que deseara ejecutar alguna tarea en su hogar mientras estuviera fuera; también, podría resultar muy útil para que el propietario pudiera comprobar, por ejemplo, mientras está de vacaciones, que los sensores de presencia de su casa no han dado ninguna alerta hasta el momento, o que a través de una cámara pudiera comprobar el estado de su jardín y activar o no el riego en consecuencia.

Anexo I

GLOSARIO

1. DEFINICIONES

- **Aparato.** Se utiliza en la memoria para referirse a cualquier actuador que pueda conectarse a un módulo de E/S de *Ergodomus S.L.*
- **Dispositivo.** Término que engloba tanto a aparatos como a módulos de E/S, y, en general, a cualquier cosa que pueda conectarse a la infraestructura del sistema domótico de *Ergodomus S.L.*

En esta memoria se distingue con frecuencia entre dos tipos de dispositivos, los dispositivos *físicos* y los dispositivos *virtuales*. El primer término hace referencia a cualquier dispositivo existente físicamente en la infraestructura del sistema domótico, mientras que el segundo se refiere a sus equivalentes en la aplicación domótica (ya sea su representación interna o la que muestra la interfaz de usuario).

- **Domótica.** Conjunto de sistemas que proporcionan una serie de automatismos a la vivienda en la que se instalan, de tal forma que se facilite y haga más segura la vida de sus ocupantes.
- **Half-duplex.** Método de transmisión de información en el que ésta puede discurrir en los dos sentidos, pero no de forma simultánea.
- **Full-duplex.** Método de transmisión de información que permite que ésta pueda discurrir entre dos interlocutores en ambos sentidos y de forma simultánea.
- **Linux.** Sistema operativo libre y gratuito basado en Unix.
- **Módulo E/S.** Dispositivo básico de la infraestructura domótica de *Ergodomus S.L.* Controlan el funcionamiento de todos los aparatos que se conectan a ellos, y a través de ellos los usuarios pueden cambiar su estado. Se conectan unos a otros de forma modular y totalmente distribuida.

En esta memoria se usa a menudo el término abreviado *módulo* en lugar del completo *módulo de E/S*.

- **Multipunto.** Tipo de red de comunicaciones en el que sólo existe una línea de comunicación que es usada por todos los terminales de la red, que pueden ver todo lo que circula por ella y compiten o negocian por su acceso.
- **RS-485.** Protocolo de comunicaciones en bus de la capa física del modelo OSI, usado como base para el bus ED del sistema domótico de *Ergodomus S.L.*



- **Salida viva.** Característica exclusiva de las salidas de los módulos de E/S que, cuando está activada, hace que la salida correspondiente compruebe constantemente si debe cambiar de estado
- **Sun.** Su nombre completo es Sun Microsystems. Es la empresa creadora del lenguaje Java y de otros logros informáticos tan importantes como la pila de protocolos TCP/IP (el sistema de protocolos de comunicaciones a través del cual suele usarse internet).
- **Swing.** Subsistema de Java encargado de la creación de interfaces gráficas portables al 100% entre distintos sistemas.

2. ACRÓNIMOS

- **C.** Lenguaje de programación de nivel medio que no obstante ofrece funcionalidades de bajo nivel, lo que ofrece un alto rendimiento.
- **C++.** Evolución del lenguaje C, realizada con el objetivo de dotarle de orientación a objetos.
- **CSMA.** *Carrier Sense Multiple Access*, o Acceso Múltiple por Detección de Portadora. Esta técnica consiste en escuchar en el medio antes de enviar información, con el objetivo de esperar en caso de detectar una portadora y, por tanto, una situación en la que potencialmente ocurriría una colisión, con lo que se evitaría ésta.
- **CRC.** *Control de Redundancia Cíclica*. Sistema basado en una función que recibe un flujo de datos, devolviendo un único valor asociado a él. Es ampliamente usado para detectar una posible corrupción en dichos datos tras ser transmitidos por un canal.
- **CU.** *Caso de Uso*.
- **ED.** *ErgoDomus*. Bus básico en la infraestructura de las comunicaciones de los sistemas domóticos de la empresa *Ergodomus S.L.*
- **GNU/GPL.** *GNU not is Unix/General Public License*, o GNU no es Unix/Licencia Pública General.
- **EIB.** *European Instalation Bus*, o Bus de Instalación Europeo. Estándar europeo para sistemas domóticos creado para competir con los estándares japoneses y norteamericanos. Se basa en una arquitectura descentralizada, usando el protocolo de niveles OSI para las comunicaciones (cableadas) entre los módulos.
EIB aspira a ser el único estándar en Europa para instalaciones domóticas en oficinas y hogares.
- **IDE.** *Intregated Development Environment*, o Entorno de Desarrollo Integrado.
- **ISO.** *International Organization for Standarization*, o Organización Internacional para la Estandarización. Organización creada tras la Segunda Guerra Mundial con el cometido de desarrollar y fomentar estándares internacionales de productos y seguridad.
- **JAXB.** *Java Architecture for XML Binding*. Proyecto Java usado en la aplicación domótica que a partir de información almacenada en formato XML permite crear objetos Java con las características correspondientes, así como guardar de nuevo las características de dichos objetos en ese u otros archivos XML.

- **JDK.** *Java Development Kit*, o Kit de Desarrollo Java. Es un software creado por Sun que permite el desarrollo de aplicaciones Java. No permite, sin embargo, su ejecución.
- **JRE.** *Java Runtime Environment*, o Entorno de Ejecución Java. A diferencia del software anterior, el JRE provee las herramientas necesarias para ejecutar una aplicación escrita en Java, mientras que no permite su desarrollo.
- **JVM.** *Java Virtual Machine*, o Máquina Virtual Java. Es un programa (parte básica del JRE) implementado específicamente para una arquitectura y sistema operativo que interpreta código objeto de Java y lo traduce a instrucciones máquinas entendibles por la plataforma.
- **LonWorks.** Tecnología propietaria de *Echelon* para la implantación de redes domóticas distribuidas. Presenta una gran robustez y, al mismo tiempo, un alto precio que la ha reducido a ámbitos profesionales.
- **OSI.** *Open System Interconnection*. Creado por *ISO*, es el modelo de referencia usado para la definición de arquitecturas de interconexión de sistemas de comunicaciones.
- **POO.** *Programación Orientada a Objetos*.
- **RC.** *Requisito de Capacidad*.
- **RR.** *Requisito de Restricción*.
- **RS.** *Requisito Software*.
- **WiFi.** Nombre comercial (carente de significado) del estándar de comunicaciones inalámbricas IEEE 802.11, implementado y promovido por la asociación industrial denominada *Wi-Fi Alliance*. Existen diferentes versiones del estándar, de las que han surgido también diferentes versiones WiFi con distintas velocidades.
- **XML.** *Extensible Markup Language*, o Lenguaje de Marcas Ampliable. Metalenguaje con el que se pueden definir a su vez lenguajes que permitan transmitir o almacenar información con un significado asociado fácilmente tratable por una máquina.
- **XSD.** *XML Schema*. Lenguaje utilizado para describir la estructura de un archivo XML, con el mismo tipo de morfología de un archivo XML normal.
- **WWW.** *World Wide Web*, o Red Global Mundial. Término que hace referencia al conjunto de documentos de hipertexto (texto con enlaces a otros textos, imágenes y otros) distribuidos por todo el mundo y accesibles a través de internet mediante un navegador web.



Anexo II

BIBLIOGRAFÍA

1. BIBLIOGRAFÍA TRADICIONAL

1. [JUN-05] Junstrand, Stefan
Domótica y Hogar Digital
Madrid, Thomson-Paraninfo, 2005
2. [ECK-00] Eckel, Bruce
Thinking in Java, 2nd Edition
New Jersey, Prentice Hall, 2000
3. [STE-00] Stevens, Perdita
Using UML: Software Engineering with Objects and Components
Harlow, Addison-Wesley, 2000

2. BIBLIOGRAFÍA EN FORMATO ELECTRÓNICO

1. Referencias sobre Domótica

1. [D-1] Ergodomus
Última visita: 12/12/2008
<http://www.ergodomus.com>

2. [D-2] Casa domo
Última visita: 12/12/2008
<http://www.casadomo.com/>

3. [D-3] Domótica.net
Última visita: 12/12/2008
<http://www.domotica.net/>

4. [D-4] Asociación Española de Domótica
Última visita: 12/12/2008
<http://www.cedom.org/>

5. [D-5] OpenDomo (Proyecto de Domótica Open Source)
Última visita: 12/12/2008
<http://www.opendomo.com/>

6. [D-6] Proyectos Domótica
Última visita: 12/12/2008
<http://www.proyectosdomotica.com/>



2. Referencias sobre otros temas informáticos

7. [I-1] Refsnes Data

XML Tutorial

Última visita: 12/12/2008

<http://www.w3schools.com/xml/default.asp>

8. [I-2] Tiobe Software

Estadísticas de uso de los lenguajes de programación

Última visita: 12/12/2008

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

9. [I-3] Wikipedia

Capa de transporte

Última visita: 12/12/2008

http://es.wikipedia.org/wiki/Capa_Transporte