



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una aplicación domótica basada en el estándar KNX y su control web

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Ana Catalá Serrat

Tutor: Juan Vicente Capella Hernández

Cotutor: Rafael Ors Carot

2013-2014

Resumen

La temática del presente proyecto es el análisis, diseño e implementación de una plataforma web para el control de sistemas domóticos tipo KNX que puede instalarse tanto en viviendas como en edificios públicos (hoteles, hospitales, etc.). Su uso aporta la ventaja de poder ser puesto en marcha desde cualquier dispositivo (portátil, Tablet, Smartphone) conectado a internet y con un único requisito, el de disponer de un navegador web para poder monitorizar e incluso controlar los determinados aspectos de la vivienda.

Palabras clave: domótica, aplicación, *MySQL*, *PHP*, *KNX*.

Abstract

The subject matter of this project is the analysis, design and implementation of a web platform for the control of type KNX automated systems that can be installed in homes as well as in public buildings (hotels, hospitals, etc.). It has the advantage of being launched from any device (laptop, tablet, smartphone) connected to the Internet and with just a single requirement, the need to dispose of a web browser in order to monitor and even control certain aspects of housing.

Keywords: automated, application, *MySQL*, *PHP*, *KNX*.

TABLA DE CONTENIDOS

1.	INTRODUCCIÓN	11
1.1.	Motivación y justificación.....	11
1.2.	Objetivos del proyecto.....	11
1.3.	Estructura de la memoria.....	12
2.	CONCEPTOS BÁSICOS.....	14
2.1.	¿Qué es la domótica?.....	14
2.2.	¿Qué es el estándar KNX?.....	14
3.	TECNOLOGÍAS UTILIZADAS	16
3.1.	¿Por qué hemos usado XAMPP?.....	16
3.2.	¿Por qué hemos usado PHP?.....	16
3.3.	¿Por qué hemos utilizado JavaScript y jQuery?.....	17
3.4.	¿Por qué hemos usado MySQL?	17
3.5.	¿Por qué era necesario el uso de HTML?.....	17
3.6.	¿Por qué era necesario el uso de CSS?.....	18
3.7.	¿Por qué hemos usado Sublime Text?	18
3.8.	¿Por qué la utilización de Evolus Pencil?.....	18
4.	DESCRIPCIÓN DEL ESCENARIO.....	19
5.	DISEÑO E IMPLEMENTACIÓN	21
5.1.	Esquemas de diseño.....	21
5.1.1.	Esquema página de inicio de sesión	21
5.1.2.	Esquema página de diseño	22
5.1.3.	Esquema página monitorización y control	23
5.1.4.	Esquema página sistema de registro de usuarios	24
5.2.	Diagramas y diseño de la Base de Datos	24
5.2.1.	Diagrama UML de clases	25
5.2.2.	Diagrama de casos de uso	26
5.2.3.	Tabla usuarios	28
5.2.4.	Tabla estancia.....	28
5.2.5.	Tabla elemento	28
5.2.6.	Tabla elementos_gráficos	29
5.2.7.	Diagrama entidad-relación.....	30
5.3.	Diseño de las imágenes utilizadas en la aplicación.	30
5.3.1.	Logotipo o logo	30
5.3.2.	Representación del valor Off	31



5.3.3. Representación del valor On	32
5.3.4. Representación de la temperatura Alta	32
5.3.5. Representación de la temperatura Media	32
5.3.6. Representación de la temperatura Baja	33
5.3.7. Representación del nivel de bajado	33
5.3.8. Representación del nivel de subido	33
5.4. Estructura e implementación de la aplicación web	34
5.4.1. Arquitectura	34
5.4.2. Implementación del sistema de inicio de sesión	35
5.4.2.1. <i>Formulario_acceso.php</i>	35
5.4.3. Implementación del diseño de la vivienda	36
5.4.3.1. <i>Diseno.php (Formulario nueva estancia y nuevo elemento)</i>	36
5.4.3.2. <i>Diseno.php (Impresión y visualización de las estancias y sus elementos)</i>	38
5.4.4. Implementación de la monitorización y el control de la vivienda	39
5.4.4.1. <i>Control.php (Monitorización)</i>	39
5.4.4.2. <i>Control.php (Control)</i>	40
5.4.4.3. <i>Control.php (Botones generales)</i>	42
5.4.5. Implementación del sistema de registro de usuarios.	42
5.4.5.1. <i>Formulario_registro.php (registro de usuarios nuevos)</i>	42
5.4.5.2. <i>Formulario_registro.php (borrado de usuarios existentes)</i>	43
5.4.5.3. <i>Formulario_registro.php (Modificación de usuarios existentes)</i>	43
6. CONCLUSIONES	45
7. BIBLIOGRAFÍA	47
ANEXO A: INSTALACIÓN Y PUESTA EN MARCHA DE XAMPP	50
ANEXO B: MANUAL DE USUARIO	51
ANEXO C: PRUEBAS CON SMARTPHONE	65

TABLA DE ILUSTRACIONES

Ilustración 1: Esquema página de inicio de sesión.....	21
Ilustración 2: Esquema página de diseño	22
Ilustración 3: Esquema página monitorización y control	23
Ilustración 4: Esquema página sistema de registro de usuarios	24
Ilustración 5: Diagrama de clases UML	26
Ilustración 6: Diagrama casos de uso.....	27
Ilustración 7: Restricción de clave foránea de la tabla elemento	29
Ilustración 8: Restricción clave foránea de la tabla elementos _gráficos.....	29
Ilustración 9: Diagrama entidad-relación de la base de datos vivienda	30
Ilustración 10: Logotipo de la aplicación web	31
Ilustración 11: Representación del Off para el tipo "On/Off".....	31
Ilustración 12: Representación del Off para el tipo "Discretizado"	31
Ilustración 13: Representación del valor On	32
Ilustración 14: Representación de la temperatura alta	32
Ilustración 15: Representación de la temperatura media	33
Ilustración 16: Representación de la temperatura baja	33
Ilustración 17: Representación de bajar	33
Ilustración 18: Representación de subir.....	34
Ilustración 19: Esquema de la arquitectura de tres niveles.....	35
Ilustración 20: Página de inicio de sesión.....	51
Ilustración 21: Formulario acceso vacío	52
Ilustración 22: Formulario acceso con datos erróneos	52
Ilustración 23: Página principal del administrador (diseño de la vivienda I)	52
Ilustración 24: Página principal del administrador (diseño de la vivienda II).....	53
Ilustración 25: Formulario nueva estancia, con nombre repetido.....	53
Ilustración 26: Formulario nueva estancia con datos correctos	54
Ilustración 27: Visualización de la estancia introducida y su enlace de borrado.....	54
Ilustración 28: Advertencia de eliminación de la estancia seleccionada	54
Ilustración 29: Formulario nuevo elemento con datos erróneos	55
Ilustración 30: Formulario nuevo elemento con datos correctos	55
Ilustración 31: Visualización de nuevo elemento en la tabla de la estancia correspondiente.....	56
Ilustración 32: Advertencia de eliminación del elemento seleccionado	56
Ilustración 33: Inexistencia del elemento eliminado de la tabla.....	56
Ilustración 34: página principal del controlador y secundaria del administrador (control y monitorización I)	57
Ilustración 35: página principal del controlador y secundaria del administrador (control y monitorización II).....	57
Ilustración 36: página principal del controlador y secundaria del administrador (control y monitorización III)	58
Ilustración 37: página principal del controlador y secundaria del administrador (control y monitorización IV).....	58
Ilustración 38: Desplegable para el cambio de estado de un elemento	59
Ilustración 39: Valor actualizado tras el cambio de estado en el control.....	59
Ilustración 40: Botones generales.....	59

Ilustración 41: Pantalla principal y única del supervisor (monitorización I)	60
Ilustración 42: Pantalla principal y única del supervisor (monitorización II).....	60
Ilustración 43: Pantalla terciaria del administrador (Registro, modificación y borrado de usuarios)	61
Ilustración 44: Formulario registro nuevo usuario con datos erróneos	62
Ilustración 45: Formulario registro nuevo usuario con datos correctos.....	62
Ilustración 46: Formulario selección de usuario a modificar	63
Ilustración 47: Formulario con los datos de la BD a modificar.....	63
Ilustración 48: Formulario de modificación de datos enviado correctamente.....	63
Ilustración 49: Formulario borrado de usuarios.....	64
Ilustración 50: Advertencia borrado de usuarios.....	64
Ilustración 51: Prueba Smartphone, formulario inicio sesión vacío	65
Ilustración 52: Prueba Smartphone, formulario acceso datos correctos	65
Ilustración 53: Prueba Smartphone, formulario acceso contraseña incorrecta	65
Ilustración 54: Prueba Smartphone, formulario nueva estancia vacío	66
Ilustración 55: Prueba Smartphone, nueva estancia repetida	66
Ilustración 56: Prueba Smartphone, nueva estancia datos correctos	66
Ilustración 57: Prueba Smartphone, nuevo elemento repetido	66
Ilustración 58: Prueba Smartphone, desplegable selección tipo elemento	66
Ilustración 59: Prueba Smartphone, desplegable selección estancia.....	66
Ilustración 60: Prueba Smartphone, visualización estancias y elementos	67
Ilustración 61: Prueba Smartphone, mensaje confirmación eliminar elemento	67
Ilustración 62: Prueba Smartphone, monitorización.....	67
Ilustración 63: Prueba Smartphone, control.....	67
Ilustración 64: Prueba Smartphone, control general.....	67
Ilustración 65: Prueba Smartphone, registro nuevo usuario	68
Ilustración 66: Prueba Smartphone, borrar o modificar usuario	68
Ilustración 67: Prueba Smartphone, modificar usuario.....	68
Ilustración 68: Prueba Smartphone horizontal	68

TABLA FRAGMENTOS DE CÓDIGO

Fragmentos de código 1: Comprobación de la existencia de usuarios en la BD	36
Fragmentos de código 2: Obtención de los datos mediante método \$_POST.....	37
Fragmentos de código 3: Inserción y actualización de la tabla elementos_gráficos	37
Fragmentos de código 4: <i>HTML</i> para el borrado de una estancia	38
Fragmentos de código 5: <i>Script</i> de JS para el borrado de una estancia.....	38
Fragmentos de código 6: <i>PHP</i> para el borrado de una estancia mediante <i>DELETE</i>	38
Fragmentos de código 7: Impresión del tipo de elemento e imagen asociada	40
Fragmentos de código 8: Impresión del desplegable del tipo "On/Off" y su imagen	41
Fragmentos de código 9: Utilización de <i>JS</i> y <i>jQuery</i> para la obtención de los valores...	41
Fragmentos de código 10: Implementación botones generales en <i>HTML</i>	42
Fragmentos de código 11: Comprobación de los campos e introducción en la BD de usuarios nuevos	43
Fragmentos de código 12: Borrado de usuarios	43
Fragmentos de código 13: Campo identificador del tipo <i>readonly</i>	44



1. INTRODUCCIÓN

1.1. Motivación y justificación

La domótica [1] es un conjunto de sistemas capaces de automatizar una vivienda, que aporta seguridad, bienestar, comunicación y servicios de gestión energética. Es una tecnología en plena evolución que requiere ampliar su control mediante la utilización de dispositivos que cada vez son más utilizados, como pueden ser un computador, una *Tablet*¹ o un *Smartphone*², a través de aplicaciones web que facilitan esta tarea, desde cualquier lugar, sin tener la necesidad de permanecer en la vivienda en el momento de cambiar el estado de un determinado elemento.

La evolución marca el ritmo de la vida y las viviendas no se quedan atrás por lo que esta ha sido nuestra principal motivación, ya que en los últimos años habíamos creado un especial interés por los temas relacionados con la domótica al ser una tecnología novedosa, de gran difusión, futuro y de cierta consideración que está siendo adquirida para la sociedad. Al realizar este proyecto me ha dado la posibilidad de trabajar con una tecnología que casi no conozco, pero que me atrae en grandes medidas y que además me ha estado ayudando con el aprendizaje de nuevos lenguajes, como puede ser *PHP Hypertext Pre-processor (PHP)*, el cual está siendo actualmente bastante utilizado en el mundo laboral.

1.2. Objetivos del proyecto

Tenemos dos objetivos generales bien diferenciados que a su vez están compuestos de otros específicos, uno de ellos está asociado a la aplicación web y el otro está más bien relacionado con lo que son los aprendizajes autodidácticos de las diferentes tecnologías y herramientas empleadas.

El primero de los objetivos generales es el diseño e implementación de una aplicación web para un sistema domótico basado en el estándar KNX, que permita su monitorización y control de forma remota y segura, a través de cualquier dispositivo conectado a internet que tenga instalado un navegador web, con un interfaz fácil de entender, dinámico e intuitivo. Partiendo de este objetivo podemos diferenciar otros de carácter específico que intentaremos abordar:

1. Definir y diseñar las diferentes estancias que componen una vivienda.
2. Definir y crear los diferentes elementos domóticos para cada estancia.
3. Asignar un tipo a cada elemento, pudiendo ser On/Off, Discretizado o Nivel.

¹ *Tablet* [2]: computadora portátil de mayor tamaño que un teléfono inteligente o una PDA, integrada en una pantalla táctil con la que se interactúa con los dedos o un estilete.

² *Smartphone* [3]: teléfono móvil inteligente, construido sobre una plataforma móvil, con una mayor capacidad de almacenar datos y realizar actividades semejantes a una minicomputadora.



4. Dibujar y monitorizar los elementos de cada estancia, para poder ver en qué estado se encuentran en cada momento.
5. Controlar el estado de cada elemento.
6. Asignar imágenes representativas de los estados a cada uno de los elementos.
7. Eliminar cada uno de los elementos y las estancias.
8. Diferenciar varias acciones entre los distintos tipos de usuario: Administrador, Supervisor o Controlador.
9. Crear usuarios nuevos.
10. Modificar y borrar usuarios.

El segundo de los objetivos generales, y no por ser el segundo es de menor importancia que el anterior, es aprender de manera autodidáctica nuevos lenguajes y nuevas tecnologías que en un futuro pueden ser de gran ayuda de cara al mundo laboral, además de aplicar los conocimientos aprendidos durante la carrera. Si partimos de este objetivo también podemos extraer otros más específicos:

1. Aplicar los conocimientos aprendidos en algunas de las asignaturas de la carrera, haciendo especial hincapié en *Base de datos y sistemas de la información, Desarrollo web, Integración de aplicaciones, Ingeniería del Software, y Diseño de aplicaciones multimedia interactivas para televisión*.
2. Estudiar y aprender de forma autodidáctica, con la ayuda de internet y otros documentos [28-37], el lenguaje de programación de código del lado del servidor *PHP*.

1.3. Estructura de la memoria

Este documento se estructura en siete capítulos, el primero de los cuales es esta presente introducción, que hemos dividido en tres apartados para dejar de una manera clara y concisa cual es la justificación y la motivación que nos ha llevado hasta la realización del presente proyecto, cuáles son los principales objetivos que hemos abarcado y como hemos estructurado nuestra memoria, para facilitar y ampliar la información que refleja la tabla de contenidos permitiendo una localización rápida de los diferentes apartados.

El segundo de los capítulos abarca la definición de los conceptos básicos principales, para poder comprender exactamente el entorno en el que nos encontramos. Concretando tendremos una definición de que es la domótica y para que se utiliza, así como también explicaremos que estándar ha sido elegido, para llevar a cabo esta aplicación.

En el siguiente de los capítulos, el tercero, se realiza un estudio de las diferentes tecnologías y herramientas utilizadas para la posible realización de la aplicación web, dando a entender el porqué de cada una de las elecciones.

El cuarto capítulo engloba la descripción del escenario para la realización, creación y diseño de la Base de Datos, con sus respectivas relaciones y atributos, lo cual nos ayuda a comprender con facilidad que es lo que debemos

tener almacenado en la Base de Datos en cada momento y para que nos sirva esta manera de almacenarlo.

El capítulo quinto es el de mayor envergadura puesto que abarca todo el diseño e implementación de lo que es la plataforma web para el control domótico de la vivienda, en el que se detallan varios subapartados, entre los que destacamos: los esquemas de diseño iniciales, el diseño de la base de datos, donde incluimos el diagrama de clases UML, el diagrama de casos de uso, el diagrama entidad-relación, y la descripción de cada una de las tablas, así como también destacamos otros dos subapartados en los que se describe el diseño de las imágenes utilizadas en la aplicación y la estructura e implementación de la misma.

En el sexto se presentan las conclusiones del presente proyecto. Y en el séptimo y último capítulo se ven reflejadas todas las referencias bibliográficas y los sitios web visitados que han hecho posible la realización, tanto de la memoria como de la plataforma web.

Por último, en el anexo se incluye un manual para la puesta en marcha del servidor *XAMPP*. El siguiente anexo es un manual de usuario de la plataforma, que ayudará en gran medida a estos en el manejo total de la misma, en otras palabras, se explicará detalladamente cual es el funcionamiento de la aplicación para los distintos tipos de usuario existentes. Y finalmente se mostrará un tercer anexo en el que se podrá ver el correcto funcionamiento de la aplicación en otros dispositivos completamente diferentes a lo que es un ordenador, como puede ser la utilización de un *Smartphone*.



2. CONCEPTOS BÁSICOS

2.1. ¿Qué es la domótica?

Se entiende por domótica [1] el conjunto de sistemas capaces de automatizar una vivienda, aportando seguridad, bienestar, comunicación y servicios de gestión energética, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, ya sean cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar.

La domótica integra una serie de automatismos en materia de electricidad, electrónica, robótica, informática y telecomunicaciones, con el objetivo de asegurar al usuario un aumento de la programación y ahorro energético, confort, seguridad, comunicaciones y accesibilidad. Aunque todos se pueden ver reflejados en nuestro proyecto, destacamos dos de ellos, que en parte son los que más nos afectan.

El primer aspecto destacado es el confort, el cual conlleva todas las actuaciones que se pueden llevar a cabo mejorando el confort de una vivienda, como puede ser el apagado general de todas las luces de la vivienda, la automatización del apagado o encendido en cada punto de luz, la automatización de cualquier sistema, instalación o equipo dotado de control eficiente, etc.

El segundo aspecto que también destacamos y que está muy presente son las comunicaciones, los sistemas o infraestructuras de comunicaciones que posee el hogar. Estas permiten la ubicuidad en el control tanto externo como interno, control remoto desde Internet bien sea desde un computador, una Tablet, un Smartphone, etc.

2.2. ¿Qué es el estándar KNX?

El estándar *KNX* o *Konnex* [4] es el único estándar abierto a nivel mundial para el control tanto de casas como de edificios. Es un sistema descentralizado en el que todos los dispositivos pueden ejercer una serie de funciones de forma autónoma o en relación con otros dispositivos.

Todos los elementos que se conectan al bus de comunicación de datos tienen su propio microprocesador por lo tanto si un elemento falla, el sistema domotizado puede seguir funcionando.

Con *KNX* a través de un sencillo panel de control se pueden controlar todas las aplicaciones de viviendas y edificios. Desde calefacción, ventilación, control de accesos, hasta control remoto de todas las aplicaciones del hogar, este estándar permite nuevas formas de incrementar el confort, la seguridad y el ahorro energético de una vivienda.

Otro aspecto a destacar es el *software* [5] relacionado con este estándar, el cual se encuentra dividido en dos grupos:

- 1. Software de gestión:** utilizado para configurar los dispositivos y dotarlo de inteligencia, con el *EIB Tools Software* (ETS). Es un programa bajo plataforma Windows que permite relacionar actuadores con sensores y traducir las comunicaciones a través de las pasarelas. Es la única forma de configurar los dispositivos *KNX*.
- 2. Software de control:** Puede ser cualquiera, puesto que a través de las pasarelas se pueden enlazar con computadoras, servidores y aplicaciones de móviles, tabletas, etc. De esta manera se pueden visualizar y controlar las instalaciones con equipos de diferentes proveedores.

3. TECNOLOGÍAS UTILIZADAS

Uno de los primeros pasos a realizar era la preparación del entorno, por lo que tuvimos que comenzar a instalar varios archivos y ejecutar varias líneas de comando para poder instalar *PHP* y *My Structured Query Language (MySQL)*, pero tras llegar a la mitad del proceso nos dimos cuenta de que no sería necesario instalar una a una las herramientas necesarias, ya que en cada una de las instalaciones nos surgía un error nuevo. Por esta razón comenzamos a pensar en otras opciones que nos facilitasen un poco el camino para llegar a tener preparado el entorno lo antes posible. Así que se nos ocurrió que en una de las asignaturas cursadas durante la carrera, concretamente en *Integración de aplicaciones*, habíamos utilizado el denominado *XAMPP*.

3.1. ¿Por qué hemos usado *XAMPP*?

XAMPP [6] es una distribución de *Apache* completamente gratuita, fácil de instalar, configurar y usar, además de ser capaz de interpretar páginas dinámicas, que contienen tanto *PHP* como *MySQL*, es decir, lo que estábamos buscando desde un principio. Esta distribución de *Apache* solo requería la descarga y ejecución de un archivo *ZIP*, con algunas configuraciones sumamente pequeñas en alguno de sus componentes, necesarias para el correcto funcionamiento del servidor web. La versión utilizada ha sido la *v3.2.1*.

Otra de las ventajas por la que nos decantamos por esta distribución es su actualización regular para incorporar las últimas versiones de *Apache*, *MySQL*, *PHP* y *Perl*, lo que permite evitar errores y permanecer en todo momento a la última. Además también incluye otros módulos como *OpenSSL* y *phpMyAdmin*, en nuestro caso hemos hecho uso solamente del segundo, el cual nos ha facilitado en gran medida el diseño, la creación y la gestión de la base de datos.

Por último indicar que esta decisión fue tomada principalmente por el hecho de estar todo integrado en una misma aplicación lo cual nos permitiría que todo fuera mucho más cómodo y fácil de gestionar. Además de que el tiempo utilizado en realizar la configuración del mismo no era nada comparable con el que se hubiera invertido si se hubieran instalado paso a paso cada uno de estos componentes.

3.2. ¿Por qué hemos usado *PHP*?

PHP [7] es un lenguaje de programación de uso general de código del lado del servidor, que se puede incorporar directamente en un documento *Hyper Text Markup Language (HTML)* en lugar de llamar a un archivo externo que procese los datos, originalmente diseñado para el desarrollo web de contenido dinámico con acceso a información almacenada en una base de datos.

Una de las razones por la que elegimos este lenguaje es su seguridad y confiabilidad puesto que el código fuente escrito en *PHP* es invisible al navegador

web y al cliente, ya que es el servidor el único que se encarga de ejecutar el código y enviar su resultado mediante *HTML* al navegador.

Pero nuestra principal razón fue su gran parecido con los lenguajes más comunes de programación estructurada como pueden ser *C*, *Java* o *Perl*, lo que nos permitió crear una aplicación compleja con una curva de aprendizaje sumamente corta e involucrarnos en una aplicación de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

3.3. ¿Por qué hemos utilizado *JavaScript* y *jQuery*?

JavaScript (*JS*) [8] es un lenguaje interpretado, que no requiere compilación. Es similar a *Java* aunque no es un lenguaje orientado a objetos, y no dispone de herencias. Este código ha sido integrado en algunas de las páginas web que componen nuestra aplicación, en concreto en el control de los elementos para poder obtener en cada uno de los desplegados su valor y posteriormente pasar a actualizarlo de una manera fácil y rápida y también para obtener las alertas de eliminación o actualización mediante *scripts*.

Asimismo también hemos utilizado la biblioteca (o *framework*) *jQuery* [9] del lenguaje *JS* más utilizada del mundo, de código abierto y de software libre, para lograr grandes resultados en un tiempo mínimo y en un menor espacio. La característica principal y también una de sus mayores ventajas es permitir el cambio de contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol *DOM* y peticiones *AJAX*, utilizando las funciones *\$()* o *jQuery()*.

3.4. ¿Por qué hemos usado *MySQL*?

MySQL [10] es un sistema de gestión de bases de datos relacional, multihilo y multiusuario de código abierto, que tiene una alta velocidad a la hora de realizar ciertas operaciones, lo que hace que sea uno de los gestores de base de datos con mejor rendimiento. Su conectividad, velocidad y seguridad permiten que el servidor de *MySQL* sea altamente apropiado para acceder a bases de datos en Internet.

La razón por la que elegimos este gestor fue su utilización en algunas de las asignaturas cursadas en estos últimos años, es decir, gracias a la familiarización que manteníamos con él mismo, a su rapidez y a su perfecta integración en *PHP*.

3.5. ¿Por qué era necesario el uso de *HTML*?

HTML [11] hace referencia al lenguaje de marcado para la elaboración de páginas web, que define una estructura básica y un código para la definición de contenido de una web, como texto, imágenes, etc.

En nuestro caso era imprescindible su uso puesto que sin él no hubiéramos podido realizar las diferentes páginas de las que consta nuestro proyecto, ya que la mayoría de ellas están realizadas en *HTML* aunque incluyan código *PHP* o, en menor cantidad, *JS*.



3.6. ¿Por qué era necesario el uso de CSS?

Cascading Style Sheets (CSS) [12] es un lenguaje de hojas de estilo utilizado para describir el aspecto y el formato de un documento escrito en un lenguaje de marcas como puede ser *HTML*. Su uso era necesario puesto que sin él no habiéramos conseguido todos los detalles incorporados en nuestra aplicación web y no habiéramos podido controlar el estilo y el formato de las múltiples páginas al mismo tiempo. Así como tampoco habiéramos tenido la flexibilidad de cambiar en cualquier momento alguna parte o la totalidad del diseño de nuestras páginas con tan solo modificar nuestra hoja de estilo, sin que ello suponga modificar el contenido.

Además otro detalle importante por el que era necesario utilizar CSS, era por el uso de los *Media Queries* [13] módulo CSS3 que permite adaptar la representación del contenido a características del dispositivo como es la resolución de pantalla, lo que ha hecho posible que pudiésemos adaptar nuestra aplicación a distintos dispositivos, ya sea una *Tablet*, *Smartphone* u ordenadores, tanto portátiles como de sobremesa.

3.7. ¿Por qué hemos usado *Sublime Text*?

Sublime Text [14] es un editor de texto y de código fuente. Hemos utilizado este editor gracias a su infinidad de grandes ventajas, entre las que destacamos: su selección múltiple de un término por diferentes partes del archivo, su soporte nativo para infinidad de lenguajes incluidos los utilizados en este proyecto *HTML*, *PHP*, *JS* o *CSS*, su coloreado y envoltura de sintaxis que ofrece el resaltado de las expresiones propias de cualquier lenguaje para facilitar su lectura, su resaltado de paréntesis e *indentación*³ cuando el usuario coloca el cursor en un paréntesis, corchete o llave, resalta esta y el paréntesis, corchete o llave de cierre o apertura correspondiente.

3.8. ¿Por qué la utilización de *Evolus Pencil*?

Evolus Pencil [15] es una herramienta que sirve para hacer presentaciones, diseño de aplicaciones y de páginas web de forma rápida y sin ninguna complicación. Para su correcta utilización no se necesita tener ningún conocimiento sobre lenguajes informáticos, por lo que facilita la obtención de un primer diseño, que sustituye al diseño en papel.

Esta herramienta ya la habíamos utilizado anteriormente en varias asignaturas cursadas este último año, y gracias a su sencillez y facilidad de uso decidimos volver a emplearla. Asimismo otra ventaja que posee y que también nos impulsó a usarla, es la posibilidad de realizar el diseño final de la web con todo tipo de detalles, con lo que se pueden llegar a efectuar las pruebas de usabilidad oportunas antes de diseñar la interfaz real de la aplicación.

³ *Indentación* [16]: Anglicismo de uso común en informática, que significa mover un bloque de texto hacia la derecha insertando espacios o tabuladores, para separarlo del margen izquierdo y distinguirlo mejor del texto adyacente. Es un tipo de notación secundaria utilizado para mejorar la legibilidad del código fuente por parte de los programadores.

4. DESCRIPCIÓN DEL ESCENARIO

Para modelar los datos de la vivienda domotizada, la base de datos debe mantener los datos sobre las estancias que posee, los elementos que contienen cada una de estas y los elementos gráficos asignados a cada uno de ellos. La descripción del mundo real a representar en la base de datos es la descrita en los siguientes párrafos.

La vivienda se divide en varias estancias. Cada estancia debe tener un nombre único, que será a su vez el identificador. Además deberá contener una ubicación, en la que se indica en qué lugar de la casa se encuentra. Y un identificador que se autoincrementa a medida que se introducen nuevas estancias, para que posteriormente puedan ser borradas, mediante el mismo. Las estancias más habituales, que toda casa suele poseer, deberán estar almacenadas por defecto, como pueden ser: el salón, el pasillo, la habitación, la cocina y el baño. Y las menos habituales, podrán ser introducidas por el usuario, como por ejemplo: el jardín, la terraza, otras habitaciones, más baños, etc. Cada estancia podrá tener varios elementos.

Cada estancia contiene un cierto número de elementos eléctricos que pueden cambiar su valor de funcionamiento mediante la domótica. Cada elemento contendrá un nombre (alarma, persiana, aire acondicionado, calefacción, luz) que deberá ser único para cada estancia y un atributo tipo, en el que se definirán los diferentes tipos a los que puede pertenecer cada elemento (*On/Off*, *Discretizado*, *Nivel*), pudiendo pertenecer cada elemento a un solo tipo.

Además se deberán mostrar los elementos gráficos, los cuales contendrán un identificador, mediante el cual obtendrá el tipo al que pertenece cada uno de ellos y diferentes imágenes que representen cada uno de los valores que puede tener cada elemento, para que el usuario mediante una representación gráfica pueda identificar de un simple vistazo en qué valor se encuentra cada elemento.

Así como también se indicará mediante tres atributos del tipo *enum* cuál es el valor o estado actual de cada elemento en un determinado momento. Para ello el atributo tipo **OnOff** contendrá dos estados, *On* u *Off*, el atributo **Discretizado** tendrá los estados, *Media*, *Baja*, *Alta* u *Off* y por último el atributo **Nivel** abarcará los valores, *Subir* o *Bajar*. Diferentes elementos gráficos podrán pertenecer a un único elemento.

También será necesario crear varios usuarios, entre los que destacaremos un usuario **administrador** que pueda realizar todas las funciones de la aplicación web, como crear nuevos usuarios, personalizar la casa o monitorizar la vivienda. Otros usuarios **controladores** que solamente podrán acceder al panel de control de los elementos de la vivienda y al apartado de monitorización, para controlar en qué estado se encuentran en ese momento los elementos. Y otros usuarios con muchos menos privilegios, que solo podrán monitorizar la casa, los usuarios **supervisores**. De cada uno de estos se deberá



Desarrollo de una aplicación domótica basada en el estándar KNX y su control web

almacenar en la base de datos un email, un nombre de usuario, que deberá ser único, una contraseña y un identificador que permita saber quién es quién.

5. DISEÑO E IMPLEMENTACIÓN

5.1. Esquemas de diseño

Para llevar a cabo el diseño y la implementación de nuestra aplicación domótica, primeramente tuvimos que realizar un pequeño croquis con la herramienta *Evolus Pencil*, en el que podíamos observar cuales eran nuestras ideas principales en un primer momento, aunque posteriormente según íbamos avanzando con la implementación estas iban modificándose en menor o mayor medida e iban incrementándose según avanzábamos con el aprendizaje, dependiendo de nuestras necesidades.

Los esquemas de diseño dibujados con la herramienta nombrada en el párrafo anterior se muestran a continuación con una pequeña descripción de lo que se tenía en mente en un primer momento. Solamente se muestran los esquemas de aquellas ventanas que eran imprescindibles, como son la página de inicio de sesión, la página de diseño, la de control y monitorización de la vivienda y la del registro de usuarios.

5.1.1. Esquema página de inicio de sesión

Está era necesaria para poder obtener un sistema de acceso a nuestra aplicación medianamente seguro, a la que solo pudieran acceder aquellos usuarios registrados, que al principio solamente eran los administradores, los únicos que podían realizar cualquiera de las funciones que ofrece nuestra aplicación.

El diagrama muestra un esquema de una página web. En la parte superior hay una barra de título de color verde oscuro con el texto "CONTROL WEB" en blanco. Debajo de esta barra, el fondo de la página es de un verde claro. En el centro de la página, el título "Inicio de sesión" está centrado. Abajo del título, hay un recuadro gris que contiene un formulario de inicio de sesión. Este formulario tiene dos campos de entrada: "User" con el texto "text" y "Password" con "*****". Debajo de los campos, hay un botón que dice "Acceder".

Ilustración 1: Esquema página de inicio de sesión

5.1.2. Esquema página de diseño

En el diseño de la vivienda era necesario dar las opciones de definir nuevas estancias y nuevos elementos para cada una de ellas, para así darle la opción al usuario administrador de poder personalizar la casa a su antojo. Desde esta misma página se da la opción de poder acceder a otras partes de la aplicación, como puede ser un botón con acceso a la monitorización y el control de los elementos de la vivienda o un enlace referencial hacia otra ventana en la que se pueden registrar nuevos usuarios.

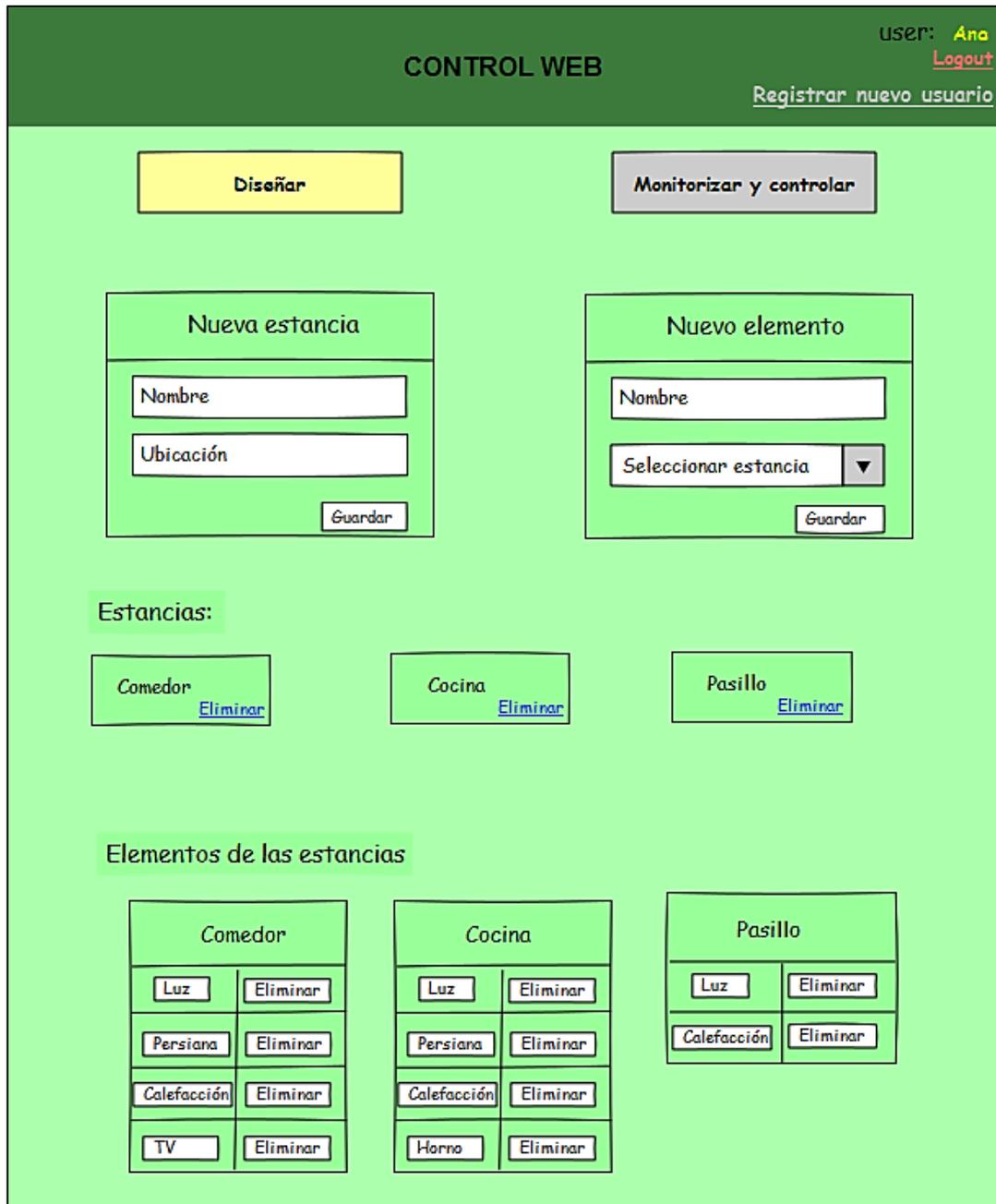


Ilustración 2: Esquema página de diseño

5.1.3. Esquema página monitorización y control

Al igual que la de diseño esta también tiene las opciones de poder acceder a otras ventanas, mediante un botón puede ser redirigido a la página de diseño y con un enlace referencial, al sistema de registro de nuevos usuarios. El control de los elementos se realizará mediante tablas que indicarán el estado en el que se encuentra cada elemento en ese mismo instante, además de dar la opción de cambiar dichos estados. En cambio la monitorización mostrará unas simples tablas que indiquen el estado tanto en imágenes representativas como en texto escrito.

user: Ana
[Logout](#)
[Registrar nuevo usuario](#)

CONTROL WEB

Diseñar

Monitorizar y controlar

Control de los elementos

Cocina	
Luz	On ▼
Calefacción	Media ▼
Persiana	Subir ▼
Horno	Media ▼

Comedor	
Luz	Off ▼
Calefacción	Alta ▼
Persiana	Bajar ▼
TV	On ▼

Pasillo	
Luz	On ▼
Calefacción	Media ▼

Monitorización de los elementos

Cocina	
Luz	
Calefacción	
Persiana	
Horno	

Comedor	
Luz	
Calefacción	
Persiana	
TV	

Pasillo	
Luz	
Calefacción	

Ilustración 3: Esquema página monitorización y control

5.1.4. Esquema página sistema de registro de usuarios

Está página en un primer momento no se había incorporado mediante un botón al igual que los de control o diseño, pero como se ve más adelante se acaba incorporando uno nuevo en el que se da la opción de registrar nuevos usuarios y modificar o eliminar los ya existentes. Pero como en este apartado se muestra lo que se tenía pensado antes de realizar la aplicación, comprobaremos que no era la mejor forma de hacerlo puesto que no se podían modificar ni borrar usuarios y al ser un enlace poco visible no se podía ver con facilidad.

user: **Áno**
[Logout](#)

CONTROL WEB

Diseñar Monitorizar y controlar

Introduzca las credenciales del nuevo usuario

NUEVO USUARIO

Usuario

E-mail

Contraseña

Repetir contraseña

Tipo de usuario

Guardar Cancelar

Ilustración 4: Esquema página sistema de registro de usuarios

5.2. Diagramas y diseño de la Base de Datos

Una base de datos es un almacén que nos permite guardar grandes cantidades de información, pertenecientes a un mismo contexto, de forma organizada para que posteriormente podamos encontrarla y utilizarla fácilmente. A pesar de que existen diferentes tipos, hemos optado por utilizar *MySQL*, que

aunque no es recomendable utilizarla para grandes volúmenes de datos si se caracteriza por su rapidez y en nuestro caso es la que mejor se adapta.

Para poder manejar la administración de *MySQL* a través de nuestra aplicación web, fue necesaria la utilización de la herramienta asignada para ello, la denominada *phpMyAdmin*, la cual se encuentra incorporada como módulo en *XAMPP*. A través de dicha herramienta hemos creado una base de datos denominada **vivienda** con cuatro tablas o relaciones, partiendo del sistema de información descrito en el capítulo “4. Descripción del escenario”, junto con la obtención de su diagrama de clases en *UML*.

Las tablas creadas son del tipo *InnoDB*⁴, esta elección ha sido tomada ya que era necesario el uso de integridades referenciales, además de fijarnos en una de sus principales características, el soporte de transacciones del tipo *Atomicity, Consistency, Isolation and Durability (ACID* ⁵) y bloqueo de registros. Primeramente empezamos a asignarles el tipo *MyISAM*⁶, pero comprobamos que no nos sería de gran ayuda a la hora de relacionar internamente las diferentes tablas y realizar las restricciones de clave foránea pertinentes, puesto que *MyISAM* no posee esta última opción y además ofrece una fiabilidad y consistencia mucho menor que la ofrecida por *InnoDB*.

5.2.1. Diagrama UML de clases

El diagrama *UML* de clases o diseño conceptual [17] es un diagrama estático que describe la estructura de un sistema mostrando sus clases, es decir mostrando una plantilla para la creación de objetos de datos según el modelo que hemos predefinido en el apartado “4. DESCRIPCIÓN DEL ESCENARIO”. En cada clase se define un conjunto de variables y los métodos apropiados para operar con dichos datos.

En nuestro caso los métodos introducidos en cada una de las clases no se implementan como tales, aunque su función final sea la misma, es decir todas las operaciones indicadas en el diagrama mostrado en la *Ilustración 5*, se implementan pero sin la utilización de métodos como tales, tal y como se explica en el apartado “5.5. Estructura e implementación de la aplicación web”.

El diagrama de clases ha sido desarrollado con la aplicación *StarUML* [18], herramienta UML de código abierto que permite la creación de diagramas de forma rápida y sin necesidad de estar familiarizado con el mismo.

La *Ilustración 5* muestra las cuatro clases que contiene nuestra aplicación, con sus relativos atributos y métodos así como también presenta una restricción de integridad, restricción que asegura la integridad entre las claves foráneas y primarias, y con la que nos aseguramos que nunca haya dos elementos iguales en una misma estancia.

⁴ InnoDB [19]: mecanismo de almacenamiento de datos de código abierto para la BD MySQL.

⁵ ACID [20]: conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción, el sistema de gestión de base de datos debe velar para que se cumplan estas propiedades, atomicidad, consistencia, aislamiento y persistencia.

⁶ MyISAM [21]: mecanismo de almacenamiento de datos usada por defecto por el sistema administrador de bases de datos relacionales MySQL.



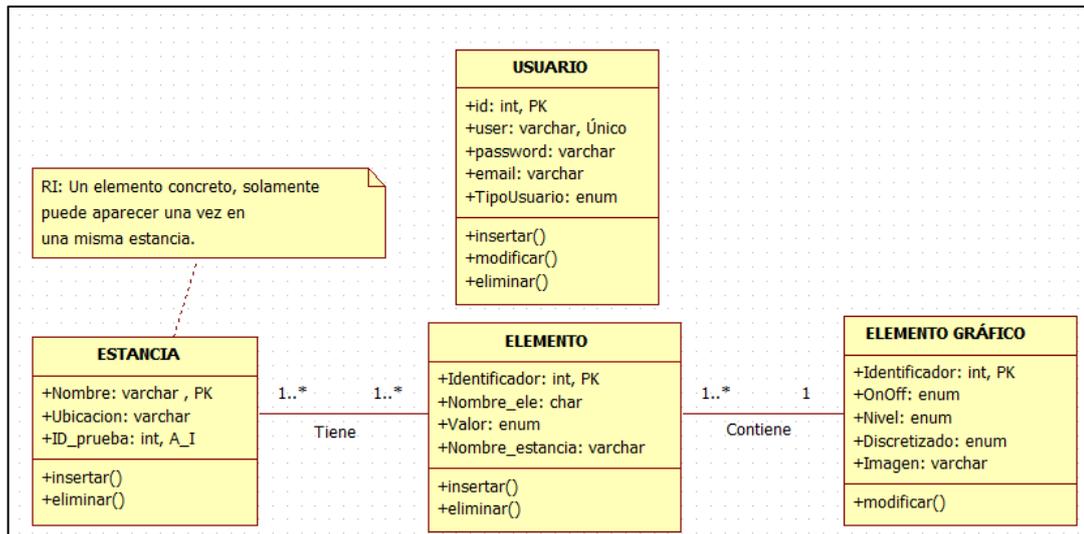


Ilustración 5: Diagrama de clases UML

5.2.2. Diagrama de casos de uso

Un caso de uso [22] es una descripción de las actividades que deben realizarse para llevar a cabo algún proceso. Los personajes que participan en el caso de uso son los denominados actores. Este diagrama sirve para la comunicación e interacción de los usuarios con nuestro sistema o aplicación web.

El diagrama creado con *StarUML* mostrado en la *Ilustración 6* muestra los casos de uso que pueden producirse por cada uno de los actores definidos, los actores en nuestro caso son los diferentes tipos de usuario existentes, los cuales son relacionados mediante vínculos de generalización. Esto es, uno de los actores especializados (Administrador) hereda los casos de uso de otro actor especializado (Controlador) y estos dos heredan los casos de uso del actor general (Supervisor). El supervisor es el actor general puesto que es el que menos acciones puede realizar, sin embargo los otros dos tipos son los especializados ya que tienen otros casos de uso propios que no están disponibles en los demás actores.

Los casos de uso definidos describen el objetivo de cada actor a la hora de utilizar nuestro sistema y la secuencia de pasos que hay que seguir para conseguir los objetivos definidos. Para la realización de esta secuencia de pasos hemos hecho uso de tres tipos de relaciones [23]:

- **Association:** Una asociación indica que el actor está interactuando con el sistema para llevar a cabo un caso de uso. (Ejemplo: enlace entre el actor Supervisor y el caso de uso Iniciar sesión)
- **Include:** Utilizado para indicar que se especifica un comportamiento común entre uno o varios casos de uso. (Ejemplo: Controlar la vivienda incluye la modificación de cada elemento, el apagado general y el bajado general)
- **Extend:** Este puede hacer ver que un caso de uso puede añadir funcionalidad a otro (Ejemplo: Si se crea un estancia se añade la funcionalidad de poder ser borrada)

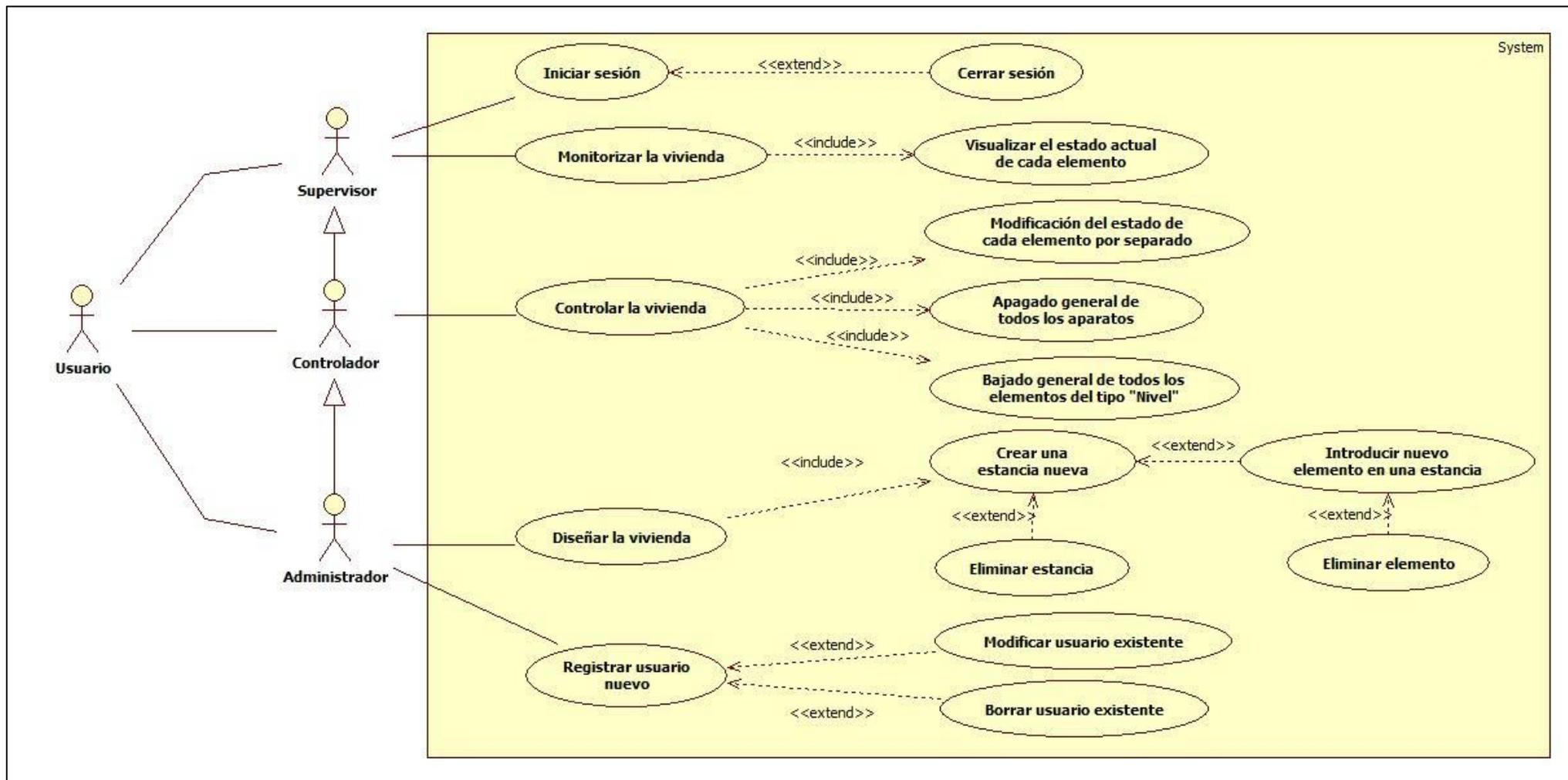


Ilustración 6: Diagrama casos de uso

5.2.3. *Tabla usuarios*

La primera de las relaciones que introducimos en la base de datos fue la de los **usuarios**, en la cual se almacenan los diferentes usuarios registrados por el administrador. En esta se pueden observar cinco atributos entre los que diferenciamos: un identificador (*id*) y un nombre de usuario (*user*), ambos claves primarias para su posible modificación o eliminación, un correo electrónico (*email*) para contactar con el usuario en caso de que se produjese algún problema, una contraseña (*password*) que le permita acceder de forma segura a la aplicación y un tipo de usuario (*tipo*) cuyo campo es del tipo *enum* que permite la elección de uno solo, bien sea Administrador, Controlador o Supervisor.

5.2.4. *Tabla estancia*

La segunda de las tablas necesaria fue la que almacena las estancias que el administrador ha introducido, denominada **estancia**, la cual contiene tres atributos: el nombre de cada una de las estancias (*Nombre*) que es clave primaria al igual que el identificador (*ID_prueba*), a través del cual se realiza el borrado de las mismas, y una ubicación (*Ubicacion*), este último atributo es menos necesario pero es de gran ayuda si se diese el caso de que hubiera dos plantas en la vivienda, y se quisiera saber con exactitud cuál es la que se está modificando.

5.2.5. *Tabla elemento*

La tercera tabla creada fue la que almacena los elementos de cada estancia denominada **elemento**, la cual contiene tres atributos: un identificador (*Identificador*), un nombre para cada uno de los elementos (*Nombre_ele*), que puede estar repetido en la base de datos pero con la restricción de que dicho nombre no aparezca duplicado en una misma estancia, y un tipo (*Tipo*) referente a los estados en los que se pueden encontrar los diferentes elementos, ya sean:

- “On/Off” para la iluminación, las alarmas, la bomba de agua de una piscina, etc.
- “Discretizado” para todos los elementos que tengan como función controlar la temperatura, como puede ser un horno, una tostadora, la refrigeración de cualquiera de las estancias, etc.
- “Nivel” para aquellos elementos cuya función sea subir o bajar, podemos introducir en este grupo los toldos, las persianas, puertas de garajes, etc.

La tabla **elemento** se relaciona con **estancia**, siendo la primera de estas hija de la segunda, en la que se han creado índices en el atributo *Nombre_ele* y el *Identificador*, y se ha añadido una restricción de clave foránea que permitirá el posterior borrado y modificación en cascada de cualquiera de las estancias que se deseen eliminar o cambiar, en otros términos, si se desea borrar una estancia también se borrarán las filas que contengan el mismo nombre de la estancia borrada en la tabla elemento.

Columna	Relación interna	Restricción de clave foránea (INNODB)
Identificador	vivienda	vivienda
Nombre_ele	vivienda	¡No se ha definido ningún índice! Cree uno más abajo
Nombre_estancia	vivienda estancia Nombre	vivienda estancia Nombre
		Nombre de la restricción elemento_ibfk_1
		ON DELETE CASCADE
		ON UPDATE CASCADE
Valor	vivienda	¡No se ha definido ningún índice! Cree uno más abajo

Ilustración 7: Restricción de clave foránea de la tabla elemento

5.2.6. Tabla elementos_gráficos

La última en ser diseñada fue la de los **elementos gráficos** tabla en la que se almacenan los estados de cada elemento y las imágenes asociadas a cada uno de estos. Contiene cinco atributos, tres son similares puesto que son del tipo *enum* y almacenan diferentes opciones, según pertenezcan a un tipo u otro:

- “OnOff”: contiene los valores “On” y “Off”.
- “Discretizado”: contiene los valores “Media”, “Baja”, “Alta” y “Off”.
- “Nivel”: contiene los valores “Subir” y “Bajar”.

De estos tres elementos siempre deberá haber dos que sean nulos, es decir, si por ejemplo estamos introduciendo el elemento persiana, el valor del atributo “OnOff” y el del atributo “Discretizado” deberá ser NULL, y el valor de “Nivel” deberá estar a “Subir” o a “Bajar”, ya que una persiana pertenece al tipo “Nivel”.

Otro de los atributos es un identificador que se corresponde con el identificador de la tabla **elemento**, y una imagen que según el valor asociado a cada elemento se le asigna una u otra.

Los **elementos gráficos** son relacionados con la tabla **elemento** mediante su identificador, identificador idéntico en ambas tablas, para una mayor facilidad a la hora de obtener el valor asignado a cada elemento y relacionarlo con el atributo tipo de la tabla **elemento**.

Columna	Relación interna	Restricción de clave foránea (INNODB)
Identificador	vivienda elemento Identificador	vivienda elemento Identificador
		Nombre de la restricción elementos_gráficos_ibfk_1
		ON DELETE CASCADE
		ON UPDATE CASCADE
OnOff	vivienda	¡No se ha definido ningún índice! Cree uno más abajo
Discretizado	vivienda	¡No se ha definido ningún índice! Cree uno más abajo
Nivel	vivienda	¡No se ha definido ningún índice! Cree uno más abajo
imagen	vivienda	¡No se ha definido ningún índice! Cree uno más abajo

Ilustración 8: Restricción clave foránea de la tabla elementos_gráficos



5.2.7. Diagrama entidad-relación

Tras la descripción del procedimiento llevado a cabo para el diseño de nuestra base de datos **vivienda**, mostramos a continuación el esquema del modelo relacional obtenido, con sus pertinentes relaciones y atributos, así como sus claves primarias y el tipo de datos asignado a cada tipo de atributo. La cardinalidad de la relación izquierda es 1:N, una estancia tiene N elementos y la de la relación derecha es 1:1, un elemento tiene asignado un elemento gráfico, dependiendo del estado en el que se encuentre.

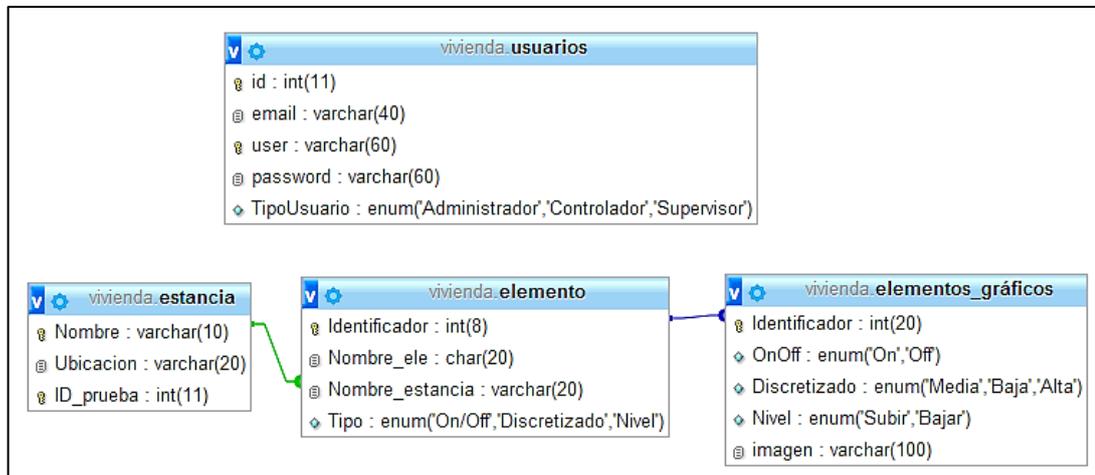


Ilustración 9: Diagrama entidad-relación de la base de datos vivienda

5.3. Diseño de las imágenes utilizadas en la aplicación.

Para llevar a cabo el diseño completo y perfeccionista de nuestra aplicación web tuvimos que diseñar varios dibujos e imágenes, como pueden ser el logotipo, signo gráfico que identifica a nuestro producto, y los iconos que indican el estado de cada uno de los elementos.

5.3.1. Logotipo o logo

Como venimos diciendo es un signo gráfico que identifica a nuestra aplicación, por lo que pensando en todo lo que nuestra aplicación realizaba, obtuvimos varias ideas y las juntamos todas en una. Las ideas obtenidas eran las siguientes:

- **Domótica**, representada mediante un botón, ya que todo se puede controlar mediante pulsadores.
- **Control web**, representado por las tres uves doble escritas en la parte inferior del botón, las que simbolizan el uso del internet y la web.
- **Vivienda**, representada dentro del botón con una silueta de un hogar.
- **Usuarios**, representados por la mano que pulsa el botón y puede controlar cualquier elemento domotizado de la vivienda.

Tras realizar varios logotipos que contuvieran dichas ideas, topamos con el que se enseña a continuación, sin duda alguna el mejor que podríamos haber encontrado y el de mayor facilidad de entendimiento.

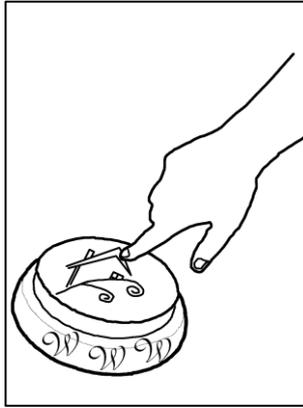


Ilustración 10: Logotipo de la aplicación web

5.3.2. Representación del valor Off

Para la representación del valor Off han sido necesarios dos iconos, uno para los elementos del tipo “On/Off” y otro para los elementos del tipo “Discretizado”. Creíamos conveniente la realización de ambos puesto que cada uno pertenece a un tipo diferente y no queríamos crearles ciertas confusiones a los usuarios.

Para el tipo “On/Off” realizamos el diseño de la silueta de una bombilla apagada, ya que cualquier aparato apagado no tiene ninguna luz encendida, y además al estar mayormente asociado con la iluminación, era el símbolo que mejor se acoplaba.



Ilustración 11: Representación del Off para el tipo “On/Off”

Para el tipo “Discretizado” realizamos el diseño de la silueta de un termómetro vacío con las letras OFF en color rojo, ya que todos los elementos discretizados contienen ciertas temperaturas, y a la hora de ser apagados la temperatura desciende hasta cero grados.

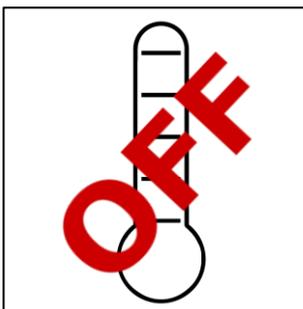


Ilustración 12: Representación del Off para el tipo "Discretizado"

5.3.3. *Representación del valor On*

Para la representación del valor On simplemente ha sido necesario colorear la silueta de la bombilla apagada, mostrada en el anterior apartado, para que aparezca como activada y los usuarios puedan ver que los elementos se encuentran encendidos con gran facilidad.

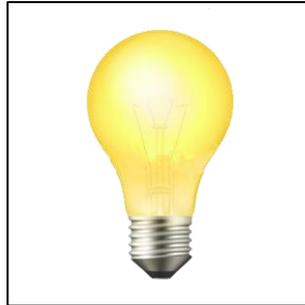


Ilustración 13: Representación del valor On

5.3.4. *Representación de la temperatura Alta*

La representación de la temperatura alta correspondiente a algún elemento del tipo “Discretizado” se vio conveniente ser representada por un termómetro que contuviera el máximo nivel de temperatura, el representado por el color rojo.

El termómetro representa lo que serían los distintos niveles de temperatura que puede llegar a alcanzar un elemento de este tipo, como puede ser un horno, un aparato de aire acondicionado o calefacción o una tostadora de pan, etc.

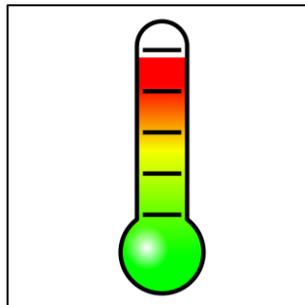


Ilustración 14: Representación de la temperatura alta

5.3.5. *Representación de la temperatura Media*

La representación de la temperatura media correspondiente a algún elemento del tipo “Discretizado” se vio conveniente ser representada por un termómetro que contuviera un nivel intermedio de temperatura, el representado por el color amarillento.

A medida que la temperatura va en descenso el color de un termómetro va decreciendo, desde un color rojo intenso hasta un color verde, pasando por el color amarillento o anaranjado, el cual indica estabilidad o mantenimiento de la temperatura.

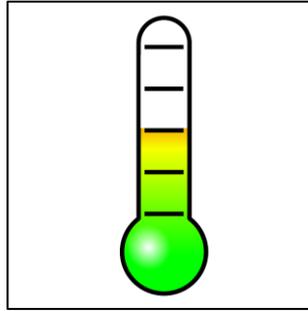


Ilustración 15: Representación de la temperatura media

5.3.6. Representación de la temperatura Baja

La representación de la temperatura baja, al igual que la temperatura media o alta se representa mediante un termómetro, pero con un nivel de grados mucho menor, tan menor que es representado por el color verde, el cual indica poca temperatura.

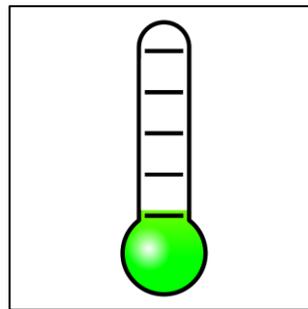


Ilustración 16: Representación de la temperatura baja

5.3.7. Representación del nivel de bajado

El nivel de bajado significa que cualquier persiana, toldo, puerta de garaje o cualquier otro elemento similar, puede ser bajado en cualquier momento, por eso hemos representado este nivel mediante una flecha que indica bajar cualquier elemento. Esta coloreada de rojo porque indica el menor de los niveles.

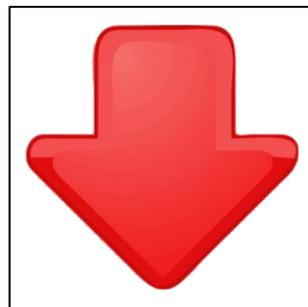


Ilustración 17: Representación de bajar

5.3.8. Representación del nivel de subido

Al igual que el nivel de bajado, es representado mediante una flecha pero en este caso una flecha hacia arriba y de color verde para indicar el subido de una persiana o de un toldo o cualquier elemento perteneciente al tipo “Nivel”.

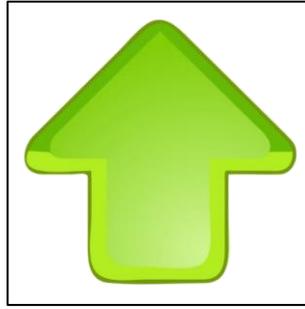


Ilustración 18: Representación de subir

5.4. Estructura e implementación de la aplicación web

En este apartado se describe que estructura sigue nuestra aplicación web y los detalles de su implementación. Tal y como se ha explicado en el apartado “1.2. Objetivos del proyecto”, el objetivo a alcanzar con la aplicación es la creación de una página que permita añadir, ver y controlar los diferentes aparatos de nuestra vivienda. Para cumplir con dicho objetivo ha sido necesaria su escritura en *HTML* con *PHP* embebido, y un poco de *JS* que opera sobre una base de datos *MySQL*, así como también *CSS*, para modificar de una forma sencilla y limpia su aspecto visual, es decir el estilo de la web.

5.4.1. Arquitectura

La arquitectura [24] utilizada se asemeja a la de tres capas, decimos se asemeja porque realmente no hemos estructurado el código de esta manera, sino que lo hemos estructurado en cuatro grandes archivos que a su vez llaman a otros más pequeños, estos dividen la aplicación en cuatro bloques: el inicio de sesión, el diseño, el control y monitorización de la vivienda, y el registro de usuarios. Lo que sí que podemos decir que tiene un gran parecido con esta arquitectura es el esquema obtenido al final de la implementación, el cual si se divide en tres capas:

- Un primer nivel consistente en la capa de presentación que incluye no sólo el navegador, sino también un servidor web, el cual es el responsable de presentar los datos al usuario, con una presentación amigable y fácil de usar en un formato coherente y adecuado.
- Un segundo nivel referente a los programas y scripts escritos en *PHP*, los cuales son ejecutados y reciben las peticiones del usuario, enviando posteriormente las respuestas tras el proceso. Esta capa es muy importante puesto que es donde se establecen todas las reglas que se tendrán que cumplir. Además este nivel también incluye lo que sería el servidor de aplicaciones utilizado, en nuestro caso *Apache*.
- Y un tercer nivel el cual proporciona al segundo los datos necesarios para la ejecución del código, es decir realiza las transacciones con la base de datos **vivienda**. Aquí es donde se definen las consultas que se realizan en la base de datos como pueden ser: la introducción, modificación o eliminación de ciertos elementos, estancias o usuarios.

Esta aplicación al igual que muchas otras, recoge los datos introducidos por el usuario (primer nivel), los envía al servidor web, el cual ejecuta uno o varios programas (segundo y tercer nivel), y el resultado obtenido lo muestra de nuevo al usuario en el navegador (primer nivel).

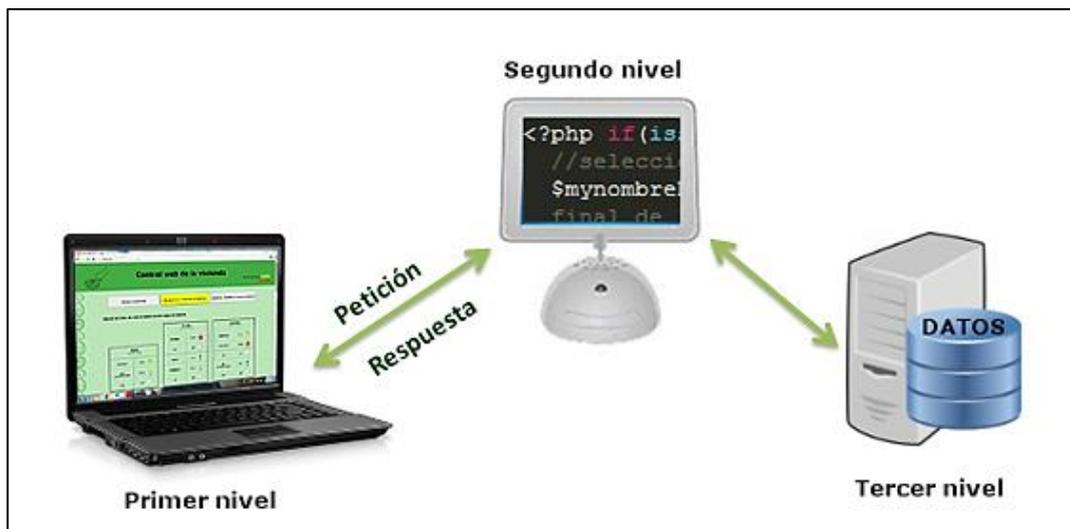


Ilustración 19: Esquema de la arquitectura de tres niveles

5.4.2. Implementación del sistema de inicio de sesión

Para poder acceder a la aplicación es necesario ser usuario de la misma, por lo que tuvimos que crear un sistema de inicio de sesión. Para su implementación fue necesario un único archivo, el cual introduce nuevas redirecciones a otros a los que solamente se puede acceder si el usuario se encuentra en la base de datos.

5.4.2.1. Formulario_acceso.php

Como su nombre indica es un formulario diseñado para poder acceder a la aplicación web, con un usuario y una contraseña. Este formulario ha sido implementado en código *HTML*, con dos campos uno del tipo *text*, en el que se introduce el usuario y otro del tipo *password* en el que se inserta una contraseña. Este compone lo que es la página de inicio de la aplicación ya que es la primera que ve el usuario antes de introducir sus credenciales.

En este archivo se realiza una comprobación de los datos que introduce el usuario para asegurarse de que estos sean correctos y se encuentren en la base de datos, lo que conlleva a realizar la conexión con la base de datos **vivienda**, para poder realizar cualquier transacción, y posteriormente hacer una consulta a la tabla **usuarios** utilizando código *PHP* y las funciones de *MySQL* (*mysql_query()*, *mysql_num_rows()* y *mysql_fetch_array()*).

Si se cumple la comprobación descrita en el párrafo anterior, y además el usuario es del tipo Administrador, será redirigido al archivo diseño.php, en el caso contrario, en el que sea Supervisor o Controlador será redirigido a control.php. Si la comprobación del párrafo anterior no se cumple, es decir el usuario no pertenece a la base de datos entonces realizamos un *else* e imprimimos un mensaje en rojo

dentro del mismo formulario, alertando al usuario de que su nombre o contraseña no son correctos. A continuación se muestra un fragmento de lo que es el código que comprueba la existencia de los usuarios en la base de datos.

```
<?php
//consultamos a la BD vivienda los usuarios disponibles en la tabla usuarios
$sql= "select * from ".$tbl_name." where "." user = '".$myusername.'" AND "." password = '".$mypassword.'";
$result=mysql_query($sql);
$count=mysql_num_rows($result);

if (($count==1)) {
    $_SESSION["myusername"]=$myusername;
    $_SESSION["mypassword"]=$mypassword;

    //si es un usuario controlador o supervisor será redirigido a la página de control.php, no
    pudiendo acceder a ninguna otra parte de la aplicación( gracias a las restricciones introducidas
    en las tres páginas correspondientes a los 3 botones de la aplicación )
    $row = mysql_fetch_array($result);

    if (($row["TipoUsuario"] == 'Controlador') || ($row["TipoUsuario"] == 'Supervisor')) {
        header("Location:control.php");
    }

    //si es un usuario administrador será redirigido a la página de diseno.php, pudiendo acceder a
    todas las demás partes de la aplicación
    elseif ($row["TipoUsuario"] == 'Administrador') {
        header("Location:diseno.php");
    }
} else {
    echo"<div id=error>Error, usuario o contraseña incorrectos</div>" ;
}
?>
```

Fragmentos de código 1: Comprobación de la existencia de usuarios en la BD

5.4.3. Implementación del diseño de la vivienda

Para hacer posible el diseño y personalización de la vivienda por parte del usuario, fue necesaria la creación de un botón que apareciese en todas y cada una de las ventanas de la aplicación, situado en la parte superior izquierda y con el nombre “Diseñar la vivienda”. Dicho botón tiene la función de re direccionar a un usuario Administrador al archivo *diseno.php*, archivo desde el cual se puede acceder a cualquiera de los apartados de la web mediante otros botones y enlaces relacionales.

Además si la condición del *if* que comprueba el inicio de sesión y el tipo de usuario, no se cumple, es decir, si una persona que no es usuario intenta acceder al archivo *diseno.php*, se le mostrará un mensaje indicándole que debe iniciar sesión como administrador para poder acceder, crear y ver nuevos contenidos.

5.4.3.1. *Diseno.php* (Formulario nueva estancia y nuevo elemento)

En este archivo primeramente se realiza la conexión con la base de datos **vivienda**, para poder realizar cualquier transacción. Se le consulta a la base de datos de que tipo es el usuario registrado, si es Administrador, el botón “Diseñar la vivienda” queda seleccionado, de manera que el usuario sepa en cada momento en que opción de la web se encuentra, dándole la posibilidad de acceder al resto de las ventanas, tanto a la de “Monitorizar y controlar la vivienda” como a la de “Registrar, modificar o borrar usuarios”. Además también se le da la opción de cerrar sesión cuando lo desee llamando al archivo *salir.php*, desde el cual se destruye la sesión iniciada y se redirige al usuario a la página de inicio de sesión.

En este mismo se han creado dos formularios en *HTML*, uno para la inserción de nuevas estancias y otro para la inserción de nuevos elementos asignados a cada una de estas. En ambos se han realizado varias consultas a la tabla **estancia** y a la tabla **elemento**, mediante las funciones de MySQL.

El primero de los formularios contiene dos campos de tipo *text* a rellenar por el usuario, el nombre de la estancia y la ubicación de la misma. Estos datos se obtienen mediante el método `$_POST` de *PHP* y si el nombre de la estancia ya está almacenado en la base de datos se le indica al Administrador mediante un mensaje de error de color rojo y se le da la posibilidad de que introduzca otra diferente. En el caso contrario se realiza un *INSERT* en la tabla **estancia**, con dichos datos. Al introducir los nuevos datos en la base de datos no tenemos por qué preocuparnos del identificador puesto que este se autoincrementa automáticamente y además no se da la opción en el formulario de introducirlo.

```
$mynombre=trim($_POST['Nombre']); //trim: Elimina espacio en blanco (u otro tipo de caracteres) del inicio y el final de la cadena
$myubicacion=trim($_POST['Ubicacion']);
```

Fragmentos de código 2: Obtención de los datos mediante método `$_POST`

El segundo contiene tres campos, dos del tipo *select*⁷, para obtener el tipo del elemento y la estancia a la que pertenecerá, y uno del tipo *text*, para la asignación de un nombre al elemento. En este punto del archivo se comprueba que no existan dos elementos iguales en una misma estancia, con una consulta a la tabla **elemento** que extrae, si la hay, la fila que contenga el nombre del elemento y la estancia introducida, atributos obtenidos mediante el método `$_POST`. Si el número de filas es distinto de cero significa que ya existe, por lo que se le avisa al Administrador que introduzca otro nombre diferente, para esa misma estancia. En el caso contrario se realiza un *INSERT* a la tabla **elemento**, introduciendo los datos recogidos desde el formulario.

Dentro del formulario para la inserción de nuevos elementos, también realizamos otro *INSERT*, a la tabla **elementos_gráficos**, para que el identificador del nuevo elemento introducido coincida con el identificador del elemento gráfico y de esta manera podamos asignarle un valor y una imagen por defecto en el mismo instante en el que es creado, actualizando la fila correspondiente con un *UPDATE*.

```
//seleccionamos el identificador y el tipo del elemento introducido anteriormente
$sql="SELECT Identificador, Tipo FROM elemento WHERE Nombre_ele='$mynombreELE' AND Nombre_estancia='$mynombreEST'";
$query = mysql_query($sql,$link);
$row = mysql_fetch_array($query);
$identificador=$row['Identificador'];
mysql_query("INSERT INTO elementos_gráficos (Identificador) values ('$identificador')",$link); //
Insertamos el identificador insertado en la tabla elemento en la tabla elementos_gráficos, para
posteriormente poder introducir su valor correspondiente.
$tipo=$row['Tipo'];
$sql="SELECT OnOff, Discretizado, Nivel, Imagen FROM elementos_gráficos, elemento WHERE elemento.
Identificador=elementos_gráficos.Identificador";

if($tipo == "On/Off"){
    mysql_query("UPDATE elementos_gráficos SET OnOff='Off', Discretizado= NULL, Nivel=NULL, Imagen=
' luz_apagada.png' where Identificador='$identificador'",$link);
}
else if($tipo == "Discretizado"){
    mysql_query("UPDATE elementos_gráficos SET OnOff=NULL, Discretizado= 'Off', Nivel=NULL, Imagen=
' tmp_apagada.png' where Identificador='$identificador'",$link);
}
else if($tipo == "Nivel"){
    mysql_query("UPDATE elementos_gráficos SET OnOff=NULL, Discretizado= NULL, Nivel='Bajar', Imagen=
' flecha_abajo.png' where Identificador='$identificador'",$link);
}
```

Fragmentos de código 3: Inserción y actualización de la tabla **elementos_gráficos**

⁷ *Select* [25]: desplegable que nos permite mostrar una lista de opciones en un formulario, para que nuestros visitantes seleccionen una o más opciones antes de enviar el formulario.



- Si el elemento es del tipo “On/Off” se le asigna el valor por defecto “Off” al atributo “OnOff”, y la imagen asociada al mismo, una bombilla apagada.
- Si el elemento es del tipo “Discretizado” se le asigna el valor por defecto “Off” al atributo “Discretizado” y la imagen asociada al mismo, un termómetro con una temperatura nula.
- Si el elemento es del tipo “Nivel” se le asigna el valor por defecto “Bajar” al atributo “Nivel” y la imagen asociada al mismo, una flecha roja hacia abajo.

5.4.3.2. *Diseno.php (Impresión y visualización de las estancias y sus elementos)*

En este mismo archivo también se realiza la impresión y visualización de las estancias de la vivienda y los elementos de cada una de ellas, con tablas. Esto se obtiene mediante la ejecución de cuatro *SELECT*⁸ a las tablas **estancia** y **elemento**. Además de la realización de varios bucle *while* que permiten la obtención dinámica de los datos, bien por defecto o bien introducidos por el usuario, y el posible borrado de cada uno de ellos, pasándole en cada llamada, a los archivos de borrado, el identificador correspondiente.

Por una parte se da la opción de borrar una a una las estancias que se crean convenientes, haciendo uso de la función *confirmar()* implementada en un script de *JS* creada dentro del *head* de *diseño.php*, en la cual se comprueba si de verdad el usuario quiere llevar a cabo dicha acción. Si dicha condición es *true*, es decir, es aceptada por el usuario, entonces se realiza una referencia al archivo *borrar_estancia.php* en la que se le pasa el identificador de dicha estancia y se procede definitivamente a su borrado. En el caso contrario en el que la condición se evalúe a *false*, no habrá ningún cambio.

```
<a id= "borrar" href="#" onclick='confirmar(<?php echo $row_estan["ID_prueba"]; ?>)'>
Eliminar <?php echo $row_estan["Nombre"]; ?></a></td></tr> <!-- preguntamos si esta
seguro el usuario de que quiere realizar dicho borrado-->
```

Fragmentos de código 4: *HTML* para el borrado de una estancia

```
<script language="JavaScript">
function confirmar (Nombre){ //funcion para comprobar si de verdad se desea eliminar una estancia
    if(confirm("¿Estás seguro que deseas eliminar la estancia " + Nombre + " ?")) {
        document.location.href= 'borrar_estancia.php?nombre='+Nombre;
    }
}
</script>
```

Fragmentos de código 5: *Script* de *JS* para el borrado de una estancia

```
if(isset($_GET['nombre'])){
    $Nombre=$_GET['nombre'];
    $consulta=("DELETE FROM estancia WHERE ID_prueba=" . $Nombre. " ");
    $eliminar_cont=mysql_query($consulta,$link) or die("Error: ".mysql_error());
    header("location:diseño.php");
}
```

Fragmentos de código 6: *PHP* para el borrado de una estancia mediante *DELETE*

⁸ *SELECT* [26]: Sentencia SQL que nos permite consultar los datos de una tabla almacenados en la base de datos.

Por la otra, se utiliza un procedimiento similar al de borrar las estancias puesto que se le da la opción al usuario de borrar cada uno de los elementos por separado usando la función `confirmar_elemento()` y el archivo `borrar_elemento.php`. El procedimiento seguido es prácticamente idéntico a los fragmentos de código mostrados anteriormente en concreto en *Fragmentos de código 4, 5 y 6*, que aunque sean los de borrar estancias son casi iguales, cambiando solamente lo que es el *DELETE* y la función de *JS*.

5.4.4. Implementación de la monitorización y el control de la vivienda

Para hacer posible la monitorización y el control de la vivienda por parte del usuario, fue necesaria la creación de un botón que apareciese en todas y cada una de las ventanas de la aplicación, situado en la parte central y con el nombre “Monitorizar y controlar la vivienda”. Dicho botón tiene la función de redireccionar a un usuario ya sea Administrador, Controlador o Supervisor al archivo `control.php`, archivo desde el cual solo los administradores pueden acceder a cualquiera de los apartados de la web mediante otros botones y enlaces relacionales y en el que también se le impide a los supervisores el acceso a la parte del control, mostrándoles solo lo que es el estado de los elementos en cada momento, es decir, el acceso a lo que es la monitorización.

Si una persona que no es usuario intenta acceder al archivo `control.php`, se le mostrará un mensaje indicándole que debe iniciar sesión como administrador, supervisor o controlador para poder acceder, monitorizar y controlar los elementos.

5.4.4.1. Control.php (Monitorización)

En este archivo primeramente se realiza la conexión con la base de datos **vivienda**, para poder realizar cualquier transacción. Se le consulta a la base de datos de que tipo es el usuario registrado, independientemente del tipo al que pertenezca dicho usuario, el botón “Monitorizar y controlar la vivienda” queda seleccionado, de manera que el usuario sepa en cada momento en que opción de la web se encuentra, dándole la posibilidad al Administrador de acceder al resto de las ventanas, tanto a la de “Diseñar la vivienda” como a la de “Registrar, modificar o borrar usuarios”, y desactivándole al Controlador y Supervisor el acceso a estas otras ventanas. Además también se le da la opción a cualquiera de los usuarios de cerrar sesión cuando lo desee llamando al archivo `salir.php`, desde el cual se destruye la sesión iniciada y se redirige al usuario a la página de inicio de sesión.

Seguidamente se realiza la impresión del estado de los valores actuales de cada elemento haciendo tres consultas a la base de datos, una para obtener todos los datos de las tablas **elemento** y **elementos_gráficos** cerciorándonos de que el identificador del elemento sea el mismo que el del elemento gráfico, otra para obtener el nombre de las estancias sin repeticiones para facilitarle el trabajo a los usuarios y la última para obtener la ubicación de cada una de las estancias. Una vez realizadas dichas consultas, se realizan varios *if* para comprobar cada elemento a que tipo pertenece y dependiendo del tipo se imprime el valor que se haya en la



base de datos en ese momento y la imagen asignada al mismo. Esta parte compone lo que es la monitorización.

```
if($tipo == "On/Off"){
    echo "<td id=tipo>".$myOnOff. "&nbsp; &nbsp;"; //imprimimos el tipo asociado a cada elemento
?>
</td>
<?php
    printf("</tr>");
}
else if($tipo == "Discretizado"){
    echo "<td id=tipo>".$myDis. "&nbsp; &nbsp;"; //imprimimos el tipo asociado a cada elemento
?>
</td>
<?php
    printf("</tr>");
}
else if($tipo == "Nivel"){
    echo "<td id=tipo>".$myNivel. "&nbsp; &nbsp;"; //imprimimos el tipo asociado a cada elemento
?>
</td>
<?php
    printf("</tr>");
}
}
```

Fragmentos de código 7: Impresión del tipo de elemento e imagen asociada

- Si es del tipo “On/Off”, su valor asignado podrá ser “On” u “Off”, dependiendo del valor que se encuentre seleccionado en la base de datos en ese momento.
- Si es del tipo “Discretizado”, su valor asignado podrá ser “Media”, “Alta”, “Baja” u “Off”, dependiendo del valor que se encuentre seleccionado en la base de datos en ese momento.
- Si es del tipo “Nivel”, su valor asignado podrá ser “Subir” o “Bajar”, dependiendo del valor que se encuentre seleccionado en la base de datos en ese momento.

5.4.4.2. *Control.php (Control)*

Para el apartado del control, situado a continuación de la monitorización, se realiza la comprobación de que el usuario *logueado*⁹ no sea del tipo Supervisor, para que el Administrador y el Controlador sean los únicos que puedan controlar los elementos de la vivienda.

Para poder realizar el control de cada elemento por separado fue necesaria la utilización, de dos bucles *while* uno para recorrer las estancias una a una y otro, ubicado dentro del primero, para recorrer cada uno de los elementos que contiene la estancia recorrida por el primero de los bucles. Dentro del segundo *while* introducimos tres comparaciones similares en las que dependiendo del tipo al que pertenece cada elemento se imprimen unos desplegables u otros utilizando formularios *HTML*, y las imágenes correspondientes a cada uno de los valores impresos. En la siguiente captura se puede observar un ejemplo de una de las tres comparaciones realizadas.

⁹ *Logueado* [27]: termino spanglish que significa que un usuario se ha registrado en una página y ha iniciado sesión en la misma.

```

if($tipo == "On/Off"){
    printf("<td id=tipo>");
}
?>
<form method="post" class="valores_intro" action="">
<select class="despOnOff" name="despOnOff"<?php echo $identificador; ?>" id=""<?php echo $
identificador; ?>">
<?php // Obtener los valores del tipo OnOff
$sql2 = "SHOW columns FROM elementos_gráficos LIKE 'OnOff'";
$con consulta = mysql_query ($sql2, $link);
$row2 = mysql_fetch_array ($consulta);

$lis = strstr ($row2[1], "("); //encontramos la primera aparición de un string
$lis = ltrim ($lis, "("); //retiramos espacios en blanco del inicio de un string
$lis = rtrim ($lis, ")"); //retiramos los espacios en blanco del final de un string
$lista = explode ("", $lis); //dividimos un string en varios
strings

// Mostramos cada valor en un OPTION
for ($i=0; $i<count($lista); $i++){
    $OnOff = trim ($lista[$i], "'");
}
?>
<option <?php if ($myOnOff == $OnOff) echo "selected";?> >
<?php echo $OnOff;?>
</option>
<?php
} //fin del for
?>
</select></br></br>
</form>
<?php
printf("</td>");
?>
<td id="imagen"> </td>
<?php
printf("</tr>");
}else if($tipo == "Discretizado"){

```

Fragmentos de código 8: Impresión del desplegable del tipo "On/Off" y su imagen

Tras intentar varios métodos para conseguir el identificador del *select* pulsado por el usuario, tuvimos que utilizar *JS* y algunas funciones de *jQuery*. Esta elección nos facilitó bastante el hecho de obtener el nuevo identificador y el nuevo valor del elemento, puesto que automáticamente se extrae el identificador asociado al *select* seleccionado, mediante la función `$(this).attr()` e inmediatamente después se obtiene el valor que se acaba de seleccionar, con la función `$(this).val()`. La utilización de dichas funciones se muestra en el siguiente fragmento de código. Una vez obtenido el valor y el identificador del desplegable se realiza una referencia, con dichos datos pasados como parámetros, a otro archivo *PHP* para poder llevar a cabo la actualización de la base de datos, estos archivos son los denominados: *actualizarOnOff.php*, *actualizarDiscretizado.php* y *actualizarNivel.php*.

```

<!-- utilizamos jQuery y Javascript para poder obtener el identificador del select que deseamos
modificar para el desplegable On/Off-->
<script type="text/javascript">
//alert('prueba');
$(".despOnOff").on('change',function(){ //si se selecciona alguna opcion del desplegable OnOff
var currentId = $(this).attr('id'); //extraemos el id asociado a dicho select
var valor = $.trim($(this).val()); //extraemos el valor que ha sido pulsado en el select, el trim
elimina los caracteres en blanco
window.location.href="actualizarOnOff.php?val="+valor+"&id="+currentId; //hacemos una referencia a
un archivo php para poder realizar la actualización de la base de datos en php i obtenemos los dos
valores necesarios el valor y el identificador
});
</script>

```

Fragmentos de código 9: Utilización de *JS* y *jQuery* para la obtención de los valores

Los archivos de actualización nombrados en el párrafo anterior son muy similares entre sí, ya que todos ellos realizan una conexión a la base de datos **vivienda** y obtienen mediante el método `$_GET` de *PHP* el identificador y el valor del desplegable seleccionado, cuyos valores serán introducidos con una transacción del tipo *UPDATE* en la tabla **elementos_gráficos**, actualizando de esta manera la fila correspondiente.



5.4.4.3. *Control.php (Botones generales)*

Por último en el archivo *control.php* también se imprimen unos botones nuevos, denominados “Botones generales” que aunque en un principio no se tenían en mente, a medida que íbamos realizando la aplicación comprobamos que podían ser de gran utilidad. Estos son dos, y están implementados como botones en *HTML* tal y como se muestra en la siguiente captura, y su función es apagar todos los aparatos cuyo tipo es “On/Off” o “Discretizado” y bajar todas las persianas a la vez. Para ello desde este mismo archivo se realizan dos referencias a otros archivos desde los cuales se lleva a cabo la actualización de todos los elementos, estos archivos son *DesactivarElementosOff.php* y *DesactivarElementosNivel.php*.

```
<br><br><div id="botones_valor">BOTONES GENERALES<br><br>
<div id="botones_generales">
  <button onClick="window.location.href='desactivarElementosOff.php'">
    Apagado general <br>
  </button>
  <button onClick="window.location.href='desactivarElementosBajar.php'">
    Bajado general <br>
  </button><br><br>
</div><!--cerramos el div de botones_generales -->
</div><!-- cerramos el div botones_valor-->
```

Fragmentos de código 10: Implementación botones generales en *HTML*

5.4.5. *Implementación del sistema de registro de usuarios.*

Existe un tercer botón situado en la parte superior derecha, que hace posible el registro de nuevos usuarios, y la modificación y el borrado de los ya existentes. Dicho botón tiene la función de re direccionar a un usuario que sea Administrador al archivo *formulario_registro.php*, archivo desde el cual solo los administradores pueden acceder a cualquiera de los apartados de la web mediante otros botones y enlaces relacionales.

5.4.5.1. *Formulario_registro.php (registro de usuarios nuevos)*

En este archivo se creó un formulario en *HTML*, para el registro de nuevos usuarios. El formulario contiene dos campos de tipo *text* para la introducción de un nombre y de un correo, dos campos de tipo *password* en el que se introduce dos veces la contraseña para comprobar que el usuario no olvida fácilmente la misma, y uno de tipo *select* en el que se puede seleccionar el tipo de usuario al que pertenecerá el que esté siendo registrado en ese mismo instante.

Tras la impresión, el relleno y el envío del formulario nombrado anteriormente se comprueban que los valores introducidos en cada campo sean correctos. Para ello se obtienen mediante el método *\$_POST* los valores introducidos y se comprueba que ninguno de ellos este vacío, ya que es necesaria la introducción de todos los datos personales.

En el caso de que todos los campos sean distintos a vacío entonces comprobamos que el nombre de usuario no exceda de seis caracteres y que además no exista un nombre igual en la base de datos, lo que significaría que dicho usuario ya existe y hay que introducir uno diferente, ya que no puede haber dos usuarios con el mismo nombre. Se verifica que el correo introducido tenga un formato coherente, para ello se utiliza la función *filter_var (\$email,*

FILTER_VALIDATE_EMAIL) la cual verifica dicho correo electrónico. Y por último se comprueba que las contraseñas introducidas sean idénticas.

Si todos los datos son introducidos correctamente y hacen que todas las condiciones explicadas en el párrafo anterior sean ciertas, entonces se realiza un *INSERT* en la tabla **usuarios**, con los valores recogidos por el formulario y se le avisa al usuario mediante una advertencia de color verde situada dentro del mismo formulario. En el caso contrario se imprimen los errores encontrados debajo de cada uno de los campos erróneos, indicando cual es el error y dándole la opción de volver a introducir otros datos que sean correctos.

```
<?php
$comprobar_user= mysql_query("SELECT user FROM usuarios WHERE user='$user'");
if( (mysql_num_rows ($comprobar_user) ==0) && (strlen($user) <= 6) && ($password == $
conf_password) && (filter_var($email,FILTER_VALIDATE_EMAIL ) ) ){
    mysql_query("INSERT INTO usuarios (user,password,email,TipoUsuario) values ('$user','$
password','$email','$TipoUsuario')", $link);
    echo "<div id=correcto><br>El usuario $user ha sido registrado correctamente<div>";
} //fin del if que comprueba que todos los campos sean correctos
?>
```

Fragmentos de código 11: Comprobación de los campos e introducción en la BD de usuarios nuevos

5.4.5.2. *Formulario_registro.php (borrado de usuarios existentes)*

Para el borrado de los usuarios se ha creado un pequeño formulario, situado al lado del formulario de modificación de usuarios, con un único desplegable del tipo *select* en el que aparecen los nombres de todos los usuarios almacenados en la tabla **usuarios**. Si dicho formulario es enviado mediante el botón del tipo *submit* entonces aparece una advertencia realizada con *JS* y *jQuery* que imprime un texto utilizando la función *confirm*, el código que implementa dicha advertencia es el mostrado en la *Ilustración 12*. En el caso de que sea aceptada, se ejecuta el código ubicado en el archivo *borrar_user.php* y se realiza el correspondiente *DELETE*, en el caso contrario, en el que se rechaza la advertencia, no sucede nada y por lo tanto no se produce el borrado.

```
<script type="text/javascript"> // Al pulsar el botón cojemos el vamos que haya seleccionado en el
select en ese momento y le preguntamos al usuario si está seguro que quiere realizar dicho
borrado
$("#boton_borrar").on('click', function(){
    var valor = $.trim($("#borrar").val());
    if(confirm("¿Estás seguro que deseas eliminar el usuario " + valor +"?")) {
        window.history.pushState(valor, "Borrar usuario", "borrar_user.php?val="+valor);
    }
})
</script>
```

Fragmentos de código 12: Borrado de usuarios

5.4.5.3. *Formulario_registro.php (Modificación de usuarios existentes)*

Si se desea modificar los datos de un usuario ya existente se posee de un pequeño formulario situado en medio de los dos formularios explicados en los puntos anteriores, el 5.3.5.1 y el 5.3.5.2, el cual contiene un único desplegable del tipo *select* en el que aparecen los nombres de todos los usuarios almacenados en la base de datos.



Tras la selección y el envío de dicho formulario, la acción del mismo es imprimir otro formulario en el que aparecen los datos del usuario a modificar. Se podrán cambiar casi todos los campos, decimos casi todos porque el campo que contiene el identificador del usuario será fijo y no podrá ser modificado por el Administrador, por lo tanto es un campo *readonly*, desde el cual el usuario puede leer el contenido pero no puede modificarlo. Todos los demás campos sí que podrán ser modificados y además contendrán las mismas comprobaciones que se realizan cuando se registra un nuevo usuario.

```
<label class="id">
<span>Identificador</span>
<input id="id_des" name="id" value="<?php if (isset($id)) echo $id; ?>" type="text"
placeholder="Identificador" readonly><br> <!-- en los valores mantenemos los datos que el
usuario a introducido-->
<br></label>
```

Fragmentos de código 13: Campo identificador del tipo *readonly*

6. CONCLUSIONES

Cerrando el texto que compone el presente proyecto, concluimos que finalmente se han podido alcanzar de manera satisfactoria tanto los dos objetivos generales, como los específicos, marcados al inicio del mismo.

Se ha podido alcanzar y obtener, el primero de los objetivos generales, el diseño y la implementación de una aplicación web para un sistema domótico basado en el estándar KNX, el cual permite la monitorización y el control de una forma remota y segura, a través de cualquier dispositivo conectado a internet que tenga instalado un navegador web, con un interfaz dinámico, intuitivo y fácil de entender, además de haber alcanzado los diez objetivos específicos que lo componen.

Hemos conseguido que se puedan definir, diseñar y eliminar las diferentes estancias y elementos domóticos que componen una vivienda, haciendo posible el asignado de tipos a cada uno de estos elementos. Además también se ha obtenido el dibujo y el control del estado de cada uno de los elementos, ofreciéndole de esta manera al usuario una mayor comodidad a la hora de visualizar o modificar el estado de estos. Así mismo en su visualización, se le han asignado imágenes representativas que se van modificando dependiendo de su estado. La aplicación, además, permite la creación, modificación y borrado de usuarios, así como se le han asignado a cada tipo de usuario diferentes acciones.

Con lo referente al segundo objetivo general, podemos decir que también ha sido logrado en su totalidad, puesto que de una manera autodidactica hemos conseguido aprender nuevos lenguajes de programación, como son *PHP* y en menor medida *Java Script* junto con la utilización de las funciones de *jQuery*. Además de haber aplicado varios de los conocimientos aprendidos durante estos años de carrera, como han podido ser la creación de diagramas de clases UML, diagramas de casos de uso, diseño de la base de datos, utilización del servidor *XAMPP*, utilización de *HTML* y *CSS*, entre muchos otros.

Conseguir todos los objetivos no ha sido nada fácil puesto que en muchos de los casos nos hemos quedado inmobilizados y no podíamos continuar, debido a los problemas que nos han ido surgiendo, entre los que destacamos dos de ellos: el primero fue el de no poder actualizar individualmente cada uno de los estados de cada elemento, puesto que cada vez que actualizábamos uno se actualizaban todos a la vez, y finalmente tuvimos que recurrir al uso de las funciones de *jQuery* de *JS* para poder obtener el identificador de cada elemento por separado, al pulsar el desplegable correspondiente, y de esta manera poder actualizar de manera individual el estado de cada elemento. El segundo fue, el de no poder imprimir el mensaje de confirmación para el borrado de los usuarios, el cual también fue resuelto mediante la utilización de las funciones de *jQuery* de *JS*, en concreto la clave estuvo en utilizar la función *window.history.pushState()*, puesto que la utilizada hasta el momento, *window.location.href* no realizaba la redirección y por lo tanto no llevaba a cabo el borrado correctamente.



Con lo referente a lo que es el mundo de la domótica podemos decir que por estar en los tiempos que estamos, aún no existen grandes cantidades de ofertas *software* para el control de los elementos dómoticos de una vivienda, a través del uso de cualquiera de los dispositivos que actualmente son el todo, como los ordenadores, las *Tablets* o los *Smartphones*. Por todo ello deducimos que hay bastantes oportunidades de negocio en este mundo, puesto que es algo que aún se encuentra en un punto innovador y que cada vez irá a más, ya que es el presente y el futuro de los hogares, y los edificios públicos.

En efecto, la domótica permite que nuestra vivienda contenga cada vez más tecnología y estilo, con un mayor control de lo que sucede en el hogar en cada momento, un mayor ahorro energético gracias al hecho de poder controlar el encendido y el apagado de las luces desde cualquier ubicación o poder disminuir la temperatura del hogar en un nivel de mantenimiento cuando no se encuentre nadie en la vivienda. Además de la comodidad que nuestra aplicación web aporta gracias a la posibilidad de administrar todo el hogar de forma remota desde cualquier dispositivo, y la seguridad que ello conlleva, ya que permite simular movimiento en la casa en el caso de que los habitantes se encuentren de vacaciones, evitando de esta manera posibles allanamientos.

De todo lo dicho hasta el momento puede concluirse que la realización de este presente proyecto ha sido una experiencia enriquecedora en muchos de los sentidos, puesto que hemos aprendido algunos aspectos informáticos y otros más técnicos, haciendo hincapié en un mundo bastante desconocido e interesante en la actualidad y con muchas posibilidades en el futuro, tanto gracias a la domótica como a las tecnologías y lenguajes aprendidos durante su realización.

7. BIBLIOGRAFÍA

- [1] Huidobro, José M. y Millán, Ramón J. (2010). Manual de domótica. España: Creaciones copyright.
- [2] Tablet PC. Ingeniatic. Disponible en: <<http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/607-tablet-pc>>. Fecha de consulta agosto de 2014.
- [3] Ilyas, Mohammed y Ahson, Syed. (2006). *Smartphones*. USA, chicago: International Engineering Consortium.
- [4] ¿Qué es KNX? Disponible en: <<http://www.knx.org/es/knx/que-es-knx/>>. Fecha de consulta agosto de 2014.
- [5] KNX. Wikipedia. Disponible en: <<http://es.wikipedia.org/wiki/KNX>>. Fecha de consulta agosto de 2014.
- [6] XAMPP. Apache Friends. Disponible en: <<https://www.apachefriends.org/es/index.html>>. Fecha de consulta julio de 2014.
- [7] ¿Qué es PHP?. The PHP Group. Disponible en: <<http://php.net/manual/es/intro-what-is.php>>. Fecha de consulta agosto de 2014.
- [8] Sánchez, Miguel A. (2012). *JavaScript*. Antequera, Málaga: INNOVA 2001.
- [9] jQuery, encuentra el significado al diseño web y la programación. Estudio surestao. Disponible en: <<http://www.estudiosurestao.com/diccionario/jquery>>. Fecha de consulta agosto de 2014.
- [10] *The world's most popular open source database. Oracle corporation and/or its affiliates*. Disponible en: <<http://www.mysql.com/>>. Fecha de consulta mayo de 2014.
- [11] El lenguaje HTML. IES Gaviota. Disponible en: <<http://www.juntadeandalucia.es/averroes/~04000134/informatica/html.html>>. Fecha de consulta agosto de 2014.
- [12] Guía Breve de CSS. W3C España. Disponible en: <<http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>>. Fecha de consulta agosto de 2014.
- [13] What are CSS Media Queries and how to implement them. CSS Media Queries. Disponible en: <<http://cssmediaqueries.com/what-are-css-media-queries.html>>. Fecha de consulta agosto de 2014.



- [14] Sublime Text. Guillermo López-Anglada & The Sublime Text Community. Disponible en: <<http://www.sublimetext.com/>>. Fecha de consulta mayo de 2014.
- [15] Pencil Project. Evolus. Disponible en: <<http://pencil.evolus.vn/>>. Fecha de consulta junio de 2014.
- [16] Indentación. Wikipedia. Disponible en: <<http://es.wikipedia.org/wiki/Indentaci%C3%B3n>>. Fecha de consulta agosto de 2014.
- [17] Departamento de sistemas informáticos y computación. Base de datos y sistemas de información. Unidad Didáctica 3: Diseño de Bases de Datos Relacionales, Parte 2: Diseño Conceptual. Universidad Politécnica de Valencia. Noviembre de 2012. Fecha de consulta mayo de 2014.
- [18] StarUML. MKLab 2014. Disponible en: <<http://staruml.io/>>. Fecha de consulta mayo de 2014.
- [19] Introduction to InnoDB. MySQL Server. Disponible en: <<http://dev.mysql.com/doc/refman/5.5/en/innodb-introduction.html>>. Fecha de consulta mayo de 2014.
- [20] Departamento de sistemas informáticos y computación. Base de datos y sistemas de información. Unidad Didáctica 4: Sistema de Gestión de Bases de Datos. Universidad Politécnica de Valencia. Noviembre de 2012. Fecha de consulta mayo de 2014.
- [21] The MyISAM Storage Engine. MySQL Server. Disponible en: <<http://dev.mysql.com/doc/refman/5.1/en/myisam-storage-engine.html>>. Fecha de consulta mayo de 2014.
- [22] Diagramas de casos de uso de UML: Instrucciones. *Microsoft Developer Network*. Disponible en: <<http://msdn.microsoft.com/es-es/library/dd409432.aspx>>. Fecha de consulta agosto de 2014.
- [23] Diagramas de casos de uso. Jesús Cáceres Tello. Disponible en: <<http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>>. Fecha de consulta agosto de 2014.
- [24] Arquitectura de las aplicaciones web. Instituto Tecnológico de Veracruz. Disponible en: <<http://www.prograweb.com.mx/pweb/0201arquiAplicaweb.html>>. Fecha de consulta agosto de 2014.
- [25] HTML <select> Tag. Refsnes Data. Disponible en: <http://www.w3schools.com/tags/tag_select.asp>. Fecha de consulta agosto de 2014.
- [26] Select Syntax. MySQL Server. Disponible en: <<http://dev.mysql.com/doc/refman/5.1/en/select.html>>. Fecha de consulta agosto de 2014.

- [27] Definición de login. Leandro Alegsa. Disponible en: <<http://www.alegsa.com.ar/Dic/login.php>>. Fecha de consulta agosto de 2014.
- [28] Manual de *PHP*. Philip Olson. Disponible en: <<http://php.net/manual/es/>>. Fecha de consulta mayo de 2014.
- [29] Fernández, Marco A. (2010). *Programación de bases de datos con MySQL y PHP*. Barcelona: Marcombo ediciones técnicas.
- [30] Desarrollo de sitios web con PHP y MySQL. José Mariano González Romano. Disponible en: <<http://www.lsi.us.es/cursos/cursophp/>>. Fecha de consulta mayo de 2014.
- [31] *HTML4 and HTML5 tutorial*. Refsnes Data. Disponible en: <<http://www.w3schools.com/html/>>. Fecha de consulta mayo de 2014.
- [32] *CSS tutorial*. Refsnes Data. Disponible en: <<http://www.w3schools.com/css/>>. Fecha de consulta junio de 2014.
- [33] Sistema de registro y login de usuarios *html-php-mysql* tutorial. Diazespina. Disponible en: <<http://www.taringa.net/posts/hazlo-tu-mismo/17512963/Sistema-de-registro-y-login-de-usuarios-html-php-mysql-tuto.html>>. Fecha de consulta junio de 2014.
- [34] Aprende a programar en php ya. Salvador López. Disponible en: <<http://aprendeaprogramarenphpya.com/blog/como-consultar-dos-o-mas-tablas-facilmente-en-php-con-mysql.html>>. Fecha de consulta junio de 2014.
- [35] Borrado de un registro con PHP. Rubén Álvarez. Disponible en: <<http://www.desarrolloweb.com/articulos/326.php>>. Fecha de consulta junio de 2014.
- [36] Obtener un múltiple SELECT en PHP. Disponible en: <<http://trucosinformaticos.wordpress.com/2010/10/05/obtener-un-multiple-select-en-php/>>. Fecha de consulta julio de 2014.
- [37] jQuery: obtener el “value” de un select. Jorge Maiden. Disponible en: <<http://www.bufa.es/jquery-obtener-value-select/>>. Fecha de consulta julio de 2014.



ANEXO A: INSTALACIÓN Y PUESTA EN MARCHA DE XAMPP

Para poder utilizar nuestra aplicación será necesaria la correcta instalación de *XAMPP*, servidor que incluye tanto el lenguaje utilizado, *PHP*, como la base de datos empleada, *MySQL*, para ello es necesario llevar a cabo los siguientes pasos:

1. Acceder a la URL <https://www.apachefriends.org/download.html>
2. Descargar aquella versión que se adapte a las características de nuestro ordenador. En nuestro caso elegimos la versión 1-8.3 / PHP 5.5.11 para Windows.
3. Iniciar la instalación siguiendo los pasos que nos indique el instalador.
4. Abrir la aplicación instalada.
5. Pulsar en “*Start*” en los módulos “*Apache*” y “*MySQL*”.
6. Comprobar que *XAMPP* funciona correctamente abriendo el navegador y accediendo a la URL <http://localhost/xampp/>

Si el *XAMPP* nos da un problema similar al mostrado en el siguiente recuadro, con el *Apache*, se deberá solucionar siguiendo los pasos que se muestran a continuación:

```
[Apache] Problem detected!  
[Apache] Port 80 in use by "Unable to open process" with PID 4!  
[Apache] Apache WILL NOT start without the configured ports free!  
[Apache] You need to uninstall/disable/reconfigure the blocking application  
[Apache] or reconfigure Apache and the Control Panel to listen on a different port
```

1. Abrimos la consola *CMD*
2. Escribimos el comando “*net stop was /y*”
3. Pulsamos las teclas *Win + r*
4. Escribimos *services.msc* y aceptamos
5. Buscamos “*Servicio de publicación World Wide Web*”
6. Desactivamos dicho servicio
7. Volvemos a arrancar el *Apache*

Por otro lado también destacamos que si se da el caso de que al iniciar el computador iniciamos primero la aplicación de *Skype*¹⁰, antes que el *XAMPP*, lo más seguro es que al iniciar el *XAMPP* empiecen a aparecernos errores en la conexión a los puertos, puesto que *Skype* y *XAMPP* suelen utilizar el puerto 80.

Por lo que deberemos iniciar como primera opción el *XAMPP* y posteriormente arrancar el *Skype*, si es que se desea el uso de ambas aplicaciones a la vez.

¹⁰ *Skype: Software* que permite comunicaciones de texto, voz y video sobre Internet (*VoIP*).

ANEXO B: MANUAL DE USUARIO

El objetivo de este anexo es explicar el funcionamiento de la aplicación web, realizada para el control web de una vivienda domótica mediante el estándar KNX. La presente aplicación permite diseñar, controlar y monitorizar los elementos domóticos de una vivienda, así como registrar usuarios nuevos y modificar o borrar los ya existentes.

TIPOS DE USUARIO

El Sistema contempla tres tipos de usuarios diferentes en el que cada uno de ellos puede realizar unas funciones u otras.

- **Usuarios Administradores:** Son aquellos que navegan por todo el sistema y pueden diseñar, controlar o monitorizar una vivienda, así como realizar el registro de nuevos usuarios, y la modificación o el borrado de los ya existentes, es decir son los únicos que pueden realizar cualquiera de las acciones que nuestra aplicación ofrece.
- **Usuarios Controladores:** Son registrados por el Administrador y pueden acceder a la monitorización de los elementos y al control de los mismos, sin tener derechos para diseñar la vivienda, ni para crear nuevos usuarios.
- **Usuarios Supervisores:** Son registrados por el Administrador y pueden acceder solamente a la parte de la monitorización de los elementos, sin tener permiso para realizar ninguna de las acciones que la aplicación nos ofrece, ya que como su nombre indica son solo supervisores.

PANTALLA DE INICIO DE SESIÓN

En la página inicial del sistema aparece la cabecera de la aplicación, junto con su logotipo. Y en la parte central se muestra el formulario para acceder a dicha aplicación, en el que se pide el nombre de usuario y su contraseña.



The screenshot shows a web browser window with the address bar displaying 'localhost/PROYECTO/formulario_acceso.php'. The page has a green header with the title 'Control web de la vivienda' and a logo on the left. Below the header, there is a light green background with a decorative scroll pattern on the left. In the center, there is a login form titled 'Introduzca sus credenciales:'. The form contains two input fields: 'Usuario' (with 'User' entered) and 'Contraseña' (with 'Password' entered). Below the fields are two buttons: 'Iniciar sesión' and 'Cancelar'.

Ilustración 20: Página de inicio de sesión

Si se diese el caso de que uno de los dos valores introducidos fuese incorrecto, es decir que el usuario no estuviera en la base de datos, se mostraría un mensaje rojo advirtiéndole de que sus datos no son correctos y debe introducir otros que si lo sean, tal y como aparece remarcado en la *Ilustración 22*.

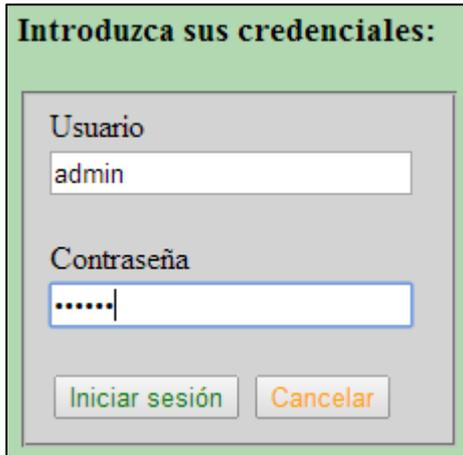


Ilustración 21: Formulario acceso vacío

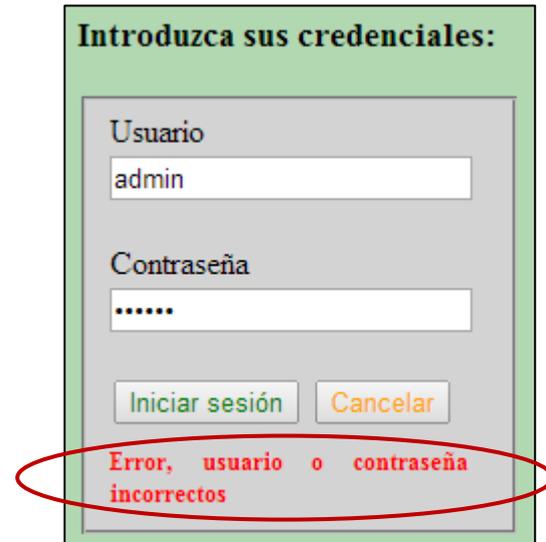


Ilustración 22: Formulario acceso con datos erróneos

Si por el contrario las credenciales introducidas son correctas y sí que existen en la base de datos, el sistema accede inmediatamente a lo que es la aplicación web, creando así una seguridad a la hora de acceder a la plataforma.

PANTALLA PRINCIPAL DEL ADMINISTRADOR (*diseño.php*)

Si el usuario que está accediendo es el administrador entonces su página principal será la de diseñar la vivienda, en la que podrá crear nuevas estancias y nuevos elementos para cada una de las estancias. Así como también podrá ver a qué tipo pertenece cada elemento y tendrá la opción de poder eliminar cualquier estancia o los elementos por separado, además de poder cerrar la sesión actual, mediante el enlace que aparece remarcado en la *Ilustración 23*.



Ilustración 23: Página principal del administrador (diseño de la vivienda I)

ELEMENTOS DE CADA ESTANCIA

Baño (Planta superior) Eliminar Baño			Cocina (Planta superior) Eliminar Cocina			Comedor (Planta superior) Eliminar Comedor		
Aire acondicionado	Discretizado	Eliminar	Horno	Discretizado	Eliminar	Aire acondicionado	Discretizado	Eliminar
Luz	On/Off	Eliminar	Luz	On/Off	Eliminar	Luz	On/Off	Eliminar
Persiana	Nivel	Eliminar	Persiana	Nivel	Eliminar	Persiana1	Nivel	Eliminar
			Tostadora	Discretizado	Eliminar	Persiana2	Nivel	Eliminar
			TV	On/Off	Eliminar	TV	On/Off	Eliminar
Dormitorio (Planta superior) Eliminar Dormitorio			Jardín (Exterior) Eliminar Jardín			Pasillo (Planta superior) Eliminar Pasillo		
Aire acondicionado	Discretizado	Eliminar	Bomba fuente	On/Off	Eliminar	Aire acondicionado	Discretizado	Eliminar
Luz	On/Off	Eliminar	Luces entrada	On/Off	Eliminar	Luz	On/Off	Eliminar
Persiana	Nivel	Eliminar	Luz general	On/Off	Eliminar			
TV	On/Off	Eliminar	Riego	On/Off	Eliminar			

Ilustración 24: Página principal del administrador (diseño de la vivienda II)

El primer formulario que contiene la página principal del administrador es el de crear una nueva estancia en el que el nombre debe ser único y obligatorio, ya que si tras intentar enviar dicho formulario al pulsar el botón **“Guardar estancia”** se verifica que existe uno idéntico a alguno de los que ya existan en la base de datos, se muestra un mensaje de error similar al remarcado en la *Ilustración 25*.

NUEVA ESTANCIA

Introduzca una estancia nueva:

La estancia Comedor ya existe, introduzca otra diferente

Introduzca la ubicación:

Ilustración 25: Formulario nueva estancia, con nombre repetido

En el caso de que el nombre introducido no exista, se almacenará correctamente en la base de datos y se mostrará un mensaje de color verde, como se muestra en la *Ilustración 26*, indicando su correcto almacenaje, además de ser visualizado en la parte inferior de la misma ventana en el apartado asignado para la impresión de las estancias y sus elementos, siendo esta mostrada sin ningún elemento, puesto que aún no se le ha designado ninguno, mostrado en la *Ilustración 27*.

NUEVA ESTANCIA

Introduzca una estancia nueva:
Baño1

Introduzca la ubicación:
Planta superior

Guardar estancia Cancelar

¡Gracias! Hemos almacenado la estancia Baño1 ubicada en Planta superior

Ilustración 26: Formulario nueva estancia con datos correctos

ELEMENTOS DE CADA ESTANCIA

Baño1
(Planta superior)

[Eliminar Baño1](#)

Ilustración 27: Visualización de la estancia introducida y su enlace de borrado

Tras introducir cualquier estancia se le da la opción al administrador de poder eliminarla mediante el enlace que aparece remarcado en la *Ilustración 27*, sin necesidad de que haya elementos almacenados en la misma, aunque si los hubiese, tampoco habría ningún problema en su eliminación puesto que al eliminar una estancia se eliminarían en cascada todos los elementos pertenecientes a dicha estancia. El usuario es avisado de la eliminación mediante una advertencia de JS, una advertencia similar a la mostrada en la *Ilustración 28*, la cual deberá ser aceptada si de verdad se quiere eliminar la estancia, aceptando de esta manera que desaparezca del apartado *Elementos de cada estancia*, y siendo también eliminada de la base de datos **vivienda**, o cancelada en el caso contrario.

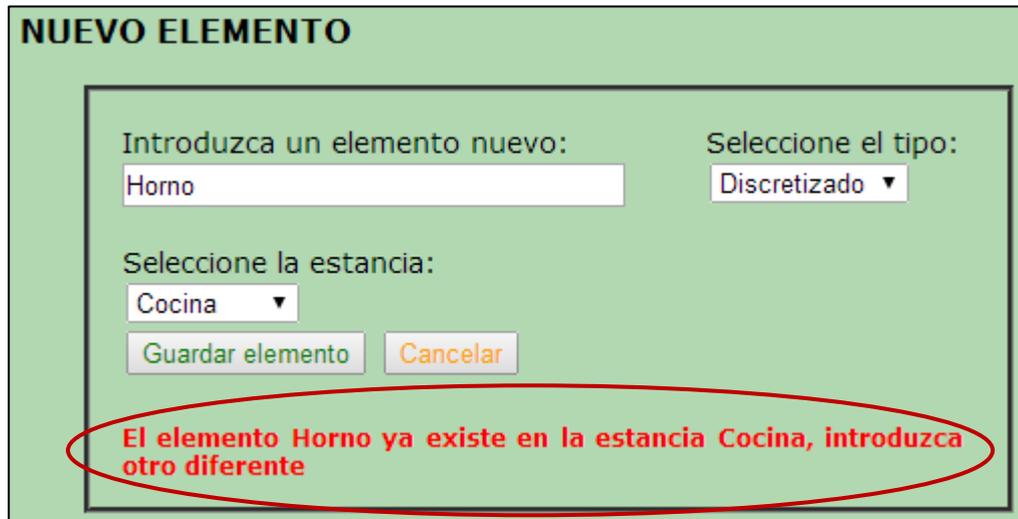
Mensaje de la página localhost:

¿Estás seguro que deseas eliminar la estancia 2?

Aceptar Cancelar

Ilustración 28: Advertencia de eliminación de la estancia seleccionada

El segundo formulario ubicado en el lateral derecho del designado para crear una nueva estancia, es el que permite la inserción de nuevos elementos en cada una de las estancias, en el que su nombre también debe ser único y obligatorio, pero único no en la base de datos sino para cada estancia, el en caso de que no lo sea se mostrará el mensaje remarcado en la *Ilustración 29*.

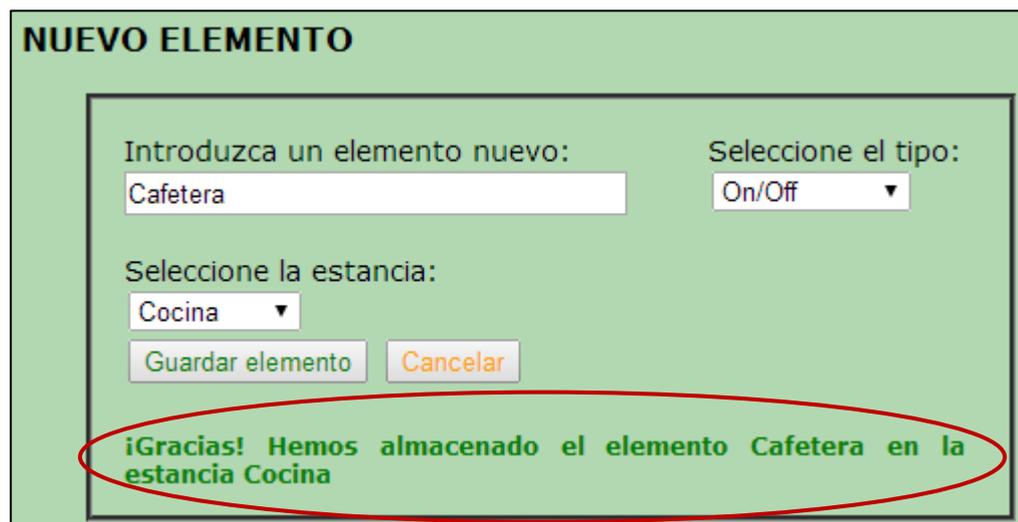


The screenshot shows a form titled "NUEVO ELEMENTO" with a light green background. It contains the following fields and controls:

- Introduzca un elemento nuevo:** A text input field containing "Horno".
- Seleccione el tipo:** A dropdown menu with "Discretizado" selected.
- Seleccione la estancia:** A dropdown menu with "Cocina" selected.
- Two buttons: "Guardar elemento" (green) and "Cancelar" (orange).
- A red oval highlights a message at the bottom: "El elemento Horno ya existe en la estancia Cocina, introduzca otro diferente".

Ilustración 29: Formulario nuevo elemento con datos erróneos

Si tras pulsar el botón “**Guardar elemento**” el elemento introducido no existe aún en la estancia seleccionada, se almacenará correctamente en la tabla **elemento** de la base de datos **vivienda**, y se le avisará al administrador mediante un mensaje de introducción satisfactoria, mostrado en la *Ilustración 30*.



The screenshot shows the same "NUEVO ELEMENTO" form, but with the following changes:

- Introduzca un elemento nuevo:** The text input field now contains "Cafetera".
- Seleccione el tipo:** The dropdown menu now shows "On/Off".
- Seleccione la estancia:** The dropdown menu still shows "Cocina".
- The "Guardar elemento" and "Cancelar" buttons remain.
- A red oval highlights a green message at the bottom: "¡Gracias! Hemos almacenado el elemento Cafetera en la estancia Cocina".

Ilustración 30: Formulario nuevo elemento con datos correctos

Tras ello se introducirá y se visualizará en la tabla de la estancia seleccionada, en el ejemplo mostrado se almacena en la tabla **cocina** de nuestra aplicación web junto con los demás elementos que ya habían sido introducidos con anterioridad, tal y como se puede ver en la *Ilustración 31*. Se introducen de manera ordenada, siguiendo un orden alfabético ascendente, empezando por la letra de menor peso en el alfabeto y acabando por la de mayor.

Cocina (Planta superior) Eliminar Cocina		
Cafetera	On/Off	Eliminar
Horno	Discretizado	Eliminar
Luz	On/Off	Eliminar
Persiana	Nivel	Eliminar
Tostadora	Discretizado	Eliminar
TV	On/Off	Eliminar

Ilustración 31: Visualización de nuevo elemento en la tabla de la estancia correspondiente

Si se desea eliminar el elemento introducido simplemente habrá que pulsar el enlace designado para dicha acción, situado a la misma altura que el nombre del elemento a eliminar. Como ejemplo borramos el elemento “Cafetera” remarcado en la *Ilustración 31* y aceptamos la advertencia que aparece posteriormente, la mostrada en la *Ilustración 32*. La aceptación del borrado hará desaparecer el elemento de la tabla quedándose con los demás (*Ilustración 33*).

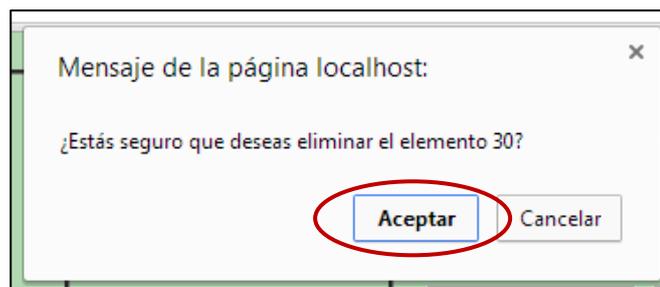


Ilustración 32: Advertencia de eliminación del elemento seleccionado

Cocina (Planta superior) Eliminar Cocina		
Horno	Discretizado	Eliminar
Luz	On/Off	Eliminar
Persiana	Nivel	Eliminar
Tostadora	Discretizado	Eliminar
TV	On/Off	Eliminar

Ilustración 33: Inexistencia del elemento eliminado de la tabla

PANTALLA PRINCIPAL DEL CONTROLADOR, Y PANTALLA SECUNDARIA DEL ADMINISTRADOR (*control.php*)

Si el usuario que está accediendo es controlador entonces su página principal será la de monitorización y control de la vivienda y si es administrador esta será su ventana secundaria. Tanto el controlador como el administrador, podrán modificar el estado de cada uno de los elementos además de poder visualizar el monitorizado de cada uno de ellos, así como también podrán realizar un desactivado general de los mismos. Además podrán cerrar sesión desde la pantalla en la que se encuentren, tal y como se remarca en la *Ilustración 34*. El administrador es el único que puede dirigirse a cualquiera de los restantes botones “Diseñar la vivienda” y “Registrar, modificar o borrar usuarios”.

The screenshot shows a web browser window with the URL localhost/PROYECTO/control.php. The page has a green header with the title 'Control web de la vivienda'. A session status bar indicates 'Sesión iniciada por: ana' with a 'Cerrar sesión' button circled in red. Below the header are three buttons: 'Diseñar la vivienda', 'Monitorizar y controlar la vivienda' (highlighted in yellow), and 'Registrar, modificar o borrar usuarios'. The main content area is titled 'VALOR ACTUAL DE LOS ELEMENTOS EN CADA ESTANCIA' and contains three control panels for 'Baño (Planta superior)', 'Cocina (Planta superior)', and 'Comedor (Planta superior)'. Each panel lists various elements like 'Aire acondicionado', 'Luz', 'Persiana', 'Horno', 'Tostadora', and 'TV' with their current status and control icons.

Ilustración 34: página principal del controlador y secundaria del administrador (control y monitorización I)

The screenshot shows the 'CONTROL DE LOS ELEMENTOS' section of the interface. It features four control panels: 'Dormitorio (Planta superior)', 'Jardín (Exterior)', 'Pasillo (Planta superior)', and 'Piscina (Exterior)'. Each panel lists various elements with their status and control icons. For example, the 'Dormitorio' panel includes 'Aire acondicionado', 'Luz', 'Persiana', and 'TV'. The 'Piscina' panel includes 'Luz volumen 0', 'Luz volumen 2', 'Bomba de agua', and 'Sistema depuración'.

Ilustración 35: página principal del controlador y secundaria del administrador (control y monitorización II)

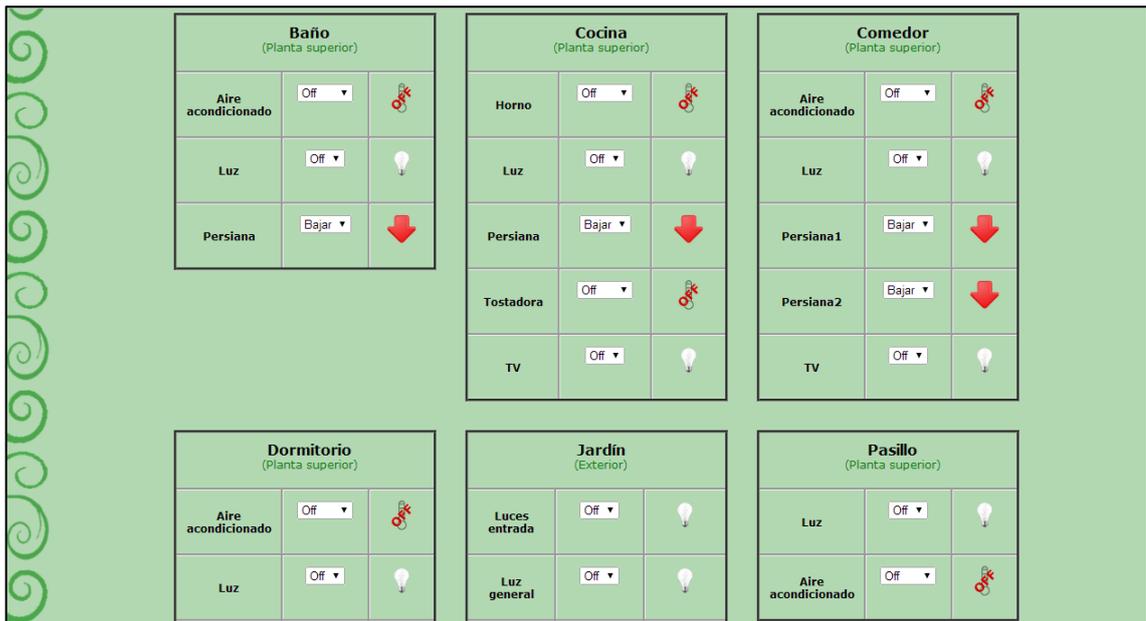


Ilustración 36: página principal del controlador y secundaria del administrador (control y monitorización III)

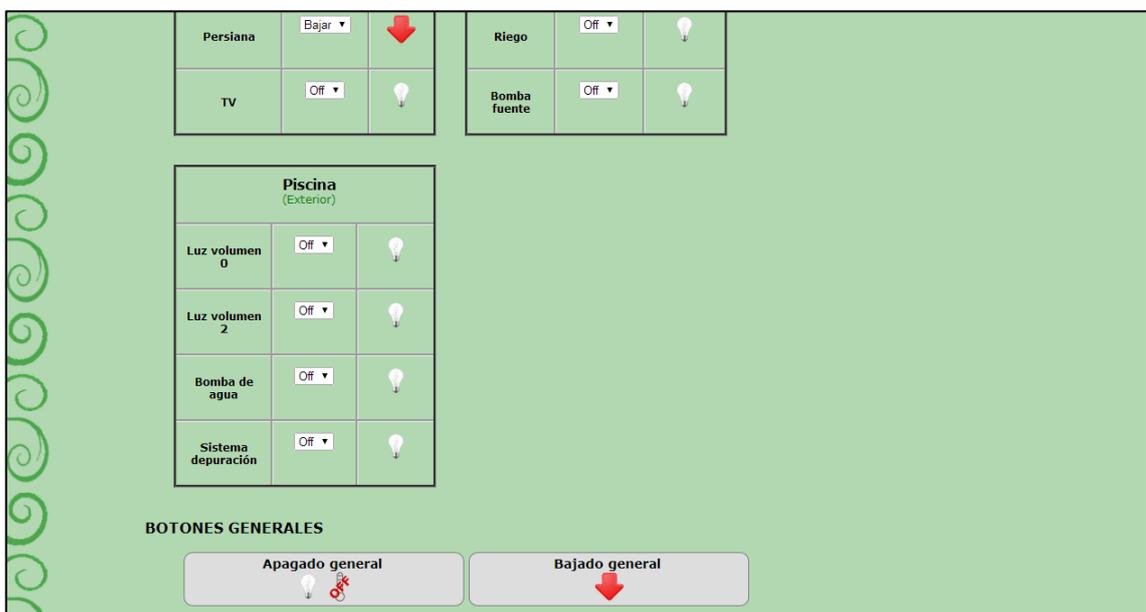


Ilustración 37: página principal del controlador y secundaria del administrador (control y monitorización IV)

Ahora bien, vistas las *Ilustraciones 34, 35, 36 y 37* describimos como debe modificar un usuario el estado de uno o varios elementos. Si se desea cambiar el estado de un elemento simplemente habrá que seleccionar una de las opciones de los desplegados situados en la misma fila que el elemento y el cambio de estado se realizará automáticamente indicándole al usuario cual ha sido el elemento actualizado mediante una alerta. No se podrán realizar varias actualizaciones a la vez, ya que de esta manera permite ir visualizando en cada momento el valor asociado y su representación gráfica, puesto que cuando se actualiza un valor, la ventana se recarga y el nuevo estado aparece actualizado tanto en el apartado del valor actual de los elementos, la monitorización, como en el del control de los mismos. En la *Ilustración 38* se puede ver un ejemplo de lo que es la selección de

un estado, mediante el desplegable que se encuentra remarcado y lo que es su posterior actualizado, remarcado en la *Ilustración 39*.



Ilustración 38: Desplegable para el cambio de estado de un elemento



Ilustración 39: Valor actualizado tras el cambio de estado en el control

Cabe decir que los controladores y los administradores también pueden apagar todos los elementos del tipo “Discretizado” y del tipo “On/Off”, a la vez pulsando el botón de “**Apagado general**”, el cual actualiza todos los elementos, asignándoles el valor “Off” y la imagen correspondiente. Además de poder bajar todas las persianas, toldos, etc. es decir todos los elementos del tipo “Nivel”, a la vez, con el botón de “**Bajado general**”, el cual actualiza todos los elementos de este tipo a “Bajar” y les asigna el icono correspondiente. En la *Ilustración 40* se muestran ambos botones y los iconos que se le asignarán a cada uno de los elementos, dependiendo del botón que sea pulsado.



Ilustración 40: Botones generales

PANTALLA PRINCIPAL Y ÚNICA DEL SUPERVISOR (*control.php*)

Todo usuario supervisor simplemente podrá visualizar cual es el estado actual de los elementos, sin tener acceso a ninguna otra parte de la aplicación web. Los estados visualizados serán los que un usuario administrador o controlador ha introducido con anterioridad desde su cuenta. Concretamente podrá visualizar una pantalla similar a la mostrada en la *Ilustración 41* y *42*, en la que tras poder ver todos los estados, se le avisa mediante un mensaje rojo de que es un usuario supervisor y por lo tanto únicamente puede observar lo que sucede, este mensaje es idéntico al que aparece remarcado en la *Ilustración 42*.

Para el usuario supervisor al igual que para el controlador, tal y como se ha descrito en el primer párrafo del apartado “PANTALLA PRINCIPAL DEL CONTROLADOR, Y PANTALLA SECUNDARIA DEL ADMINISTRADOR

Desarrollo de una aplicación domótica basada en el estándar KNX y su control web

(*control.php*)” los botones de “**Diseñar la vivienda**” y “**Registrar, modificar o borrar usuarios**” permanecerán desactivados para este tipo de usuarios. El supervisor también podrá cerrar la sesión actual siempre que lo desee, puesto que solamente tendrá que pulsar el enlace referencial situado debajo de su nombre, el cual aparece remarcado en la *Ilustración 41*.



Ilustración 41: Pantalla principal y única del supervisor (monitorización I)

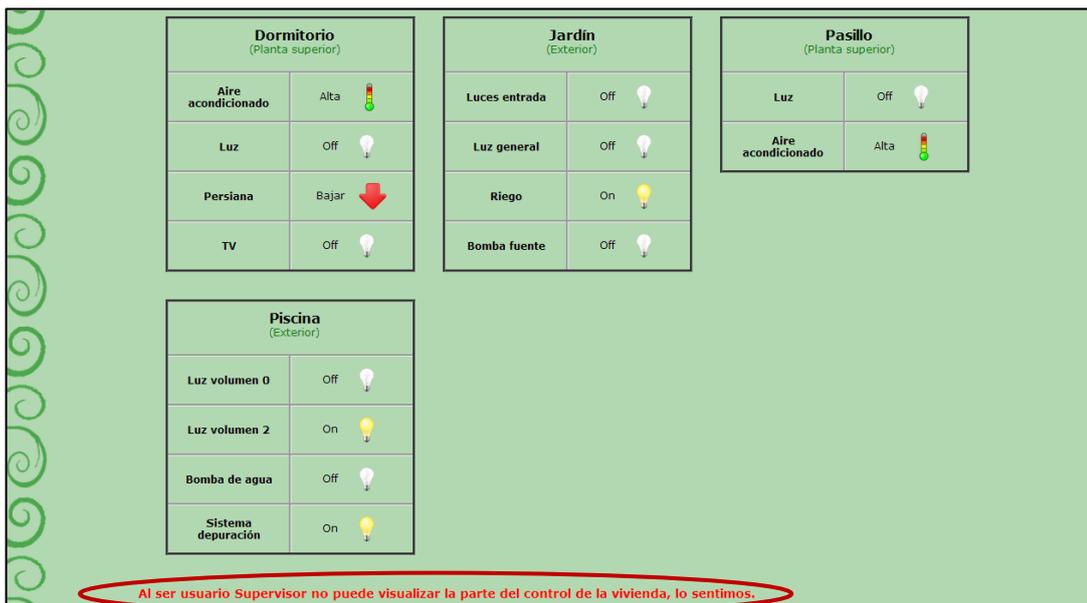


Ilustración 42: Pantalla principal y única del supervisor (monitorización II)

PANTALLA TERCIARIA DEL ADMINISTRADOR (Formulario_registro.php)

En la tercera de las pantallas en la que solo pueden tener acceso los administradores, se pueden realizar varias acciones, como la introducción de usuarios nuevos en la base de datos, para que puedan acceder a la aplicación con un nombre de usuario y una contraseña, la modificación de los datos y el borrado

de los mismos, de alguno de los usuarios que ya se encuentre registrado en nuestra aplicación.

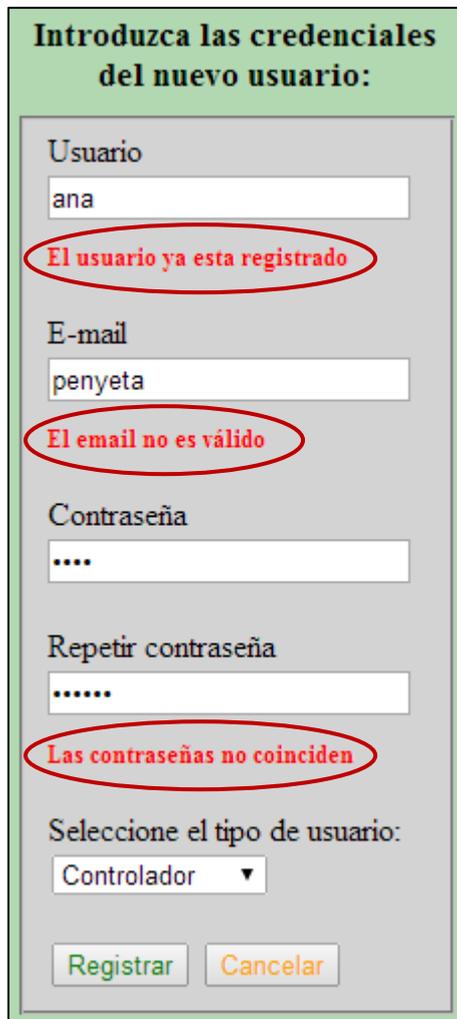
Ilustración 43: Pantalla terciaria del administrador (Registro, modificación y borrado de usuarios)

Esta ventana se divide en tres formularios, el primero de ellos es el de introducir las credenciales del nuevo usuario, para que dicha información conste en la base de datos de nuestra plataforma, y con el nombre de usuario y contraseña introducidos pueda posteriormente, iniciar sesión en nuestro sistema. El correo sirve para el hipotético caso en el que hubiese algún problema poder contactar con el usuario a través del mismo.

Si el administrador introduce datos erróneos en este formulario le aparecen mensajes de color rojo debajo de cada una de las casillas indicándole cual es el problema y porque no se puede llevar a cabo su almacenaje en la base de datos **vivienda**. Se pueden llegar a observar tres mensajes de error diferentes, descritos a continuación y los cuales se pueden visualizar en la *Ilustración 44*.

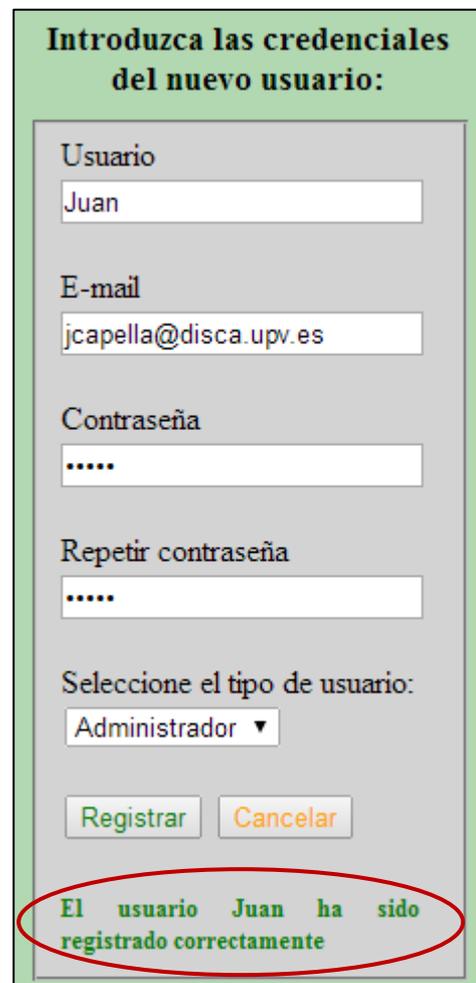
- El primero de los errores le está avisando al administrador de que el nombre de usuario que ha introducido ya se encuentra en la base de datos y por lo tanto al ser único no puede introducir otro idéntico, por lo que deberá cambiarlo por otro que aún no exista.
- El segundo de los errores indica que el *email* introducido en el campo designado para ello, no es correcto puesto que no tiene un formato similar al de un correo electrónico, no posee ni la arroba, ni el punto, ni ninguna característica que le haga parecerse a un *email*.
- El tercero de los errores que puede aparecerle es el de que las contraseñas no coinciden, se han introducido en los campos designados para la introducción de contraseñas caracteres diferentes, lo que indica que el administrador no recuerda con certeza cuál es la contraseña que desea asignarle al nuevo usuario.

Tras realizar los cambios pertinentes para que los datos introducidos puedan ser almacenados correctamente en nuestra base de datos, se efectuará el envío del formulario mediante el botón “**Registrar**” y tras ello se le avisará al administrador del correcto almacenaje con la impresión de un nuevo mensaje de color verde ubicado tras los botones del formulario, tal y como aparece remarcado en la *Ilustración 45*.



The screenshot shows a registration form titled "Introduzca las credenciales del nuevo usuario:". The form contains several input fields: "Usuario" (containing "ana"), "E-mail" (containing "penyeta"), "Contraseña" (containing "...."), and "Repetir contraseña" (containing "....."). A dropdown menu for "Seleccione el tipo de usuario:" is set to "Controlador". Three red ovals highlight error messages: "El usuario ya esta registrado" (circled around the Usuario field), "El email no es válido" (circled around the E-mail field), and "Las contraseñas no coinciden" (circled around the password fields). At the bottom, there are "Registrar" and "Cancelar" buttons.

Ilustración 44: Formulario registro nuevo usuario con datos erróneos



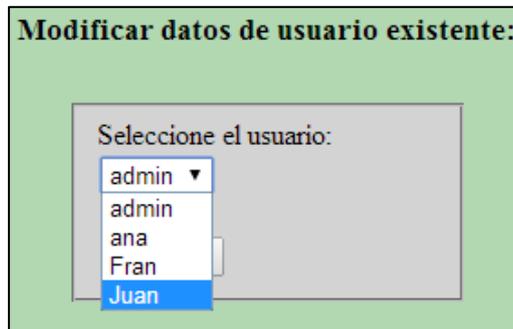
The screenshot shows the same registration form titled "Introduzca las credenciales del nuevo usuario:". The "Usuario" field contains "Juan", "E-mail" contains "jcapella@disca.upv.es", "Contraseña" contains ".....", and "Repetir contraseña" contains ".....". The "Seleccione el tipo de usuario:" dropdown is set to "Administrador". At the bottom, there are "Registrar" and "Cancelar" buttons. A red oval highlights a green success message: "El usuario Juan ha sido registrado correctamente".

Ilustración 45: Formulario registro nuevo usuario con datos correctos

Una vez se tiene el nuevo usuario almacenado en la tabla **usuarios** se puede llevar a cabo la modificación de sus datos, ya que tal vez tras el paso del tiempo el email introducido ya no es el mismo o no se recuerda bien cuál es la contraseña, etc. Para ello en la ventana del *formulario_registro.php* se tiene acceso al segundo formulario, al del cambio de datos, en el cual se da opción de seleccionar mediante un desplegable el usuario a modificar, tal y como se señala en la *Ilustración 46*.

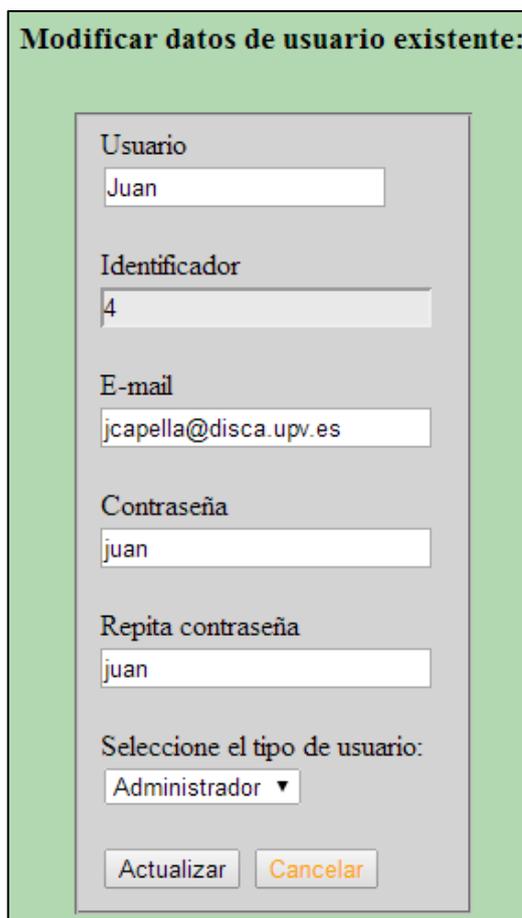
Tras haber enviado el nombre del usuario a modificar, se imprime un nuevo formulario en el que aparecen impresos los datos de dicha persona y se da la opción de modificarlos, este es similar al mostrado en la *Ilustración 47*. La comprobación de los nuevos datos introducidos se lleva acabo de manera similar al

formulario mostrado en la *Ilustración 44*, ya que los errores que pueden aparecer en el caso de que sean incorrectos, son los mismos. En el caso contrario en el que los datos introducidos sean correctos aparecerá el mensaje remarcado en la *Ilustración 48*.



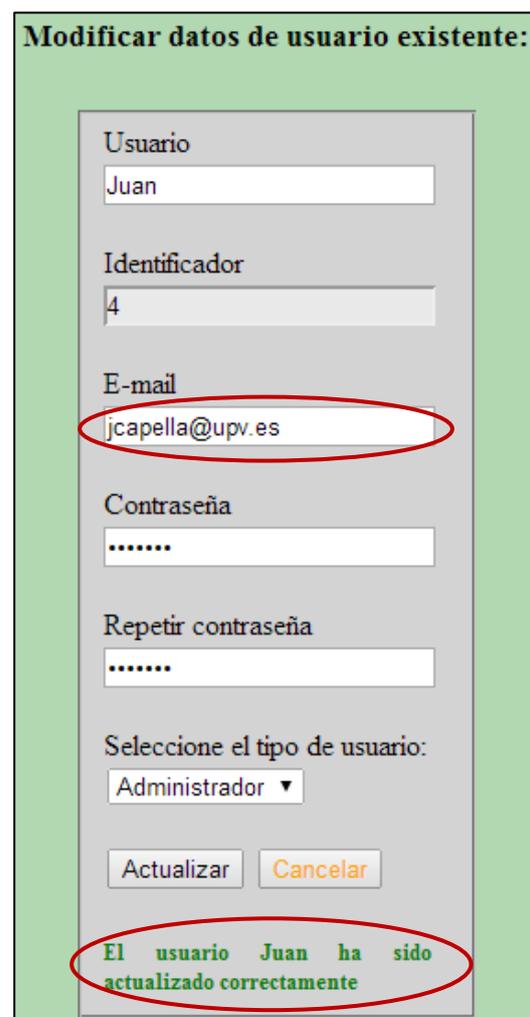
The screenshot shows a form titled "Modificar datos de usuario existente:". Below the title is a label "Seleccione el usuario:" followed by a dropdown menu. The dropdown menu is open, showing a list of user names: "admin", "admin", "ana", "Fran", and "Juan". The "Juan" option is highlighted in blue, indicating it is the selected user.

Ilustración 46: Formulario selección de usuario a modificar



The screenshot shows the "Modificar datos de usuario existente:" form with the following fields filled out: "Usuario" (Juan), "Identificador" (4), "E-mail" (jcapella@disca.upv.es), "Contraseña" (juan), "Repita contraseña" (juan), and "Seleccione el tipo de usuario:" (Administrador). At the bottom, there are two buttons: "Actualizar" and "Cancelar".

Ilustración 47: Formulario con los datos de la BD a modificar



The screenshot shows the "Modificar datos de usuario existente:" form with the same data as in Illustration 47. The "E-mail" field (jcapella@upv.es) and the "Actualizar" button are circled in red. At the bottom of the form, a green message is displayed: "El usuario Juan ha sido actualizado correctamente", which is also circled in red.

Ilustración 48: Formulario de modificación de datos enviado correctamente

Por último existe un tercer formulario, mostrado en la *Ilustración 49*, para el borrado de usuarios existentes en la base de datos, dicho formulario contiene un único desplegable en el que el administrador debe seleccionar el usuario que desea eliminar, incluyéndose a él mismo. Tras dicha selección se pulsa el botón “**Borrar usuario**” para que esta acción sea llevada a cabo, aunque tras haber pulsado el botón antes de que se realice el correspondiente borrado, se le pregunta al administrador si realmente quiere eliminar al usuario que ha seleccionado, puesto que puede haber pulsado el botón por error y esta advertencia evitaría que se borrara de forma incorrecta.

Si la advertencia es aceptada entonces se realiza el pertinente borrado de usuario y ya no aparece en ninguno de los desplegados, ni en el de modificación de usuarios ni en el del borrado. En el caso contrario, en el que se rechaza la advertencia no sucede nada, y por lo tanto no se ejecuta el borrado. Esta advertencia será similar a la mostrada en la *Ilustración 50*, decimos similar puesto que dependiendo del usuario a borrar cambiara el nombre remarcado en la advertencia.

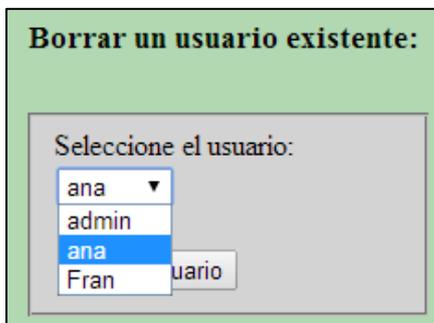


Ilustración 49: Formulario borrado de usuarios

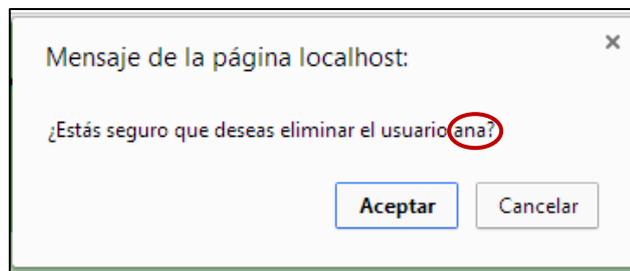


Ilustración 50: Advertencia borrado de usuarios

ANEXO C: PRUEBAS CON SMARTPHONE

Para dejar constancia de que esta aplicación web funciona correctamente en dispositivos que cada vez son más utilizados, como son el computador, cuyo correcto funcionamiento se puede ver descrito en el *Anexo B* mediante descripciones e ilustraciones que lo corroboran, un Smartphone, cuyo correcto funcionamiento se muestra en este anexo, o una Tablet, cuyo funcionamiento no se detalla puesto que no poseemos de una, para poder realizar las pertinentes pruebas, aunque si funciona correctamente con el emulador de google Chrome.

La aplicación domótica desarrollada se puede probar desde cualquier lugar, sin necesidad de permanecer en la vivienda, nosotros lo hemos probado en la red local, puesto que la aplicación no ha sido publicada. Para ello nos hemos conectado a través de nuestro Smartphone, a la red local, compartida entre los diferentes dispositivos en la misma conexión *wifi*. Simplemente era necesario disponer de cualquier navegador en nuestro dispositivo, nosotros hemos accedido a través del navegador *Google Chrome*, a la URL “http://192.168.0.204/PROYECTO/Formulario_acceso.php”. (Dirección IP de nuestro portátil, donde se encuentra almacenada la plataforma y la base de datos, y ubicación de la carpeta)

Después de realizar este simple paso, pudimos acceder a la página de inicio de nuestra aplicación domótica y pudimos comprobar que se ajustaba perfectamente a la pantalla de nuestro dispositivo móvil, validando al mismo tiempo la funcionalidad de todos los botones y campos de texto, que se encuentran en dicha ventana.



Ilustración 51: Prueba Smartphone, formulario inicio sesión vacío

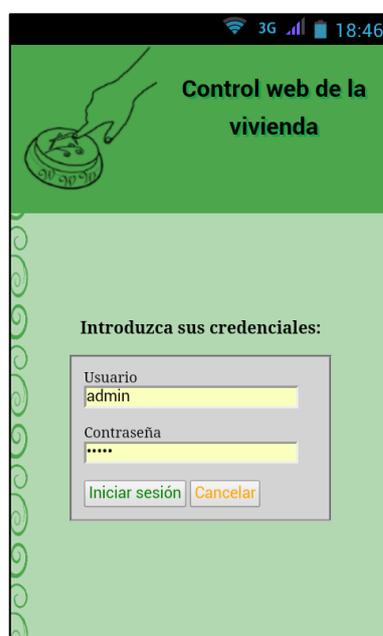


Ilustración 52: Prueba Smartphone, formulario acceso datos correctos

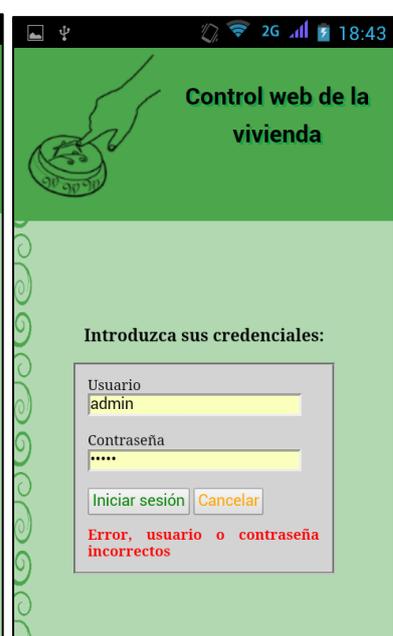


Ilustración 53: Prueba Smartphone, formulario acceso contraseña incorrecta

Tras pulsar el botón “Iniciar sesión” se accede a la aplicación web, y se redirige al usuario a la pantalla principal, en la cual se diseñan las estancias y los elementos. Su funcionalidad es idéntica a la que se realiza en la aplicación utilizando un computador, pero la visión es un poco diferente puesto que al ser una pantalla mucho más pequeña que la de un computador, el tamaño de las letras disminuye y además las tablas aparecen verticalmente, es decir unas detrás de otras y los desplegables se muestran en toda la pantalla. Las diferentes acciones que se pueden realizar en dicha ventana se ven reflejadas en las siguientes ilustraciones.

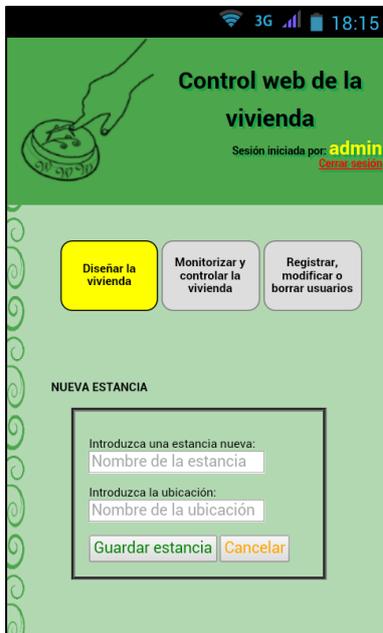


Ilustración 54: Prueba Smartphone, formulario nueva estancia vacío

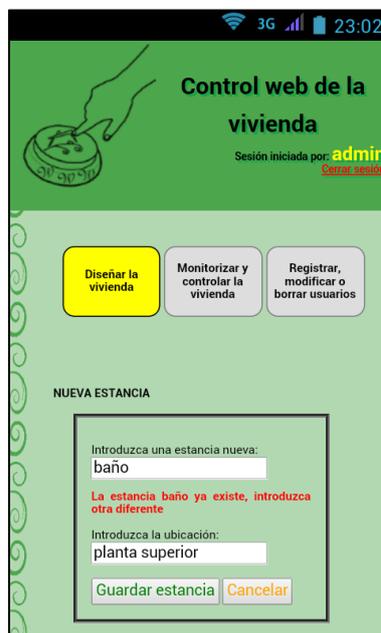


Ilustración 55: Prueba Smartphone, nueva estancia repetida

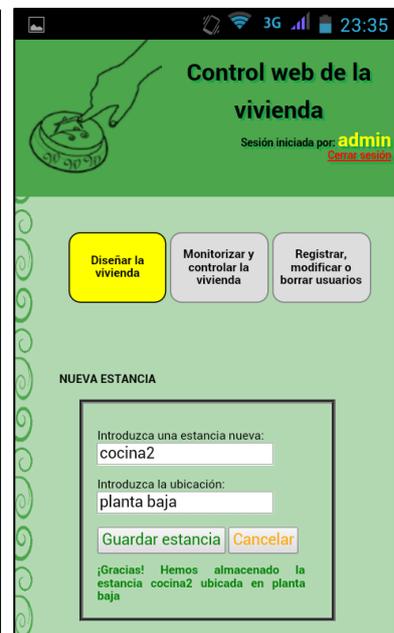


Ilustración 56: Prueba Smartphone, nueva estancia datos correctos

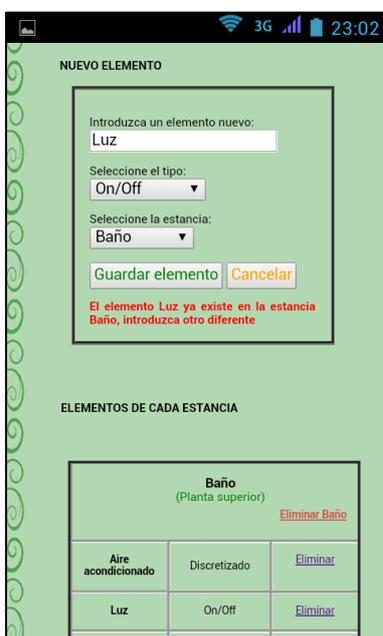


Ilustración 57: Prueba Smartphone, nuevo elemento repetido

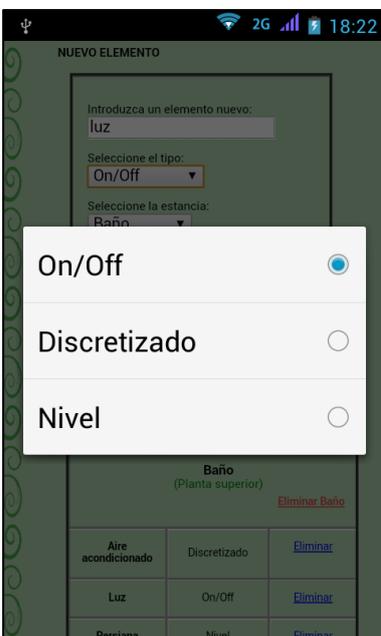


Ilustración 58: Prueba Smartphone, desplegable selección tipo elemento

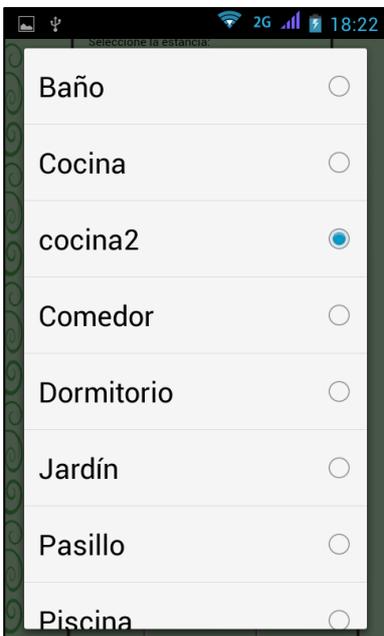


Ilustración 59: Prueba Smartphone, desplegable selección estancia

Al introducir correctamente los campos de la nueva estancia y del nuevo elemento a almacenar, se visualizan en la misma ventana diferentes tablas separadas por estancias, en las que se pueden ver los elementos y las estancias introducidas de manera vertical, donde se da la opción de eliminar tanto la estancia como el elemento, por separado. En las siguientes capturas podemos ver la estancia y el elemento, finalmente visualizado, que se ha ido introduciendo tal y como se ha mostrado en las capturas desde la *Ilustración 54* a la *59*, así como también se muestra el mensaje de confirmación de borrado del elemento, en la *Ilustración 61*.

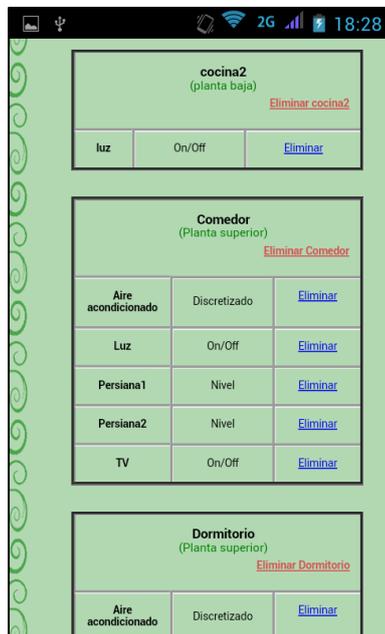


Ilustración 60: Prueba Smartphone, visualización estancias y elementos

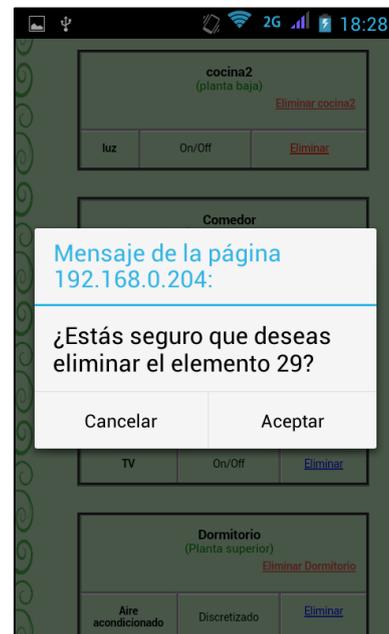


Ilustración 61: Prueba Smartphone, mensaje confirmación eliminar elemento

Si pulsamos el botón para controlar y monitorizar los elementos de la vivienda, nos aparecerán la pantalla mostrada en la *Ilustración 62*, *63* y *64* con las tablas ordenadas verticalmente y ajustadas al ancho de la pantalla.

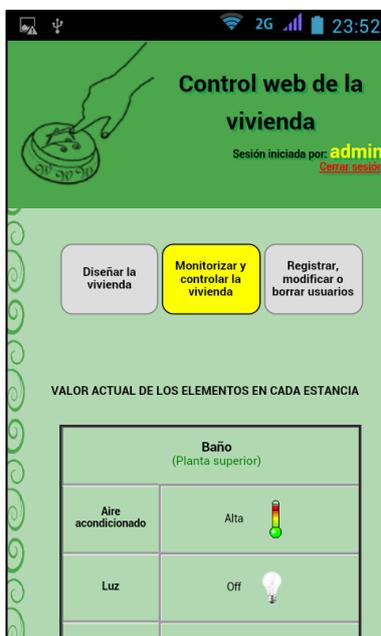


Ilustración 62: Prueba Smartphone, monitorización



Ilustración 63: Prueba Smartphone, control

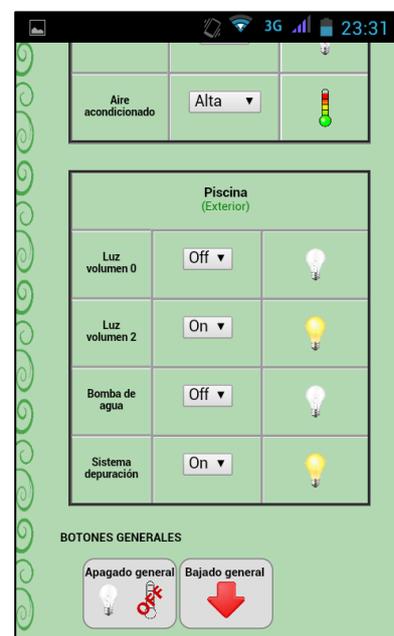


Ilustración 64: Prueba Smartphone, control general



Al pulsar el tercero de los botones se nos muestra la pantalla de registro, borrado y modificación de usuarios, cuyos formularios también se muestran de manera vertical, unos detrás de otros y ajustados al ancho de lo que es la pantalla de nuestro móvil, la funcionalidad de los desplegados y los campos de texto funcionan correctamente, tal y como se ha podido observar en las ilustraciones anteriores, por lo que no es necesario volver a mostrar su correcto funcionamiento.



Ilustración 65: Prueba Smartphone, registro nuevo usuario



Ilustración 66: Prueba Smartphone, borrar o modificar usuario

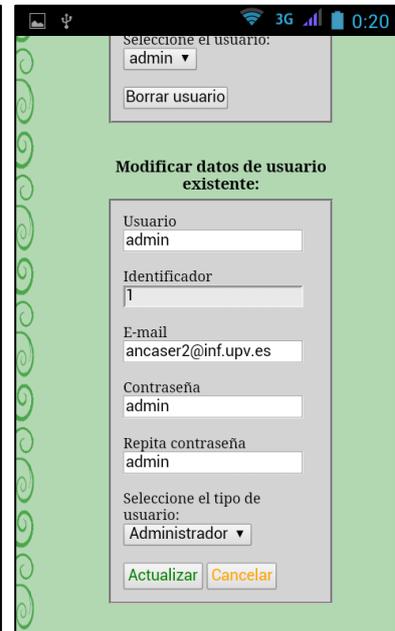


Ilustración 67: Prueba Smartphone, modificar usuario

Por último solo queda destacar que esta aplicación también funciona correctamente en el caso de que la pantalla del Smartphone sea volteada, es decir cambiemos de posición el móvil, de vertical a como se han venido mostrando las capturas de la pantalla a horizontal, de manera que todas las tablas, botones y campos que componen la aplicación domótica, se ven ajustados al ancho de la pantalla, a continuación mostramos una última ilustración que demuestra dichas palabras.

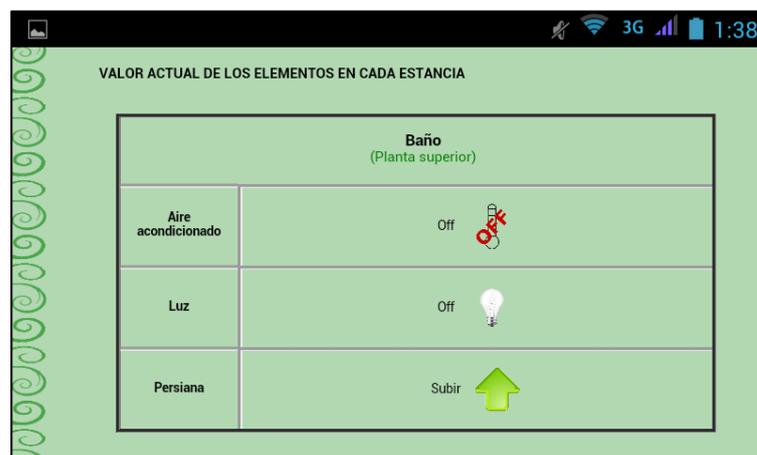


Ilustración 68: Prueba Smartphone horizontal