



Universidad
de La Laguna

Casa Virtual Domotizada con lógica borrosa

Virtual Domotic House with fuzzy logic

Carlos Alberto González Carballo

Ingeniería Informática

Escuela Superior de Ingeniería y Tecnología

Sección de Ingeniería Informática

Trabajo de Fin de Grado

La Laguna, 7 de septiembre de 2014

Dña. **Vanesa Muñoz Cruz**, con N.I.F. 78.698.687-R profesora Ayudante Doctor adscrita al Departamento de Ingeniería Informática de la Universidad de La Laguna.

Dña. **Silvia Alayón Miranda**, con N.I.F. 43.812.596-B profesora Contratado Doctor adscrita al Departamento de Ingeniería Informática de la Universidad de La Laguna.

C E R T I F I C A

Que la presente memoria titulada:

“Casa Virtual Domotizada con lógica borrosa”

ha sido realizada bajo su dirección por D. Carlos Alberto González Carballo, con N.I.F. 78.625.394-D.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firma la presente en La Laguna a 7 de septiembre de 2014.

Agradecimientos

A Dña. Vanesa Muñoz Cruz por mostrarme su apoyo, brindarme su tiempo, aportarme sus conocimientos, y ayudarme en la realización de este proyecto.

A Dña. Silvia Alayón Miranda por permitirme realizar este proyecto.

A mi familia, por permitirme realizar esta carrera en la época en la que estamos, además del apoyo y confianza que siempre tienen en mí.

A Virginia, por siempre estar ahí apoyándome y animándome cuando más lo necesitaba, y por estar a mi lado siempre, en lo bueno y en lo malo.

A mis amigos, sobre todo a los grandes amigos que he hecho a lo largo de estos años en la ULL, sin ellos esto no sería posible, gracias a su ayuda, apoyo, risas, fiesta, etc. Esto no acaba aquí, solo comienza.

Y para finalizar he de agradecer a la Universidad de La Laguna por esta oportunidad y a todos los docentes que han hecho posible que llegará hasta aquí.

Resumen

En los últimos años, la tecnología avanza a pasos agigantados, y aunque para algunos es fácil seguir ese avance, para otros no lo es tanto. Mucha gente desconoce ciertas tecnologías nuevas, que facilitan la vida cotidiana, y no por no querer conocerla, sino por falta de ayuda, tiempo, y multitud de razones.

Una de esas tecnologías desconocidas para mucha gente es la domótica, la cual nos puede aportar más confort, seguridad, eficiencia en la vida cotidiana, en nuestra propia casa. También es reseñable decir, que aunque hay gente que sí la conoce piensa que es muy caro incorporar sistemas domotizados a la vivienda y no se da cuenta que realmente puedes ahorrar en los gastos que genera una casa, dígase agua, luz, gas, etc.

El objetivo de este proyecto es crear una aplicación que contenga una casa virtual dotada de un sistema domótico en cual se pueda aprender o simplemente entender que es y cómo funciona la domótica. Además se ha utilizado lógica borrosa para controlar algunos de los sistemas domóticos.

Palabras clave

Tecnología, domótica, casa, lógica difusa.

Abstract

In recent years, technology has advanced rapidly. Although for some people it is easy to keep on track with the progress; for other ones, this evolving process leaves them behind. This is why many people are unaware of new technologies that could facilitate their daily life, although they would actually like to embrace them if they had the chance. However, the lack of external support, time enough, and some other reasons, keeps them away from it.

One of these unknown technologies to the general public is the domotic, which can brings us comfort, safety and efficiency in everyday life at home. It is also noteworthy that although there are people who know about it, they have the wrong believe that incorporating housing domotic systems is very expensive. Despite, you can actually save on home expenses, i.e. water, electricity, gas, etc.

The main goal of this project is to create an application that contains a virtual house equipped with a domotic system in which to learn or at least understand what is domotic and how it works. Besides, fuzzy logic has been used to control some domotic systems.

Keywords

Technology, domotic, home, fuzzy logic.

Índice General

Agradecimientos	5
Resumen	7
Abstract	9
Índice General	11
Índice de figuras y tablas	14
Capítulo 1. Introducción	16
1.1 Lógica borrosa/difusa.	18
Capítulo 2. Objetivo	20
Capítulo 3. Herramientas.	21
3.1 Herramientas de creación y tratamiento de imágenes	21
3.1.1 Sweet Home 3D.	21
3.1.2 Adobe Photoshop CS6.	22
3.2 Herramientas de programación.	23
3.2.1 Java.	23
3.2.2 Eclipse IDE for Java EE Developers.	23
3.2.3 jFuzzyLogic.	24
Capítulo 4. Diseño e implementación.	26
4.1 Diseño de la casa.	26
4.2 Elementos de la casa.	27
4.3 Funciones Domóticas.	27
4.3.1 Sensor de Presencia.	27
4.3.2 Sensor y extractor de humo.	27
4.3.3 Sensor y extractor de agua o inundación.	28
4.3.4 Sensor de temperatura exterior y actuador.	28
4.3.5 Sensor de humedad exterior y actuador.	29
4.3.6 Medidor de viento exterior y actuador.	30
4.3.7 Funciones adicionales.	30
4.4 Programación.	30

Capítulo 5. Resultados.	33
5.1 La casa.	33
5.2 Panel lateral.	34
5.2.1 Panel Exterior.	34
5.2.2 Panel interior.	36
5.3 Panel inferior.	36
5.3.1 Baño.	36
5.3.2 Cocina.	37
5.3.3 Dormitorio, escaleras y salón	37
5.3.4 Jardín.	38
Conclusiones y Trabajos Futuros	40
Summary and Conclusions	42
Bibliografía	44
Anexo I: Código de la lógica borrosa de la sistema de cristales.	46
Anexo II: Código de la lógica borrosa del sistema de riego.	47
Anexo III: Código clase padre de las secciones de la casa.	49
Anexo IV: Código de la clase encargada del panel lateral.	52

Índice de figuras y tablas

Figura 1. Aspectos de la domótica.	17
Figura 2. Algunas opciones de la domótica.	17
Figura 3. Panel táctil y Smartphone conectado a un sistema domótico.	18
Figura 4. Casa con sistema de domótica.	20
Figura 5. Proyecto en Sweet Home 3D.	22
Figura 6. Adobe Photoshop CS6.	22
Figura 7. Logo de Java.	23
Figura 8. Eclipse IDE for Java EE Developers.	24
Figura 9. Diseño de la casa.	26
Figura 10. Lámpara con sensor de presencia.	27
Figura 11. Cocina con humo.	28
Figura 12. Casa con los cristales oscurecidos por la temperatura exterior.	29
Figura 13. Sensores y riego del jardín.	29
Figura 14. Persianas a medias por la actuación de sensor del viento.	30
Figura 15. Aplicación.	33
Figura 16. Panel lateral.	34
Figura 17. Función del panel lateral exterior.	35
Figura 18. Función del panel lateral exterior.	35
Figura 19. Función del panel lateral exterior.	35
Figura 20. Función del panel lateral exterior.	35
Figura 21. Función del panel lateral interior.	36
Figura 22. Panel inferior.	36
Figura 23. Panel inferior: baño.	36
Figura 24. Panel inferior: Cocina.	37
Figura 25. Panel Inferior común.	37

Capítulo 1. Introducción

Desde hace años, la tecnología va avanzando a gran ritmo. Todo lo que nos rodea se va actualizando, modernizándose, informatizándose con el objetivo de ir mejorando los procesos o acciones que realizamos en el día a día, ya sea en el trabajo, en la calle o en casa para mejorar la calidad de vida de las personas o los procesos en el trabajo para mayor eficiencia.

Además de los grandes avances tecnológicos, también se viene concienciando a la población de la importancia del ahorro energético, de la necesidad de luchar contra el cambio climático y de minimizar la dependencia de los combustibles fósiles.

Por todo esto, surgen cada vez más sistemas automatizados, que realizan las tareas más eficientemente ahorrando energía y prestando un servicio más confortable a las personas.

La electricidad nos ha permitido elevar el nivel de confort en nuestras casas y ha dado paso a la entrada de los electrodomésticos, máquinas capaces de realizar tareas cotidianas de forma casi autónoma, elevando nuestro nivel de confort a cotas en otro tiempo inimaginables. Estas máquinas no existirían sin el desarrollo de una nueva evolución: la electrónica, permitiendo realizar programaciones que regulan cada proceso.

La siguiente evolución que ha llegado es la domótica. El término proviene del latín *domus* añadiéndole el final de la palabra "informática" y, según explica la propia Real Academia Española de la Lengua, es el "**conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda**". El principal objetivo de estas tecnologías es la mejora de la calidad de vida incrementando la comodidad de los inquilinos, sin embargo, se ha demostrado que usando dicha tecnología se puede ahorrar y a su vez ser más ecológico ya que se puede implementar en distintos ámbitos del hogar. Desde los típicos controladores de la calefacción hasta sistemas de gestión del agua, iluminación, gas o incluso sistemas automáticos de persianas y toldos basados en sensores de incidencia solar y temperatura. Todos ellos para conseguir ahorrar agua, gas y, sobre todo, electricidad. De hecho, según datos, se puede llegar a ahorrar más del 50 por ciento del consumo eléctrico de un hogar en función de múltiples variables. De ahí que, en muchas ocasiones la domótica no se observe como un gasto a la hora de realizar una reforma en la vivienda, sino como una verdadera inversión de futuro.

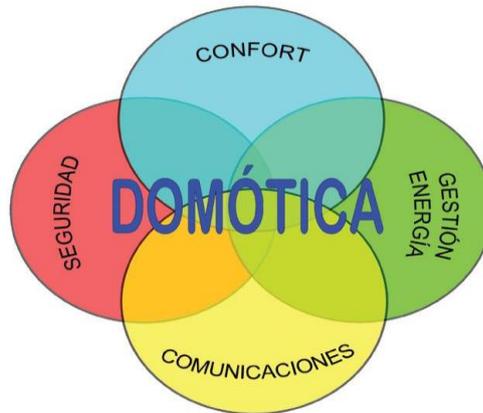


Figura 1. Aspectos de la domótica.

Lo primero que se inventó fueron los **automatismos**, elementos como un enchufe con temporizador, un interruptor que hace que se active el mecanismo de subir/bajar la persiana...

La domótica va más allá, haciendo que la casa disponga de una “inteligencia”, pero que realmente está basada en acción - reacción, programada por nosotros mismos. Un ejemplo sería programarla para que a las 11 de la mañana suba las persianas a media altura, a las 21h encienda luces (hasta las 23h por ejemplo) para simular que hay alguien viviendo, por si nos hemos ido de vacaciones, detectar si hay un escape de gas mediante un sensor y que el modulo encargado envíe una señal de cierre a la válvula general del gas.

Pero además también la podemos controlar con un Smartphone u ordenador desde internet, de manera que si estamos en invierno y sabemos que en media hora llegaremos a casa, podemos activar la calefacción para que cuando llegemos, la casa ya este a una temperatura adecuada.



Figura 2. Algunas opciones de la domótica.

Pero incluso pueden actuar sin ser programados por el usuario, es decir, si la instalación cuenta con un sensor de viento, y éste mide una velocidad de viento que pueda producir daños a los cristales de la casa, puede actuar en consecuencia y automáticamente bajar las persianas para proteger los cristales. En este trabajo de fin de grado se ha utilizado la lógica borrosa o difusa para realizar estas actuaciones

1.1 Lógica borrosa/difusa.

La lógica borrosa es una rama de la inteligencia artificial que se funda en el concepto "Todo es cuestión de grado", lo cual permite manejar información de difícil especificación si quisiéramos hacer cambiar con esta información el funcionamiento o el estado de un sistema específico. Es entonces posible con la lógica borrosa gobernar un sistema por medio de reglas de 'sentido común' las cuales se refieren a cantidades indefinidas.

Las reglas involucradas en un sistema borroso, pueden ser aprendidas con sistemas adaptativos que aprenden al 'observar' como operan las personas los dispositivos reales, o estas reglas pueden también ser formuladas por un experto humano. En general la lógica borrosa se aplica tanto a sistemas de control como para modelar cualquier sistema continuo de ingeniería, física, biología o economía.

La lógica borrosa es entonces definida como un sistema matemático que modela funciones no lineales, que convierte unas entradas en salidas acordes con los planteamientos lógicos que usan el razonamiento aproximado.

Se fundamenta en los denominados conjuntos borrosos y un sistema de inferencia borroso basado en reglas de la forma " SI..... ENTONCES..... ", donde los valores lingüísticos de la premisa y el consecuente están definidos por conjuntos borrosos.



Figura 3. Panel táctil y Smartphone conectado a un sistema domótico.

Capítulo 2. Objetivo

El objetivo de este proyecto es realizar una aplicación en java, la cual incluya una casa virtual que disponga de un sistema domótico en el que podamos experimentar el uso de dicha tecnología y probar diferentes configuraciones.

Dicha aplicación tiene que implementar diferentes elementos que se usen en los sistemas domóticos, desde los elementos más sencillos, dígase un sensor de presencia, hasta elementos más complejos que utilicen lógica borrosa.



Figura 4. Casa con sistema de domótica.

Capítulo 3. Herramientas.

A continuación, se detallarán las herramientas utilizadas durante el desarrollo del trabajo de final de grado.

Para la elección de estas herramientas, se hizo una comparación entre las más usadas en la actualidad, pero se buscaron aquellas herramientas en las cuales se tuviera alguna experiencia, para facilitar el proceso.

Se listarán las herramientas utilizadas y su finalidad. Se dividirán en dos grandes grupos de herramientas: herramientas de creación y tratamiento de imágenes, y herramientas de programación.

3.1 Herramientas de creación y tratamiento de imágenes

3.1.1 Sweet Home 3D.

Sweet Home 3D es una aplicación libre de diseño de interiores que te ayuda a colocar tu mobiliario en un plano 2D de la casa, con una vista previa 3D [1].

Este programa está dirigido a personas que quieren diseñar interiores con rapidez, si trabajan en ello o si sólo quieren rediseñar su casa. Numerosas guías visuales le ayudan a sacar el plano de su casa y el diseño de muebles. Usted puede dibujar las paredes de sus habitaciones a partir de la imagen de un plano existente y, a continuación, arrastrar y colocar los muebles en el plano a partir de un catálogo organizado por categorías. Cada cambio en el plano 2D es a la vez actualizado en la vista 3D, para mostrar una representación realista en su diseño.

En este proyecto, dicho programa ha permitido crear la imagen de la casa para usarla como base de la aplicación, ha facilitado la creación de esta, ya que dispone de multitud de objetos, cotidianos o no, que se necesitan o suelen estar en una casa.

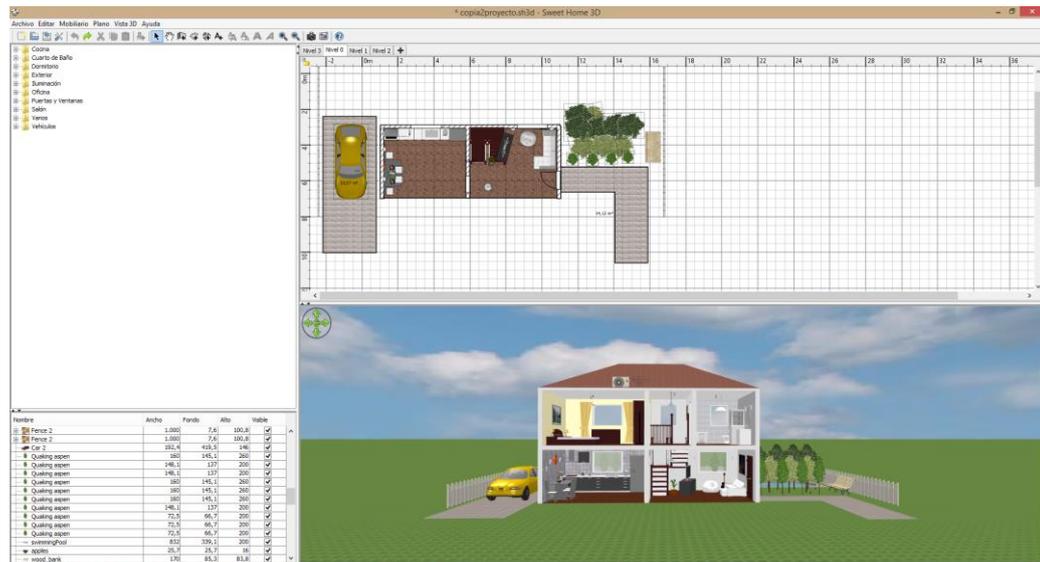


Figura 5. Proyecto en Sweet Home 3D.

3.1.2 Adobe Photoshop CS6.

Adobe Photoshop CS6 es un programa utilizado para editar, componer, retocar y transformar imágenes. Su gran facilidad para crear y manejar distintas capas superpuestas, nos permite combinar distintos objetos y efectos [6] [7].

En este proyecto, se ha hecho uso de esta herramienta para conseguir los efectos de diferentes cielos, persianas, cristales, luces, etc. Con la facilidad de uso que tiene su sistema de capas, nos permite variar la imagen según nos convenga.

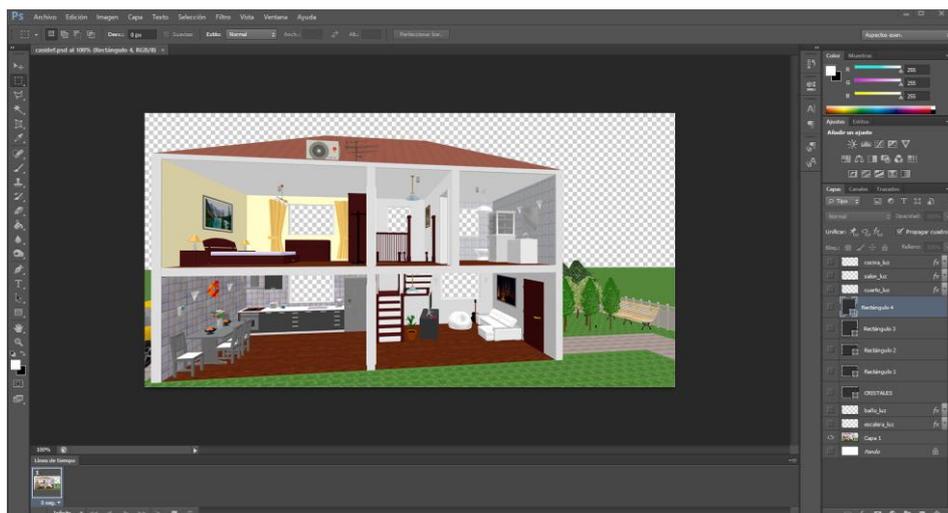


Figura 6. Adobe Photoshop CS6.

3.2 Herramientas de programación.

3.2.1 Java.

Java es el lenguaje de programación escogido para implementar la parte de programación de este proyecto. Es un lenguaje de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra [4].



Figura 7. Logo de Java.

3.2.2 Eclipse IDE for Java EE Developers.

Eclipse IDE for Java EE Developers es una herramienta para desarrolladores Java que permite crear aplicaciones. Posee una estructura modular, extensible mediante plugins, que le permite trabajar con cualquier tipo de recurso: gráficas, vídeo, modelos 3D, contenido web, etcétera. Otros lenguajes que también pueden utilizarse en Eclipse son: C/C++, PHP, Ruby, TCL o Javascript [5].

Como IDE para Java cuenta con algunas funciones interesantes, entre ellas: desarrollo de aplicaciones en grupo, unidad integrada de depuración y pruebas, compilación y construcción incremental, etcétera.

En este proyecto, se decidió usar esta herramienta ya que facilita el trabajo a la hora de programar, pero sobre todo a la hora de depurar. Otra ventaja es que ya se contaba con experiencia de uso de este IDE.

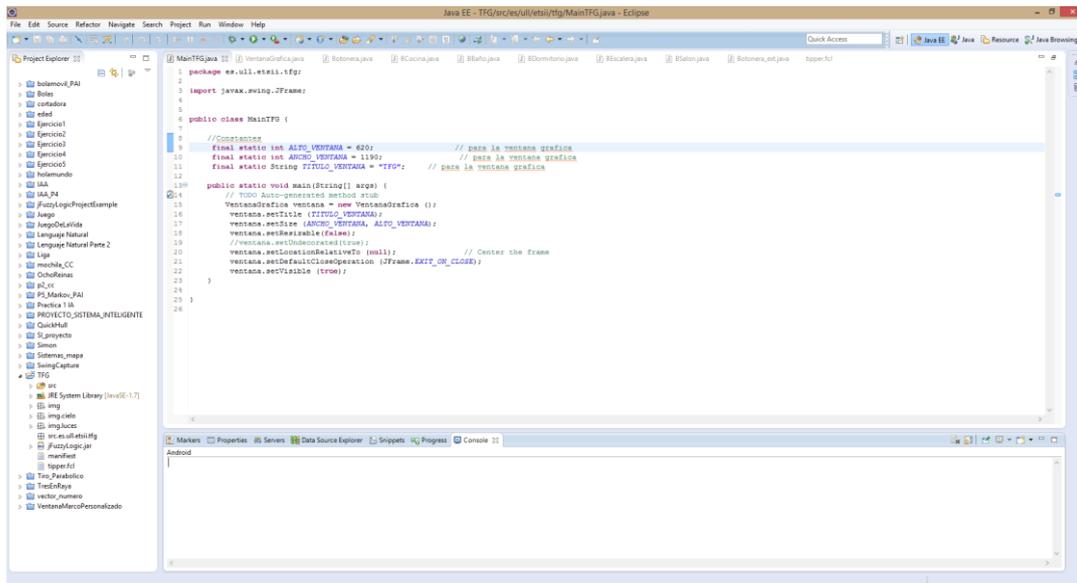


Figura 8. Eclipse IDE for Java EE Developers.

3.2.3 jFuzzyLogic.

jFuzzyLogic es una librería de lógica difusa de código abierto para simplificar los desarrollos de los sistemas difusos. Implementa el lenguaje de control Fuzzy (FCL), así como una biblioteca completa que simplificará en gran medida el desarrollo de los conjuntos, reglas, métodos de desborrosificación, etc [8].

Se ha hecho uso de esta librería para la parte de lógica difusa o borrosa de los elementos de la domótica que lo necesitan.

Capítulo 4. Diseño e implementación.

4.1 Diseño de la casa.

Para el diseño de la casa se parte de la base de que tiene que ser sencillo pero a su vez pueda contener el mayor número de opciones domóticas que existen hoy en día. Se decidió por tanto una distribución de dos plantas, tal como se ve en la figura siguiente.



Figura 9. Diseño de la casa.

En total la casa dispone de seis secciones, que son: cocina, salón, dormitorio, baño, escaleras y jardín.

En la primera planta está la cocina, situada a la izquierda de la imagen, y el salón a continuación, el cual dispone de unas escaleras que llevan a la segunda planta. La entrada de la casa está situada en la derecha.

En la segunda planta está el dormitorio a la izquierda, y el baño a la derecha, y en medio se encuentra una escalera que conecta el salón con la primera planta.

Fuera de la casa está el jardín con un camino de baldosas de piedra que lleva hasta la puerta.

4.2 Elementos de la casa.

La casa que se ha diseñado para la simulación dispone de todos los elementos que se encuentran normalmente en una casa real como pueden ser persianas, luces, mobiliario, etc.

4.3 Funciones Domóticas.

Se han añadido multitud de funciones domóticas a la aplicación, muchas de ellas están presentes en casi todas las secciones de la casa y otras son específicas de alguna de ellas. A continuación, en los siguientes apartados se describirán los sistemas domóticos que se han implementado.

4.3.1 Sensor de Presencia.

En todas las secciones de la casa se han incluido sensores de presencia. Su función es detectar si hay alguien en esa sección o habitación y en el caso de la simulación se utilizará para seleccionar la sección. Lo que hace es encender o apagar la luz, dependiendo del estado del sensor, siempre que sea necesario, ya que si es de día y la luminosidad es buena, no hace falta encender la luz.



Figura 10. Lámpara con sensor de presencia.

4.3.2 Sensor y extractor de humo.

Esta función está disponible para la cocina, y lo que hace es que detecta mediante un sensor si existe humo y en qué cantidad, pudiendo activar un extractor si fuera necesario para su eliminación.

En este sistema se usa la lógica borrosa, ya que no se ha tenido en cuenta solo la cantidad de humo que pueda haber, sino de si la persiana está abierta o no, por tanto para su implementación se ha usado un sistema de dos entradas y una salida.

Las entradas del sistema borroso serían, la cantidad de humo existente y la posición de la persiana. La salida sería si activar o no el extractor de humo, y a qué velocidad. Desde que exista humo la persiana se abrirá para proceder a su mejor extracción.

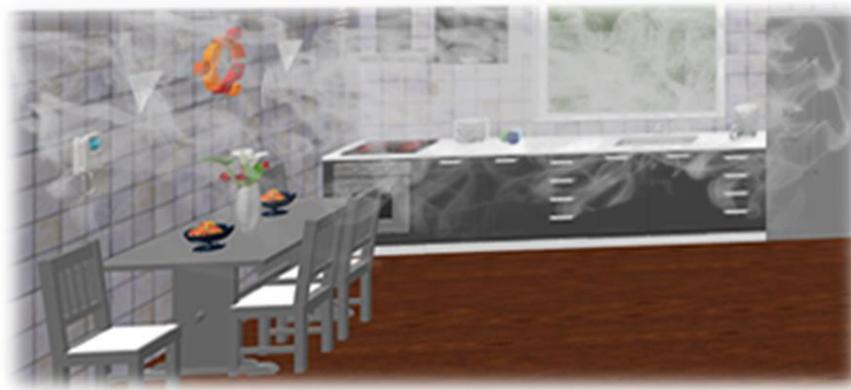


Figura 11. Cocina con humo.

4.3.3 Sensor y extractor de agua o inundación .

Esta función está disponible para el baño, y lo que hace es que detecta si existe un nivel de agua anormal en el suelo del baño, si así fuera cortaría el suministro de agua al baño, y abriría un desagüe para proceder a su evacuación.

4.3.4 Sensor de temperatura exterior y actuador.

Está función no está ligada a ninguna sección concreta de la casa, sino a la casa en sí. Este sensor mide la temperatura exterior a la casa, y según la medición actúa sobre los cristales, usando una tecnología que oscurece los cristales para mantener una temperatura en el interior, y las persianas.

También mediante lógica borrosa, se le da a un sistema borroso la medida de la temperatura y este da la salida de la opacidad de los cristales y la posición de las persianas. Si la temperatura sube el cristal estará más oscuro para compensar el calor, y si la temperatura baja el cristal se aclara. Respecto a las persianas, si la temperatura baja mucho,

se cerrarán para compensar el cambio, y evitar así una bajada de temperatura dentro de la casa.

En el Anexo I se pueden consultar los detalles del sistema borroso implementado.



Figura 12. Casa con los cristales oscurecidos por la temperatura exterior.

4.3.5 Sensor de humedad exterior y actuador.

Esta función está disponible para el jardín, dependiendo de la humedad exterior que mida el sensor, por medio de un sistema de lógica borrosa se calculará junto con la temperatura exterior si se tiene que regar o no el jardín y en qué cantidad. Por lo tanto tenemos un sistema borroso de dos entradas (humedad y temperatura) y una salida (cantidad de riego).

En el Anexo II se pueden consultar los detalles del sistema borroso implementado.



Figura 13. Sensores y riego del jardín.

4.3.6 Medidor de viento exterior y actuador.

Igual que en el caso de la temperatura exterior, esta función no está ligada a ninguna sección concreta sino que influye en toda la casa. Este sensor mide la velocidad del viento que incide en la casa, y según la medición actúa sobre las persianas de la casa, para evitar así, cualquier daño a los cristales.

Aquí también se usa lógica borrosa, y según la velocidad del viento se determinará la posición de las persianas.



Figura 14. Persianas a medias por la actuación de sensor del viento.

4.3.7 Funciones adicionales.

Se dispone también, además de las anteriores funciones, de un actuador que permite subir o bajar manualmente todas las persianas de la casa.

4.4 Programación.

Para la programación de la aplicación se realizó una estructura de clases, que dependen de una principal en la cual están todos los eventos necesarios para ejecutar las funciones domóticas implementadas.

Para las secciones de la casa se ha creado una clase padre que contuviera las funciones que existen en todas ellas y a partir de dicha clase se usa herencia para crear cada una de las secciones. En el Anexo III se pueden consultar los detalles la clase implementada.

También se ha dotado a la aplicación de diferentes tiempos climáticos, es decir, se puede ir cambiando los escenarios y con ello se cambian variables tales como temperatura exterior, humedad, viento, etc.

Capítulo 5. Resultados.

A continuación, se mostrarán los resultados obtenidos durante el desarrollo del trabajo de final de grado.

La siguiente figura muestra el resultado visual de la aplicación.



Figura 15. Aplicación.

En la aplicación se pueden distinguir tres partes, la primera es la vista de la casa en sí, la segunda es el panel lateral que hace uso de las funciones generales de la casa, y la tercera es el panel inferior que nos permite acceder a las funciones más específicas de cada sección de la casa.

A continuación se explicará cada parte por separado, para así poder entender mejor su funcionamiento.

5.1 La casa.

En esta parte es donde se muestra lo más interesante de la aplicación, que es la simulación de los efectos de las funciones domóticas. Es la parte gráfica del sistema domótico.

Además de mostrar la simulación, se puede acceder a las funciones específicas de cada sección de la casa, haciendo clic en ella.

5.2 Panel lateral.

Está situado a la derecha de la aplicación, y hace uso de las funciones generales de la casa, tales como temperatura y humedad exterior, etc. Dicho panel se puede dividir en dos partes: panel exterior y panel interior.

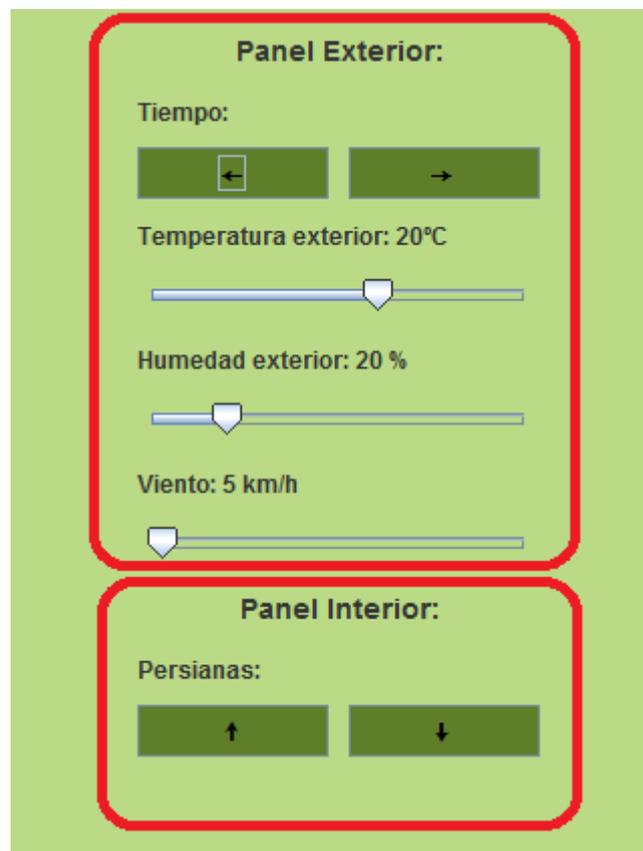


Figura 16. Panel lateral.

5.2.1 Panel Exterior.

En este panel se dispone de cuatro funciones, las cuales permiten ver y variar el valor de las variables que le afectan a la casa exteriormente:

- **Tiempo:** función que dispone de dos botones de avance lateral para ir cambiando la meteorología que hace en la simulación de la aplicación. Cada vez que se

cambia el tiempo, se actualizan los valores de temperatura, humedad y viento exterior, ya que cada uno tiene un valor según el tiempo que se ponga. Se dispone de varios tiempos diferentes: soleado, nublado de día, con viento, noche despejada con estrellas, noche cubierta, etc.



Figura 17. Función del panel lateral exterior.

- **Temperatura exterior:** función que dispone de un slider el cual permite cambiar el valor en grados Celsius de la temperatura. Según la temperatura que marque, afectará a la casa, variando así el valor de la posición de las persianas y/o la opacidad de los cristales, ya que uno de los sistemas de lógica borrosa tiene como entrada la temperatura exterior.



Figura 18. Función del panel lateral exterior.

- **Humedad exterior:** función que dispone de un slider el cual permite cambiar el valor en porcentaje de la humedad exterior. La variación de la humedad afectará al jardín, ya que el sistema de lógica borrosa que tiene el riego del jardín depende de la humedad exterior, por tanto, dicho sistema podría hacer que salte el riego automático.

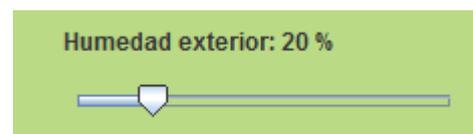


Figura 19. Función del panel lateral exterior.

- **Viento:** función que dispone de un slider el cual permite cambiar el valor en km/h del viento. Según lo que marque actuará sobre las persianas de la casa, ya que dicho valor se le pasa al sistema borroso y este devuelve la posición en la que deben estar las persianas.

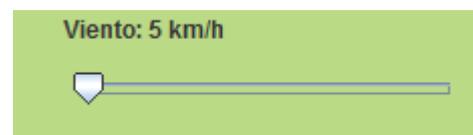


Figura 20. Función del panel lateral exterior.

5.2.2 Panel interior.

Este panel solo dispone de una función, que es bajar o subir todas las persianas que tiene la casa por medio de dos botones, uno de subida y otro de bajada.



Figura 21. Función del panel lateral interior.

5.3 Panel inferior.

En este panel se dispone de los botones que permiten acceder a las funciones de cada sección de la casa, a las cuales también se puede acceder clicando en ellas en la vista de la casa.

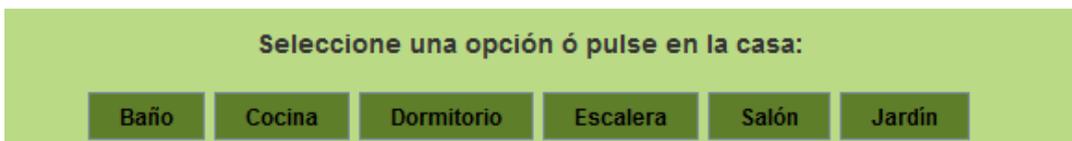


Figura 22. Panel inferior.

Una vez que se pulse en alguna sección, clicando en su botón, este panel será sustituido por el panel específico de la sección seleccionada.

5.3.1 Baño.

Cuando se selecciona la sección correspondiente al baño, aparecerá el panel de la siguiente figura.

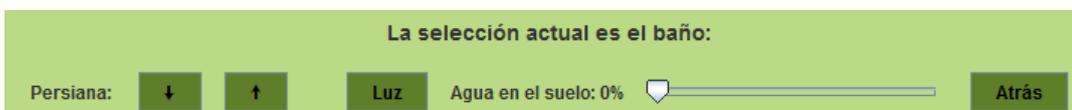


Figura 23. Panel inferior: baño.

En esta sección se dispone de subida y bajada de la persiana correspondiente al baño, botón para encender o apagar la luz, independientemente del sensor de presencia y de un botón “Atrás”, que lleva al panel inferior principal.

Además de lo anterior, también se dispone de un slider, que controla la cantidad de agua que queremos introducir en el suelo del baño, esto sirve para que actué el detector y envíe una señal al sistema borroso y que este pueda actuar en consecuencia para eliminarla.

5.3.2 Cocina.

Al seleccionar la sección correspondiente a la cocina, aparecerá en la aplicación el panel de la siguiente figura.

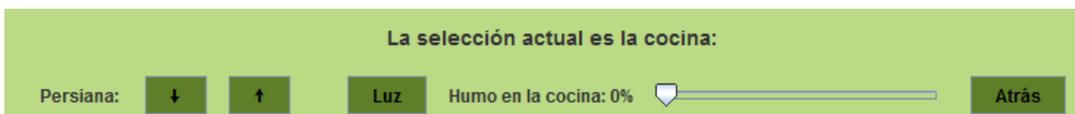


Figura 24. Panel inferior: Cocina.

En esta sección se dispone de subida y bajada de la persiana correspondiente al baño, botón para encender o apagar la luz, independientemente del sensor de presencia y de un botón “Atrás”, que lleva al panel inferior principal.

Además de lo anterior, también se dispone de un slider, que controla la cantidad de humo que queremos introducir en la cocina. Cuando se detecta humo en la cocina, el detector de humo envía una señal al sistema borroso y este actúa en consecuencia para eliminarlo.

5.3.3 Dormitorio, escaleras y salón

Cuando se seleccione la sección correspondiente a alguna de estas tres, aparecerá el panel de la siguiente figura.



Figura 25. Panel Inferior común.

Estas secciones solo tienen las funciones básicas y son comunes a la mayoría de ellas, disponemos de subida y bajada de la persiana correspondiente al baño, botón para encender o apagar la luz, independientemente del sensor de presencia y de un botón “Atrás”, que nos dirige al panel inferior principal.

5.3.4 Jardín.

Cuando se selecciona la sección correspondiente al jardín, aparecerá el panel de la siguiente figura.

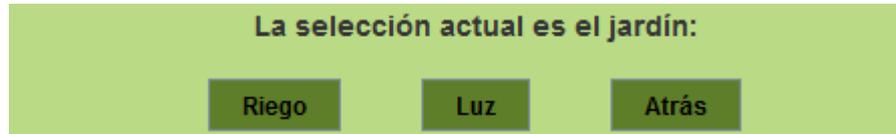


Figura 26. Panel inferior: jardín.

En esta sección se dispone de un botón para encender o apagar la luz, independientemente del sensor de presencia, y de un botón “Atrás”, que lleva al panel inferior principal.

Además incluye un botón “Riego” que al pulsarlo acciona el riego del jardín, este botón es independiente del sistema borroso del riego, que actúa según la humedad del ambiente.

Conclusiones y Trabajos Futuros

Este proyecto de final de grado ha tenido como resultado la creación de una aplicación en java amigable y sencilla que permite conocer y experimentar con la domótica, a la vez que incluye también control mediante lógica borrosa de los sistemas domóticos.

Para poder llevar a cabo este proyecto se comenzó conociendo los diferentes sistemas domóticos que existen, estudiando sus funciones, y observando cuales eran los más usados en la actualidad.

Tras comprender cuales eran las funciones que se tenían que representar en la aplicación, se empezó a pensar en el diseño de esta misma, pero en especial en lo que se refiere a la casa. Se buscaron diferentes programas y se decidió por el que se creía mejor para lo que se buscaba.

Creamos las imágenes necesarias, y se empezó con la programación de la aplicación, usando java como lenguaje de programación y eclipse como apoyo para su creación.

Dentro de la parte de programación, se hizo especial hincapié en la parte de lógica difusa, ya que era el terreno que menos se conocía. Se hizo uso de la lógica difusa para la programación de algunas funciones domóticas.

Todo este proceso ha permitido ampliar los conocimientos con las herramientas que no se conocían o no se tenía mucha experiencia, además de mejorar en aquellas que ya se conocía.

El presente Trabajo Final de Grado ha sido un proyecto enriquecedor, ha proporcionado y reforzado nuevos conocimientos.

Finalmente, las líneas de trabajo futuras son:

- La realización de test para poner a prueba la usabilidad de la aplicación.
- Integrar alguna función domótica más.
- Insertar una opción en la cual se pueda cambiar los sistemas de lógica difusa.
- Generar una casa en 3D con recorrido interno.

Summary and Conclusions

During the final project, we have developed a user-friendly Java application that allows users to understand and experiment domotic. Furthermore, it also includes fuzzy logic control for domotic systems.

In order to carry out this project, the first step was to understand the main big differences among the several domotic systems available at the time of the study, analyzing their main features and their use rates, to find out the most useful and used ones.

After understanding the functions that had to be implemented in the application, we began planning the design of it, adding special emphasis on the house issue. After checking a range of software solutions, we went for the one believed to be the best one for what was intended to do.

We created the required images and started the application development using Java and the IDE Eclipse. When it comes to the programming side of it, a special emphasis was placed on the fuzzy logic, as we were new to the subject. The fuzzy logic was used to program some domotic functions.

The process has enhanced our knowledge about new tools and some others tools we were already familiar with - but we never had much experience with them until now.

It has been a very rewarding project, as it has provided us with new insights and reinforced new knowledge.

Finally, future lines of work are:

- Test the application usability.
- Include some extra domotic features.
- Add an option to adjust the fuzzy logic systems.
- Generate a 3D house internal tour.

Bibliografía

- [1] <http://www.sweethome3d.com/>
- [2] <http://www.lawebdelprogramador.com/foros/>
- [3] <http://www.wikipedia.com>
- [4] <http://docs.oracle.com/javase/specs/jls/se7/html/index.html>
- [5] <https://www.eclipse.org/>
- [6] <http://www.solophotoshop.com/tutoriales/>
- [7] <http://www.fotografodigital.com/category/tutoriales-photoshop/>
- [8] <http://jfuzzylogic.sourceforge.net/html/index.html>

Anexo I: Código de la lógica borrosa de la sistema de cristales.

Este código pertenece al sistema borroso que se encarga de controlar la función del sistema domótico que se encargará de la opacidad de los cristales de la casa según la temperatura.

```
FUNCTION_BLOCK temp // Definición del bloque

VAR_INPUT // Variables de entrada
    temperatura : REAL; // Variable entrante de temperatura
END_VAR

VAR_OUTPUT // Variable de salida
    cristales : REAL; // Variable opacidad cristales
END_VAR

FUZZIFY temperatura // Fuzzify input variable
    TERM temp := (0, 0) (40, 1) ;
END_FUZZIFY

DEFUZZIFY cristales // Defuzzify output variable
    TERM opacidad := (0,0) (30, 0.50) (60, 1);
    METHOD : LM; // Use 'Center Of Gravity'
    DEFAULT := 0; // Default value is 0
END_DEFUZZIFY

RULEBLOCK No1
    AND : MIN; // Use 'min' for 'and'
    ACT : MIN; // Use 'min' activation method
    ACCU : MAX; // Use 'max' accumulation method

    RULE 1 : IF temperatura IS temp THEN cristales IS opacidad;
END_RULEBLOCK

END_FUNCTION_BLOCK
```

Anexo II: Código de la lógica borrosa del sistema de riego.

Este código pertenece al sistema borroso que se encarga de controlar la función del sistema domótico que se encargará del riego del jardín de la casa según la temperatura y la humedad.

```
FUNCTION_BLOCK riego // Definición del bloque

VAR_INPUT // Definición de variables de entrada
  humedad : REAL;
  temperatura : REAL;
END_VAR

VAR_OUTPUT // Definición de variables de salida
 regar : REAL;
END_VAR

// Fuzzify input variable
FUZZIFY temperatura
  TERM baja := (0, 1) (6, 1) (14, 0) ;
  TERM normal := (14, 0) (17,1) (22,1) (25,0);
  TERM alta := (25, 0) (35, 1) (50, 1);
END_FUZZIFY

// Fuzzify input variable
FUZZIFY humedad
  TERM baja := (0, 1) (20, 1) (40, 0) ;
  TERM normal := (40, 0) (50,1) (60,1) (70,0);
  TERM alta := (70, 1) (85, 0) (100, 0);
END_FUZZIFY

// Defuzzify output variable
DEFUZZIFY regar
  TERM ok := (0, 0) (50, 0.50) (100, 1);
  METHOD : LM; // Use 'Center Of Gravity' defuzzification method
  DEFAULT := 0;
END_DEFUZZIFY

RULEBLOCK No1
  AND : MIN; // Use 'min' for 'and'
  ACT : MIN; // Use 'min' activation method
  ACCU : MAX; // Use 'max' accumulation method

// Reglas del sistema
RULE 1 : IF temperatura IS baja AND humedad IS baja THEN regar IS ok;
RULE 2 : IF temperatura IS baja AND humedad IS normal THEN regar IS ok;
```

RULE 3 : IF temperatura IS baja AND humedad IS alta THEN regar IS ok;
RULE 4 : IF temperatura IS normal AND humedad IS baja THEN regar IS ok;
RULE 5 : IF temperatura IS normal AND humedad IS normal THEN regar IS ok;
RULE 6 : IF temperatura IS normal AND humedad IS alta THEN regar IS ok;
RULE 7 : IF temperatura IS alta AND humedad IS baja THEN regar IS ok;
RULE 8 : IF temperatura IS alta AND humedad IS normal THEN regar IS ok;
RULE 9 : IF temperatura IS alta AND humedad IS alta THEN regar IS ok;
END_RULEBLOCK

END_FUNCTION_BLOCK

Anexo III: Código clase padre de las secciones de la casa.

Este código pertenece a la clase que se encarga de generar en la aplicación el panel inferior, el encargado de acceder a las diferentes secciones de la casa y de mostrar las funciones de cada sección.

Dicho código es la clase padre, todas las secciones extienden de él, así cuando son llamados lo sustituyen.

```
public class Botonera {  
  
    // Constantes  
    final static int COLOR_ROJO = 95126042;  
    final static int COLOR_FONDO_FRAME = 16744499;  
    final static int COLOR_LETRA = 000000;  
  
    // Botones para acceder a cada sección  
    private JButton cocina = new JButton("Cocina");  
    private JButton baño = new JButton("Baño");  
    private JButton dormitorio = new JButton("Dormitorio");  
    private JButton salon = new JButton("Salón");  
    private JButton escalera = new JButton("Escalera");  
    private JButton jardin = new JButton("Jardín");  
  
    // Botones de las persianas  
    private JButton arriba = new JButton("↑");  
    private JButton abajo = new JButton("↓");  
  
    private JButton luz = new JButton("Luz");  
  
    private JButton atras = new JButton("Atrás");  
  
    private boolean luces = false;  
    private int persiana = 0;  
  
    // Etiquetas  
    private JLabel titulo = new JLabel("Seleccione una opción ó pulse en la casa:");  
    private JLabel Tpersiana = new JLabel("Persiana: ");  
  
    // Colores a usar  
    private Color fondo_boton = new Color(95, 126, 42);  
    private Color fondo = new Color(187, 218, 134);  
    private Color letra_boton = new Color(COLOR_LETRA);  
  
    // Panel contenedor
```

```

private JPanel contenedorBotones;

// Constructor de la clase, inicia el panel inferior
public Botonera() {
    setContenedorBotones(new JPanel());

    getContenedorBotones().setLayout(new GridLayout(2,1));
    // Paneles auxiliares
    JPanel texto = new JPanel(new FlowLayout());
    JPanel botones = new JPanel(new FlowLayout());

    getContenedorBotones().add(texto);
    getContenedorBotones().add(botones);

    // Título del panel
    getTitulo().setFont(new java.awt.Font("Dialog", 1,14));
    texto.add(getTitulo());

    // Añadimos todos los botones en la fila de abajo
    getbaño().setForeground(letra_boton);
    getbaño().setBackground(fondo_boton);
    botones.add(getbaño());

    getcocina().setForeground(letra_boton);
    getcocina().setBackground(fondo_boton);
    botones.add(getcocina());

    getcocina().setForeground(letra_boton);
    getcocina().setBackground(fondo_boton);
    botones.add(getcocina());

    getdormitorio().setForeground(letra_boton);
    getdormitorio().setBackground(fondo_boton);
    botones.add(getdormitorio());

    getEscalera().setForeground(letra_boton);
    getEscalera().setBackground(fondo_boton);
    botones.add(getEscalera());

    getSalon().setForeground(letra_boton);
    getSalon().setBackground(fondo_boton);
    botones.add(getSalon());

    getJardin().setForeground(letra_boton);
    getJardin().setBackground(fondo_boton);
    botones.add(getJardin());

    // Ponemos los colores
    texto.setBackground(fondo);
    botones.setBackground(fondo);

```

```

        getContenedorBotones().setBackground(fondo);
    }

    // Métodos para subir y bajar las persianas
    public void arriba() {
        if (getPersiana() != 0) {
            persiana -= 1;
        }
    }

    public void abajo() {
        if (getPersiana() < 60) {
            persiana += 1;
        }
    }

    public void arriba2() {
        if (getPersiana() != 0) {
            persiana -= 10;
        }
    }

    public void abajo2() {
        if (getPersiana() < 60) {
            persiana += 10;
        }
    }

    //Método para encender o apagar la luz según este.
    public void luz() {
        if(isLuces()) {
            setLuces(false);
        } else {
            setLuces(true);
        }
    }

    // Getters y setters de los atributos de la clase

    public void ...
    .
    .
    .
    .
    .
    .
    }

}

```

Anexo IV: Código de la clase encargada del panel lateral.

Este código pertenece a la clase que se encarga de generar en la aplicación el panel lateral.

```
public class Botonera_ext {  
  
    //Constantes  
    final static int COLOR_FONDO_FRAME = 16744499;  
    final static int COLOR_LETRA = 000000;  
    final static int COLOR_ROJO = 14233600;  
    final static int DISTANCIA_LATERAL_GRIDLAYOUT = 2;  
    final static int DISTANCIA_GRIDLAYOUT = 5;  
    final static int COLUMNAS_GRIDLAYOUT = 1;  
    final static int FILAS_GRIDLAYOUT = 20;  
  
    //Atributos  
    private JLabel titulo;  
  
    private JSlider temperatura_slider = new JSlider (JSlider.HORIZONTAL,-  
10,40,20);  
    private JLabel temperatura_label;  
  
    private JSlider humedad_slider = new JSlider (JSlider.HORIZONTAL);  
    private JLabel humedad_label;  
  
    private JSlider viento_slider = new JSlider (JSlider.HORIZONTAL);  
    private JLabel viento_label;  
  
    private JLabel tiempo;  
    private JButton mas;  
    private JButton menos;  
  
    private JLabel panelInterior;  
    private JLabel persianas = new JLabel("Persianas:"); ;  
  
    private JButton subirPersianas = new JButton("↑");  
    private JButton bajarPersianas = new JButton("↓");  
  
    private int cielo = 0;  
    private ImageIcon cielo_img;  
  
    private JPanel contenedorBotones;  
  
    // Constructor de la clase, inicia el panel lateral  
    public Botonera_ext () {
```

```

Color letra_boton = new Color(COLOR_LETRA);
Color fondo_boton = new Color(95, 126, 42);
Color fondo = new Color(187, 218, 134);

setContenedorBotones (new JPanel (new GridLayout
(FILAS_GRIDLAYOUT, COLUMNAS_GRIDLAYOUT, DISTANCIA_GRIDLAYOUT,
DISTANCIA_GRIDLAYOUT)));
setTitulo(new JLabel("Panel Exterior:"));

getTitulo().setHorizontalAlignment(SwingConstants.CENTER);
getTitulo().setFont(new java.awt.Font("Dialog", 1, 14));
getContenedorBotones().add(getTitulo());

setTiempo(new JLabel("Tiempo:"));
getContenedorBotones().add(getTiempo());

JPanel auxiliar = new JPanel(new GridLayout (1,2,10,10));
getContenedorBotones().add(auxiliar);
auxiliar.setBackground(fondo);
setMenos(new JButton("←"));
getMenos().setForeground(letra_boton);
getMenos().setBackground(fondo_boton);
auxiliar.add(getMenos());
setMas(new JButton("→"));
getMas().setForeground(letra_boton);
getMas().setBackground(fondo_boton);
auxiliar.add(getMas());

setTemperatura_label(new JLabel("Temperatura exterior: 20 °C"));
getContenedorBotones().add(getTemperatura_label());
getTemperatura_slider().setBackground(fondo);
getContenedorBotones().add(getTemperatura_slider());

setHumedad_label(new JLabel("Humedad exterior: 20 %"));
getContenedorBotones().add(getHumedad_label());
getHumedad_slider().setValue(20);
getHumedad_slider().setMaximum(100);
getHumedad_slider().setForeground(letra_boton);
getHumedad_slider().setBackground(fondo);

getContenedorBotones().add(getHumedad_slider());

setViento_label(new JLabel("Viento exterior: 20 km/h"));
getContenedorBotones().add(getViento_label());
getViento_slider().setValue(20);
getViento_slider().setMaximum(200);
getViento_slider().setBackground(fondo);
getContenedorBotones().add(getViento_slider());

```

```

setPanelInterior(new JLabel("Panel Interior:"));

getPanelInterior().setHorizontalAlignment(SwingConstants.CENTER);
getPanelInterior().setFont(new java.awt.Font("Dialog", 1, 14));
getContenedorBotones().add(getPanelInterior());

getContenedorBotones().add(getPersianas());

JPanel auxiliar2 = new JPanel(new GridLayout (1,2,10,10));
getContenedorBotones().add(auxiliar2);
auxiliar2.setBackground(fondo);

getSubirPersianas().setForeground(letra_boton);
getSubirPersianas().setBackground(fondo_boton);
auxiliar2.add(getSubirPersianas());

getBajarPersianas().setForeground(letra_boton);
getBajarPersianas().setBackground(fondo_boton);
auxiliar2.add(getBajarPersianas());

cambiarCielo();

getContenedorBotones ().add (new JLabel (" "));
getContenedorBotones().setBackground(fondo);
}

//Método para cambiar la imagen del cielo.
public void cambiarCielo () {
    if (getCielo() == 0) {
        setCielo_img(new ImageIcon("img/cielo/cielo2.png"));
        cambiarValores(20, 20, 5);
    } else if (getCielo() == 1) {
        setCielo_img(new ImageIcon("img/cielo/cielo3.png"));
        cambiarValores(28, 20, 1);
    } else if (getCielo() == 2) {
        setCielo_img(new ImageIcon("img/cielo/nuboso.png"));
        cambiarValores(18, 20, 20);
    } else if (getCielo() == 5) {
        setCielo_img(new ImageIcon("img/cielo/noche1.png"));
        cambiarValores(15, 20, 10);
    } else if (getCielo() == 4) {
        setCielo_img(new ImageIcon("img/cielo/noche2.png"));
        cambiarValores(17, 20, 4);
    } else if (getCielo() == 3) {
        setCielo_img(new ImageIcon("img/cielo/noche3.png"));
        cambiarValores(14, 20, 40);
    }
}
}

```

//Método para cambiar los valores de los slider de las variables exteriores a la casa, le pasamos t para temperatura, h para humedad y v para el viento.

```
public void cambiarValores(int t, int h, int v) {  
    getTemperatura_slider().setValue(t);  
    getTemperatura_label().setText("Temperatura exterior: " +  
getTemperatura_slider().getValue() + "°C");  
    getHumedad_slider().setValue(h);  
    getHumedad_label().setText("Humedad exterior: " +  
getHumedad_slider().getValue() + "%");  
    getViento_slider().setValue(v);  
    getViento_label().setText("Viento: " + getViento_slider().getValue() + " km/h");  
}  
// Getters y setters de los atributos de la clase
```

```
public void ...
```

```
.  
.   
.   
.   
.   
.   
.   
}
```

```
}
```

