

DOMOTICA HIBRIDA INALAMBRICA (Jeedom, Z-Wave, WiFi y Open Source)

Manuel Calero Herruzo
Grado de Tecnologías de Telecomunicación
Sistemas embebidos

Jordi Bécares Ferrés
Pere Tuset Peiró

13 de enero 2019



Esta obra está sujeta a una licencia de [Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>DOMOTICA HIBRIDA INALAMBRICA (Jeedom, Z-Wave, WiFi y Open Source)</i>
Nombre del autor:	<i>Manuel Calero Herruzo</i>
Nombre del consultor:	<i>Jordi Bécares Ferrés</i>
Nombre del PRA:	<i>Pere Tuset Peiró</i>
Fecha de entrega (mm/aaaa):	<i>01/2019</i>
Titulación:	<i>Grado Tecnologías de la Telecomunicación</i>
Área del trabajo final:	<i>Sistemas encastrados</i>
Idioma del Trabajo:	<i>Castellano</i>
Palabras clave	<i>Domótica, Jeedom, Z-Wave, libre y código abierto</i>

Resumen del Trabajo

El objetivo principal del este proyecto es implantar un sistema domótico usando tecnologías sin hilos y libre, por ello, como protocolo de comunicación, se ha decidido usar Z-Wave y WiFi. Además, se pretende que este proyecto sea una base de diseño, para un sistema domótico abierto, flexible, escalable y de bajo coste. Así, que se va a centrar en la integración de distintos elementos de diferentes naturalezas, tanto elementos comerciales, como elementos de hardware y software libre.

Para conseguir el cumplimiento de dichos objetivos se va a utilizar, referido a elementos comerciales el MultiSensor 6 que controla temperatura, humedad, luminosidad, presencia, vibraciones y rayos ultravioleta. Por otro lado, en cuanto a los elementos libres se usará, como controlador domótico el software de código abierto Jeedom y como dispositivos, se diseñarán usando hardware libre como Raspberry pi, Z-Uno y MSP432.

Con dichos elementos, se pretende desarrollar una maqueta de actuadores y sensores, para controlar distintos puntos de luz (2 comedor, 1 pasillo, 1 habitación), distintas persianas monitorizadas (3 comedor, 1 habitación), temperatura, humedad, presencia y luminosidad de las distintas instancias.

A todo lo anterior, se le añadirá la inteligencia necesaria, para tomar decisiones personalizadas como si hay presencia en el pasillo enciende la luz de éste, si es de día y a una determinada hora se sube las persianas, si la habitación baja de cierta temperatura enciende el radiador. Con todo ellos, se habrá diseñado un sistema domótico mínimo y funcional, demostrando el gran potencial que ofrecen las distintas plataformas abiertas.

Abstract

The main objective of this project is to implement a home automation system using wireless and open source technologies, therefore, as a communication protocol, it has been decided to use Z-Wave and WiFi. In addition, this project is intended to be a design basis for an open, flexible, scalable and low-cost home automation system. Thus, it is focused on the integration of different elements of various natures, both commercial elements, such as elements open source.

To achieve the fulfillment of these objectives, it will be used as commercial elements, the MultiSensor 6 that controls temperature, humidity, luminosity, presence, vibrations and ultraviolet rays. On the other hand, as for the open source elements, it will be used as a domotic controller Jeedom software and as devices, they will be designed using open hardware such as Raspberry pi, Z-Uno and MSP432.

With these elements, it is intended to develop a model of actuators and sensors, to control different points of light (2 dining room, 1 corridor, 1 room), different monitored blinds (3 dining room, 1 room), temperature, humidity, presence and brightness of the different instances.

To all the above, will be added the necessary intelligence, to make decisions as if there is a presence in the hall, lights the light, a certain time, up the blinds, if the room falls of a certain temperature turn on the radiator. Finally, it will has been designed a minimal and functional domotic system, demonstrating the great potential offered by the different open source platforms.

Índice

FICHA DEL TRABAJO FINAL.....	3
Abstract.....	4
Lista de figuras	7
Lista de Tablas	9
Capítulo 1: Introducción	10
1.1 Contexto y justificación del proyecto	10
1.2 Descripción del proyecto	11
1.3 Objetivos Técnicos	13
1.3.1 Objetivos principales	13
1.3.2 Objetivos extraordinarios	13
1.4 Objetivos transversales	13
1.4 Enfoque y método seguido	14
1.5 Planificación de proyecto.....	15
1.6 Recursos empleados	19
1.7 Productos obtenidos	19
1.8 Breve descripción de los otros capítulos de la memoria	20
Capítulo 2 antecedentes	21
2.1 Estado del arte.....	21
2.1.1 Topologías de red	22
2.1.2 Tecnologías inalámbricas	24
2.1.2.1 ZigBee	24
2.1.2.2 WiFi	25
2.1.2.3 Z-Wave.....	26
2.1.3 Plataformas de desarrollo	27
2.1.3.1 Plataformas de desarrollo Z-Wave.....	28
2.1.3.2 Plataformas de desarrollo WiFi	28
2.2 Estudio de mercado.....	29
2.2.1 Controladores	29
2.2.1.1 Definición de controlador doméstico.....	29
2.2.1.2 Controladores en el mercado.....	29

2.2.2 Dispositivos	31
2.2.2.1 dispositivos para desarrollar	31
2.2.2.2 dispositivos Comerciales.....	33
Capítulo 3. Descripción funcional.....	36
3.1 Diagrama de bloques del sistema	36
3.2 Propuesta de arquitectura	37
3.3 Propuesta de protocolo inalámbrico	37
3.4 Propuesta de controladora	38
3.5 Propuesta de dispositivo y desarrollos	39
3.5.1 Dispositivo comercial	39
3.5.2 Dispositivo Basado en Z-Uno	39
3.5.3 Dispositivo basado en Raspberry pi	40
3.5.6 Dispositivo basado en MSP432 y CC3100	41
3.6 Sistema de tomas de decisiones (Escenas).....	41
Capítulo 4	42
4.1 Propuesta de controladora y redes Z-Wave WiFi.....	42
4.2 Dispositivo comercial	44
4.3 Dispositivo basado en Z-Uno.....	45
4.4 Dispositivo basado en Raspberry pi	47
4.5 Dispositivo basado en MSP432P401R+CC3100	51
4.4 Sistema de tomas de decisiones (Escenas).....	53
Capítulo 5 Viabilidad técnica.....	55
Capítulo 6 Valoración económica.....	56
Capítulo 7 conclusiones	59
7.1. Una descripción de las conclusiones	60
7.2. Autoevaluación.	61
7.3. Líneas futuras	63
Glosario	64
Bibliografía.....	66

Anexo	70
Anexo I: Instalación de Jeedom.....	70
Anexo II Proceso de creación de certificado	71
Anexo III Flujo Espera persiana.....	72
Anexo IV Flujo persianas amanecer atardecer	72
Anexo V Flujo luz pasillo.....	73
Anexo VI Flujo calefacción instancia	73
Anexo VII Flujo calefacción general	74
Anexo VIII Flujo Alarma	74

Lista de figuras

Figura 1: Plano casa con distribución dispositivos.	12
Figura 2: Planificación inicial	16
Figura 3: Planificación final.....	18
Figura 4: Áreas de actuación domótica	22
Figura 5: Arquitectura centralizada4.....	23
Figura 6: Arquitectura descentralizada4	23
Figura 7: Arquitectura distribuida4.....	23
Figura 8: Arquitectura Híbrida / Mixta4	24
Figura 9: Modelo OSI ZigBee	25
Figura 10: Figura comparativa Z-wave, ZigBee y WiFi	27
Figura 11: Z-Uno.....	31
Figura 12: Raspberry pi	32
Figura 13: MSP432P401R.....	32
Figura 14: Shelly 2.....	33
Figura 15: Shelly humedad y temperatura	33
Figura 16: Neo	33
Figura 17: Shelly Sense.....	33

Figura 18: USB Z-Stick GEN5	34
Figura 19: MultiSensor 6.....	34
Figura 20: RaZberry.....	34
Figura 21: Doble Relé Zipato Plus.....	34
Figura 22: Detector de Moho POPP	35
Figura 23: Detector movimiento Hank	35
Figura 24 Sensor de luz Everspring	35
Figura 25: Diagrama de bloques general del sistema.....	36
Figura 26: Diagrama controlador domótico	38
Figura 27: Diagrama funcional sensor comercial MultiSensor 6	39
Figura 28: Diagrama funcional dispositivo desarrollado Z-Uno.....	40
Figura 29: Diagrama funcional dispositivo desarrollado Raspberry	40
Figura 30: Diagrama funcional dispositivo desarrollado MSP432+CC3100	41
Figura 31: Esquema eléctrico dispositivo desarrollado Z-Uno (X3 indica que hay dos más) .	45
Figura 32: DHT22	46
Figura 33: BH1750.....	46
Figura 34: HC-SR501 PIR	46
Figura 35: Comparación sin usar Virtuel (izquierda) usando Virtuel (derecha)	47
Figura 36: Esquema eléctrico dispositivo desarrollado Raspberry pi (X3 indica que hay dos más).....	48
Figura 37: Esquema eléctrico dispositivo desarrollado MSP432P401R+CC3100 (X3 indica que hay dos más)	51
Figura 38: interruptores virtuales para escenas	54
Figura 39: Cuadro de mandos final	61
Figura 40: Flujo Espera persiana.....	72
Figura 41: Flujo persianas amanecer atardecer.....	72
Figura 42: Flujo luz pasillo	73
Figura 43: Flujo calefacción instancia.....	73

Figura 44: Flujo calefacción general.....	74
Figura 45: Flujo Alarma	74

Lista de Tablas

Tablas 1: Presupuesto proyecto filosofía libre.....	57
Tablas 2: Presupuesto equivalente con elementos comerciales	58

Capítulo 1: Introducción

1.1 Contexto y justificación del proyecto

En la actualidad, se potencia el ahorro y la eficiencia energética, además, hay una alta preocupación por la contaminación, la producción de energía, la seguridad y el cuidado personal. Muchas de las soluciones, para controlar dichas preocupaciones, se basan en el concepto internet de las cosas (IoT)¹, refiriéndose a la interconexión digital de objetos cotidianos con internet, dicho concepto hoy en día está muy en auge. Por ello, se está imponiendo todas las soluciones, que ofrecen una conexión y comunicación total, entre ellas las soluciones domóticas.

¿Por qué la domótica? básicamente, porque gracias a todas las posibilidades que ofrece, se consume de una manera más eficiente, con el consecuente ahorro de dinero y energía que esto supone, ganando en confort y personalización de entornos y tareas. Gracias a la domótica, se puede activar la luz solo cuando se detecta presencia, subir la persiana remotamente o a una hora deseada, vigilar el hogar e incluso simular que éste no está vacío en periodos de ausencia.

También, gracias a la aparición de protocolos sin hilos como WiFi y Z-Wave, la instalación del sistema es menos costosa, mucho más sencilla, estéticamente integrada, a la vez que se gana en flexibilidad para su redimensionamiento. A pesar de todo, los dispositivos inalámbricos, siguen sin ser todo lo económicos que se pueden desear. Por ello, este proyecto, con distintas tecnologías, comerciales y de hardware y software libre/abierto, se centrará en desarrollar varios dispositivos inalámbricos, que permitan control de luces y persianas, temperatura, humedad, presencia, etc. Combatiendo los problemas históricos, de la domótica, como son:

- **Alto coste:** Se requería una gran inversión, ya que no solo los componentes del sistema eran caros, sino que también requieren obras mayores para su instalación.
- **Poca flexibilidad:** Los sistemas permitían realizar pocas acciones, eran cerrados y poco configurables.
- **Poco escalable:** Introducir un nuevo elemento domótico no era sencillo, de hecho, habitualmente había que invertir en cableado y nueva obra.

¹ (wikipedia, 2018) https://es.wikipedia.org/wiki/Internet_de_las_cosas

1.2 Descripción del proyecto

Este proyecto, diseña e implementa un sistema domótico sin hilos, abierto, flexible, escalable y económico. Esto, se logrará integrando una amplia variedad de tecnologías, tanto comerciales como libres. El sistema, se basará en una arquitectura centralizada, donde habrá un controlador principal. Dicho controlador, consiste en una máquina virtual con el sistema operativo Debian y un software domótico de código abierto Jeedom². Como protocolos de comunicación sin hilos, se usará Z-Wave y WiFi, por lo que dicha máquina requiere conectado a ella, un dispositivo USB, que hará de interfaz Z-Wave y de la propia tarjeta WiFi de la máquina real, que hará de interfaz WiFi.

En cuanto a los dispositivos, se desarrollan de dos tipos Z-Wave y WiFi. Dentro de los dispositivos Z-Wave, se tiene un sensor comercial MultiSensor 6, un sensor desarrollado basado en Z-Uno y otro dispositivo RaZberry, que conectado a una Raspberry pi, convierte está en un controlador Z-Wave, esta parte solo queda planteada, pero no es funcional en este proyecto, debido a un fallo de concepto y planificación descubierto (apartado [1.5](#), [5](#) y [7](#)). Por parte, de los dispositivos WiFi, se tiene una Raspberry pi y las placas de Texas instruments MSP432P401R y CC3100.

Con el dispositivo Z-Wave MultiSensor 6 (comercial) se controla, la temperatura, humedad, luminosidad, rayos ultravioletas, vibraciones y presencia del pasillo de la casa. Con el Z-Uno, que es una microcontroladora programable basado en Arduino, pero con interfaz Z-Wave, y los sensores y electrónica necesaria se controla, dos puntos de luz del salón, un punto de luz del pasillo y tres persianas del salón, además de, temperatura, luminosidad, presencia y humedad del salón.

Respecto los dispositivos WiFi, por un lado, está la Raspberry pi, que es un ordenador de placa simple de tamaño reducido, al cual, también se le añaden los sensores y la electrónica necesaria para controlar un punto de luz, una persiana, temperatura, luminosidad, presencia y humedad de la habitación principal. Dicha Raspberry, también tendrá instalado Jeedom y el dispositivo RaZberry, el cual, es un añadido que le agrega una interfaz Z-Wave y la capacidad de ser un controlador, al que poder añadir dispositivos Z-Wave en un futuro. Por otro lado, están las placas MSP432P401R y CC3100, la primera una microcontroladora programable y la segunda un añadido a ésta, para ofrecerle un interfaz WiFi, que junto a la electrónica y

² (jeedom, 2018) <https://www.jeedom.com/site/fr/>

sensores que se añadirán, tendrá capacidad para controlar un punto de luz, una persiana, temperatura, luminosidad, presencia y humedad de la habitación secundaria.

Todos estos dispositivos, son manejados desde el controlador principal con el software Jeedom, dicho software ofrece entre otras cosas una interfaz web de control y configuración para el sistema domótico. De tal forma, que este software incorpora la inteligencia necesaria para la toma de decisiones. Estas decisiones, se configurarán mediante una serie de reglas, que Jeedom llama escenas, dando estas escenas lugar a que las persianas se levanten o cierren a cierta hora, que si se detecta presencia en el pasillo y hay poca luminosidad en éste se encienda la luz, si la temperatura individual de una habitación baja de una cifra se encienda el radiador de dicha habitación, que si la temperatura media del hogar baja de cierta cifra se encienda todos los radiadores del hogar, si se activa la alarma y se detecta algún tipo de presencia se mande un mensaje por telegram. Todas estas escenas, dispondrán de un botón virtual para activarlas o desactivarlas, según el deseo del usuario. También, destacar que Jeedom, será accesible vía web desde internet y de forma segura.

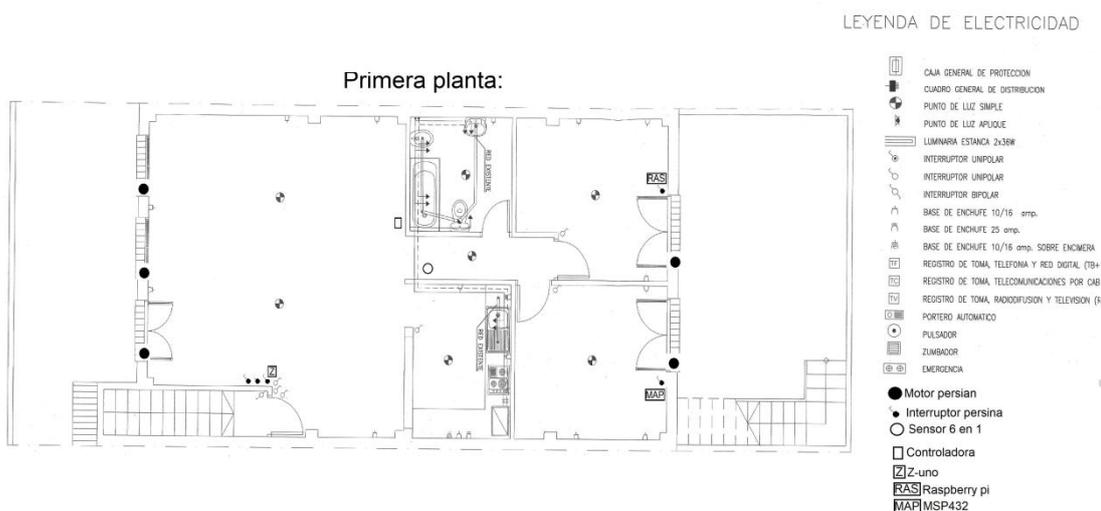


Figura 1: Plano casa con distribución dispositivos.

1.3 Objetivos Técnicos

1.3.1 Objetivos principales

Los objetivos del modelo domótico que se va a diseñar son:

1. Despliegue de una red Z-Wave.
2. Despliegue de la plataforma Jeedom.
3. Integración de dispositivos en la plataforma Jeedom.
4. Despliegue dispositivo comercial Z-Wave.
5. Sistema de control de luz, persiana y calefacción basado en el dispositivo Z-Uno y protocolo Z-Wave.
6. Integración del conjunto Raspberry pi 3 y RaZberry.
7. Sistema de control de luz, persiana y calefacción basado en Raspberry pi y Z-Wave.
8. Sistema de control de luz, persiana y calefacción basado en Raspberry pi y WiFi.

1.3.2 Objetivos extraordinarios

El modelo domótico, que se va a diseñar, tiene como objetivos extraordinarios:

1. Monitorización en los puntos de control de temperatura, humedad, presencia y luminosidad.
2. Sistema automático para:
 - 2.1. Activación luz pasillo.
 - 2.2. Persianas
 - 2.3. Calefacción individual por instancias y general.
 - 2.4. Alarma
3. Integración del módulo ZM5304 en la placa MSP432
4. Desarrollo de dispositivo propio con tecnología Z-Wave basado en MSP432.
5. Control de luces, persiana y calefacción, temperatura, humedad, presencia y luminosidad mediante las placas MSP432 y CC3100 usando WiFi.

1.4 Objetivos transversales

Los objetivos transversales, no técnicos, que se pretenden alcanzar en el proyecto son dos:

1. Diseñar un sistema domótico de bajo coste, flexible, escalable y fácil de instalar.
2. Demostrar el potencial y los beneficios que ofrece usar elementos libres o de código abierto.

1.4 Enfoque y método seguido

Este proyecto, tiene un enfoque práctico y de uso de elementos libres y por definición abierto. En él, se busca un sistema domótico, flexible y escalable, y gracias a ello, lograr que la implementación del sistema domótico tenga una reducción del coste final o en su defecto poder instalar un sistema mínimo, pero que en el futuro se le puedan añadir nodos fácilmente. Por ello, este proyecto integra y desarrolla distintos dispositivos, de naturalezas muy distintas, desde comerciales a libres o de código abierto, para demostrar el gran potencial de esta filosofía. Para su comunicación, se usan protocolos inalámbricos, buscando favorecer la flexibilidad de instalación física del sistema.

Se elige este enfoque, en vez de uno totalmente comercial, ya que se considera que al utilizar hardware y software abierto se logra una considerable ventaja, a la hora de integrar nodos de distintas naturalezas, ya sea esta naturaleza comercial o libre. Pero, lo más destacado, es que gracias a la flexibilidad y escalabilidad que ofrece este enfoque, se pueden realizar diseños más abiertos, según las necesidades de cada sistema y la distribución que cada hogar tenga. Además, no se depende de un solo fabricante el cual puede dejar de fabricar o variar el coste de los elementos que interesan, siendo necesaria una migración de todo el sistema. Todo esto, se traduce en un sistema menos costoso, tanto para su implementación inicial, como para su ampliación.

Durante la realización del proyecto, se establecen ciertos hitos, los cuales, se van a organizar en tres fases y que van de menor a mayor dificultad. Por ello, se inicia el proyecto con la integración de un dispositivo comercial y se avanza hacia el desarrollo de dispositivos basados en hardware y software libre o abierto. Primeramente, se desarrollan las acciones más simples, como es el manejo simple de un relé para encender y apagar las luces, para continuar con acciones más complejas, como son el control de persianas. Una vez, logrados todos los objetivos con todas las variantes de hardware, se procede a ampliar sus funciones, añadiéndoles más sensores y finalizando con la comunicación de todos ellos para tomar decisiones y personalizar el sistema domótico. Durante todo el desarrollo y planificación se consideran y controlan los plazos de cada hito. Así, conforme estos hitos se puedan cumplir, se

trabajaré para mejorar el rendimiento o prestación de lo desarrollado o, por el contrario, se simplificará el desarrollo si se detecta riesgo de no cumplir los plazos planificados.

1.5 Planificación de proyecto

Para la planificación del proyecto, se han previsto tareas planteadas en tres fases independientes. También, se han considerado ciertas horas reservadas para contingencias y ampliaciones. Éstas, son horas para afrontar retrasos debidos a riesgos, o cambios de planificación, para asignar a tareas en las que se haya detectado necesidad de refuerzo y en el caso de que sobren, se usarán para la fase final del proyecto.

Así, en la planificación se han considerado días de 5 horas, respetando fin de semanas y festivos. Inicialmente, la planificación se plantea como la que se puede ver en el siguiente diagrama de Gantt:

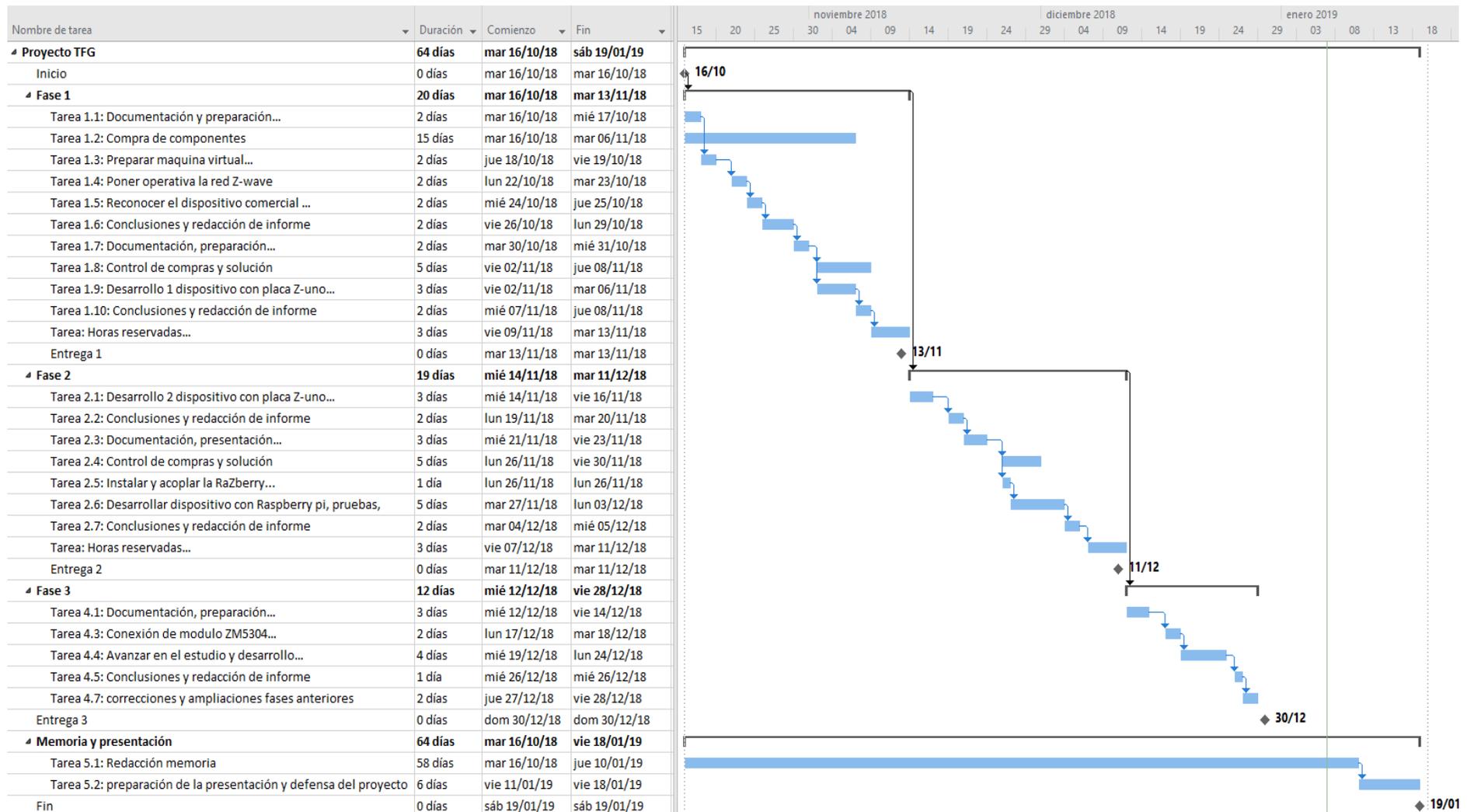


Figura 2: Planificación inicial

Esta planificación, se pudo seguir hasta el punto 2.6 (figura [2](#)), ya que en el punto 2.3 se detectaron varios fallos del planteamiento inicial, los cuales se detallarán en el punto [5](#) y [7](#). Por un lado, se detectó que no era posible desarrollar el dispositivo Z-Wave basado en Raspberry pi con módulo RaZberry y por otro, se comprobó que la integración del módulo ZM5304 en la MSP432 y el desarrollo de un dispositivo Z-Wave basado en ello, podía complicarse debido a los medios disponibles y no ser objetivo de este proyecto, si no de uno aparte. Así, gracias a las horas reservadas contempladas, se añadieron otros objetivos y tareas, modificando la planificación para quedar de la siguiente forma:

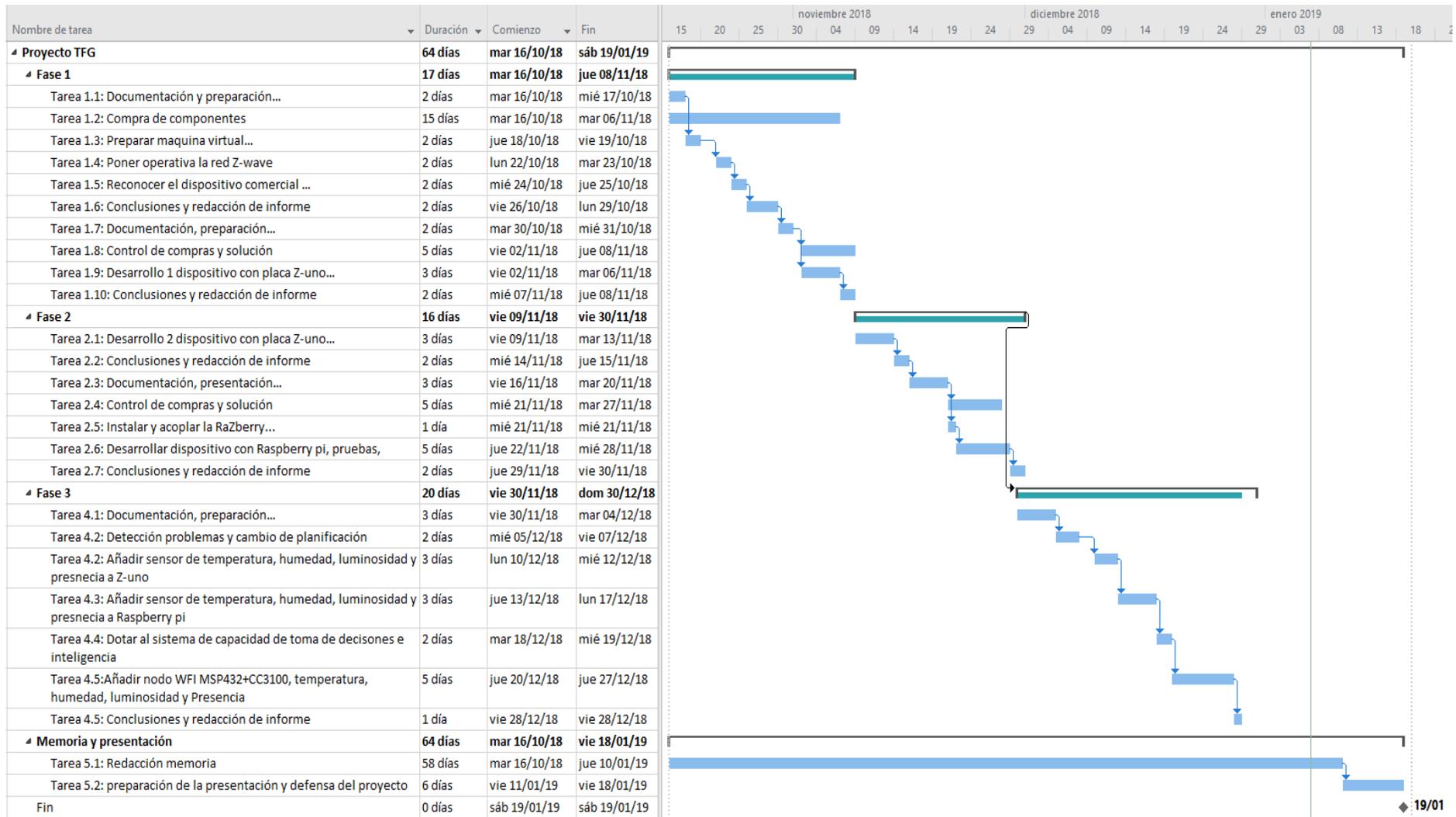


Figura 3: Planificación final

1.6 Recursos empleados

Durante el proyecto, se han utilizado una serie de hardware y software que se detalla a continuación:

- Software de desarrollo:
 - IDE Energía e IDE Arduino.
- Software controlador:
 - Jeedom, VirtualBox.
- Software de terceros:
 - Librería para sensor BH1750 de nombre BH1750, librería para el sensor DHT22 de nombre DHT11 y librería asociada a la DHT11 de nombre Adafruit_Sensors.h7
- Hardware controlador:
 - Máquina virtual Debian, Adaptador USB Aeotec GEN5, tarjeta WiFi y RaZberry.
- Hardware desarrollo dispositivos:
 - Z-Uno, Raspberry pi 3 B+, SD para Raspberry, alimentador Raspberry, MSP432 y CC3100.
- Hardware usados como sensor:
 - Sensor temperatura y humedad DHT22, sensor de luz bh1750 y sensor de presencia Neuftech HC-SR501 PIR.
- Hardware actuador:
 - Módulos relés.
- Otros hardware necesarios para las conexiones:
 - Resistencias, cables y alimentadores USB.

1.7 Productos obtenidos

A la finalización de este proyecto, se ha obtenido:

1. Memoria con los detalles del desarrollo.
2. Dos redes de sensores, una Z-Wave otra WiFi.
3. Un controlador domótico, basado en un software de código abierto, Jeedom.
4. Despliegue para el pasillo de un sensor comercial MultiSensor 6 ZW100-C de Aeotec, temperatura, presencia, humedad, vibración, rayos ultravioleta y luminosidad.

5. Desarrollo de un dispositivo basado en Z-Uno y red Z-Wave que controla tres persianas, tres puntos de luz, temperatura, humedad, luminosidad, presencia y calefacción del salón.
6. Desarrollo de un dispositivo basado en Raspberry pi y red WiFi que controla una persiana, un punto de luz, temperatura, humedad, luminosidad, presencia y calefacción, de la habitación 1.
7. Desarrollo de un dispositivo basado en MSP432, CC3100 y red WiFi que controla una persiana, un punto de luz, temperatura, humedad, luminosidad, presencia y calefacción, de la habitación 2.
8. Reglas para automatismos de hogar según datos.
9. Códigos y Scripts necesarios para los distintos desarrollos.

1.8 Breve descripción de los otros capítulos de la memoria

Seguidamente, en el capítulo [2](#), se realizará un análisis desde el punto de vista teórico de las distintas tecnologías y protocolos que se usan o podrían usar en el proyecto. Se compararán las distintas opciones y se explicará cada decisión tomada. También, se hará un estudio de los dispositivos existentes en el mercado para resolver las mismas situaciones que se plantean a lo largo de este proyecto.

Respecto al capítulo [3](#), se explica la implementación y decisiones de diseño tomadas, en cada uno de los dispositivos. Estas explicaciones, serán a alto nivel, solo centrándose en las funciones y diagrama de bloques de cada dispositivo a desarrollar, junto a los protocolos de comunicación y sensores que usan.

En el capítulo [4](#), se explicará, con mayor detalle lógico y funcional, todas las implementaciones citadas en el capítulo 3. También, se expondrán los puntos interesantes y destacados del desarrollo de cada dispositivo. Así, se empezará con la implementación del controlador domótico, se continuará con el despliegue de la red Z-Wave, para dar paso a cada dispositivo integrado y desarrollado (comercial, Z-Uno, Raspberry pi y MSP432 junto con CC3100), para finalizar, con la explicación de las lógicas programadas en el sistema, para lograr tomar decisiones mediante los datos obtenidos.

En el capítulo [5](#), se hablará de la viabilidad del proyecto, barreras a dicha viabilidad que se han tenido que superar, porqué han aparecido y cómo se han solucionado.

En el capítulo 6, se realizarán dos valoraciones económicas, una la valoración económica de este proyecto usando una filosofía de código abierto y libre y otra una valoración de un sistema equivalente, diseñado a partir de elementos completamente comerciales, para finalizar el capítulo se compararán ambas valoraciones.

Para finalizar, en el capítulo 7 se comentarán los objetivos que se han alcanzado y cuáles no junto con el porqué de ello, se expondrá que se ha trabajado y logrado con este proyecto, se realizara una autovaloración, y se acabara citando futuras mejoras que se pueden realizar en la línea de este proyecto.

Capítulo 2 antecedentes

2.1 Estado del arte

Según Wikipedia³ “Se llama domótica a los sistemas capaces de automatizar una vivienda o edificación de cualquier tipo, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar. Se podría definir como la integración de la tecnología en el diseño inteligente de un recinto cerrado”. Un sistema domótico doméstico, se compone de varios sensores distribuidos estratégicamente por la vivienda y normalmente un controlador que es capaz de recoger la información proveniente de los sensores, procesarla y emitir órdenes a unos actuadores.

Las áreas de actuación de la domótica dentro de una vivienda se pueden agrupar en cinco bloques, que son:

- **Gestión energía:** trata todo lo referido a iluminación, climatización, agua caliente sanitaria, riego, electrodomésticos, ventanas, persianas, etc.
- **Accesibilidad:** facilita por diversos medios y formas la vida en la vivienda de las personas con discapacidad.
- **Seguridad:** Gestionaría tanto la vigilancia automática de personas, animales y bienes, como las incidencias y averías que pueden surgir en el domicilio.

³ (wikipedia, 2017) <https://es.wikipedia.org/wiki/Dom%C3%B3tica>

- **Confortabilidad:** convierte la vivienda en un hogar más confortable, a través de la gestión de dispositivos y actividades domésticas, por ejemplo, intensidad de luz, climatización óptima, confort de una sala de audiovisuales, etc.
- **Comunicación:** todo lo referido a comunicación tanto entre dispositivos, como remota. La vivienda domótica, puede permite el acceso a internet y el intercambio entre dispositivos, acceso a servicios de telefonía IP, etc.



Figura 4: Áreas de actuación domótica

Históricamente, los sistemas domóticos eran altamente costosos de implementar, ya que los protocolos de comunicación requerían cableado y, además, el hardware existente era poco flexible y propietario. Esto, implicaba fuertes inversiones en obra doméstica, con el gasto que esto supone, sin embargo, en los últimos años, han aparecido nuevas tecnologías y protocolos vía radio, que ayudan a la expansión y accesibilidad de los sistemas domóticos.

Aunque en la actualidad, hay multitud de variantes y posibilidades para diseñar un sistema domótico, este proyecto, se va a centrar en la domótica sin hilos, por eso, primeramente, se expondrán diferentes topologías de red a usar, se continuará con diferentes protocolos y tecnologías de comunicación sin hilos, para pasar a explicar distintas plataformas de desarrollo existentes, no expresamente de domótica, sino de electrónica con la cual se puede diseñar un sistema domótico.

2.1.1 Topologías de red

Según la documentación⁴ “La arquitectura de los sistemas hace referencia a la manera en que están organizados sus componentes” las posibles arquitecturas son:

⁴ (domoticaudem, 2015) <https://domoticaudem.wordpress.com/arquitectura-de-los-sistemas/>

- **Arquitectura centralizada:** Este sistema está organizado de tal forma que el controlador sea el que recibe la información de los sensores, la analiza y enviando una orden a los actuadores.

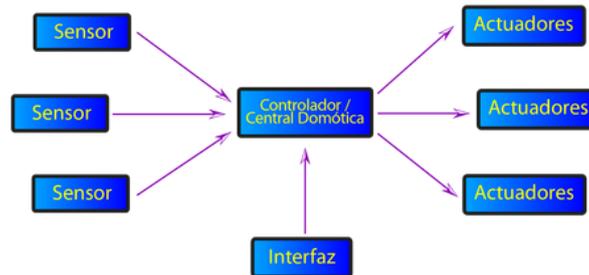


Figura 5: Arquitectura centralizada⁴

- **Arquitectura descentralizada:** En un sistema descentralizado existen varios controladores, conectados a sensores y actuadores, quienes a su vez están interconectados por medio de un bus:

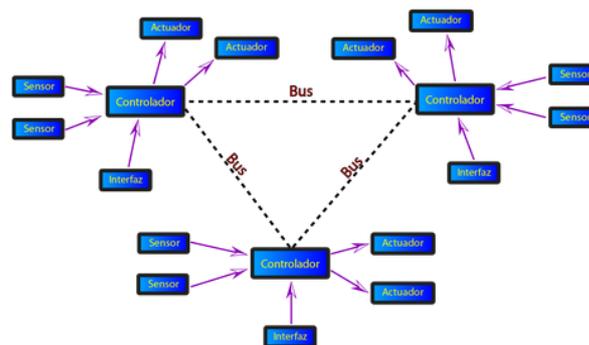


Figura 6: Arquitectura descentralizada⁴

- **Arquitectura distribuida:** Este tipo de arquitectura se diferencia por tener sensores y actuadores que son a su vez controladores, es decir, son capaces de analizar la información y están conectados a través de un bus central.



Figura 7: Arquitectura distribuida⁴

- **Arquitectura Híbrida / Mixta:** Se combinan las arquitecturas de los sistemas distribuidos, centralizados o descentralizados. Por lo que puede disponer de un controlador central o varios controladores descentralizados, los dispositivos de interfaces, sensores y actuadores pueden también ser controladores y procesar la información.

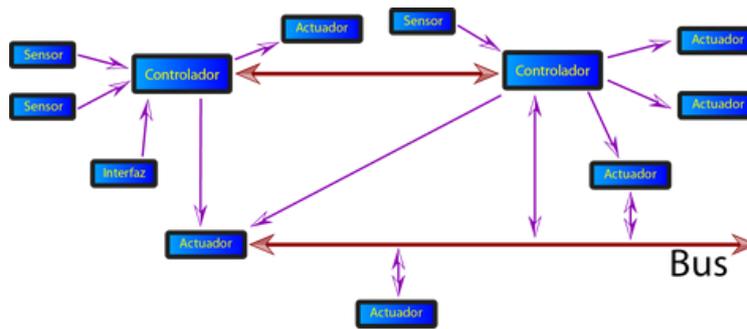


Figura 8: Arquitectura Híbrida / Mixta⁴

Los distintos dispositivos que va a contener el sistema se pueden clasificar en:

- **Controlador:** es el dispositivo central que gestiona el sistema según la programación y la información que recibe.
- **Actuador:** es un dispositivo capaz de ejecutar y recibir una orden del controlador y realizar una acción sobre un aparato o sistema.
- **Sensor:** es el dispositivo que monitoriza el entorno, tanto interior como exterior, captando información que transmite al controlador del sistema.
- **Bus:** es el medio de transmisión que transporta la información entre los distintos dispositivos. Este proyecto, se decide por un medio inalámbrico que se comenta en el siguiente apartado [2.1.2](#).

2.1.2 Tecnologías inalámbricas

Para interactuar con los dispositivos lo primero que se requiere es un protocolo de comunicación. Este proyecto considera únicamente protocolos sin hilos. Por ello, se estudian los protocolos de comunicación sin hilos ZigBee, WiFi y Z-Wave.

2.1.2.1 ZigBee

ZigBee⁵, está basado en el estándar IEEE 802.15.4. ZigBee o HomeRF Lite, se trata de un protocolo sin hilos con velocidades comprendidas entre 20 kB/s y 250 kB/s y cuya distancia de propagación, puede ir de los 10 a 75 m. Usa las bandas libres ISM de 2,4 GHz, 868 MHz (Europa) y 915 MHz (EEUU).

⁵ (domodesk, 2018) <http://www.domodesk.com/216-a-fondo-zigbee.html>

(zigbee, 2018) <http://www.zigbee.org/>

(wikipedia, 2018) <https://es.wikipedia.org/wiki/ZigBee>

(ieee, 2019) https://standards.ieee.org/standard/802_15_4-2015.html

ZigBee tiene capacidad para soportar hasta 255 nodos, los cuales suelen estar dormidos hasta que se les indica lo contrario, siempre con objeto de consumir menos energía, para que pueda ser alimentado con dos pilas AA durante meses, incluso años.

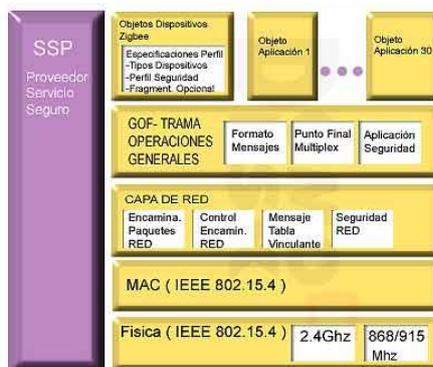


Figura 9: Modelo OSI ZigBee

En cuanto a la seguridad, ZigBee utiliza un modelo de seguridad de la subcapa MAC IEEE 802.15.4, que especifica 4 servicios de seguridad: control de accesos, datos encriptados, integración de tramas, secuencias de refresco.

2.1.2.2 WiFi

WiFi⁶, es otro protocolo de comunicación sin hilos, es el más conocido, el más implementado y del que más información existe. WiFi, es un protocolo desarrollado de la mano de la Alianza WI-FI, que garantiza la compatibilidad entre dispositivos y la cual los prueba y certifica.

WiFi, cumple con los estándares 802.11, dichas normas se diseñaron para sustituir las capas físicas y MAC de la norma 802.3 (Ethernet). Esto, implica que la única diferencia entre una red WiFi y una red Ethernet está en cómo se transmite la información, el resto es idéntico.

Existen cuatro tipos de WiFi:

⁶ (wikipedia, 2018) <https://es.wikipedia.org/wiki/WiFi>

- IEEE 802.11b, IEEE 802.11g e IEEE 802.11n, los cuales trabajan en la frecuencia de 2,4GHz y rinden a una velocidad de hasta 11Mbit/s, 54Mbit/s y 300Mbit/s, respectivamente y con un alcance entorno a los 100m.
- IEEE 802.11ac, es el más actual, trabaja a 5 GHz que, en la actualidad posee los canales menos saturados, por lo cual, se producen muy pocas interferencias. Dicho estándar trabaja a mayor velocidad entorno a los 867Mbit/s, pero en contrapartida y debido a su frecuencia mayor de trabajo, el alcance se reduce aproximadamente un 10 %.

Actualmente uno de los problemas habituales del WiFi es la saturación de su espectro y canales, debido al gran número de usuarios.

En cuanto a la seguridad, existen varias alternativas de cifrado, lo más común, es utilizar protocolos de cifrado de datos como el WEP, el WPA, o el WPA2 que se encargan de cifrar la información transmitida, protegiendo así su confidencialidad.

2.1.2.3 Z-Wave

El protocolo de comunicación de Z-Wave⁷ es un protocolo propietario perteneciente a Silicon lab⁸, del cual se ha liberado la gran parte de su código. Este protocolo, está diseñado para realizar comunicaciones fiables de baja latencia en paquetes de hasta 100 Kbits/s. La ventaja de Z-Wave es que trabaja a una frecuencia de 900 MHz, en lugar de a 2,4 GHz. Este hecho, proporciona un rendimiento superior por dos motivos: menos interferencias y mayor penetración de las ondas en paredes, pisos y muebles. Cada red Z-Wave puede incluir hasta 232 dispositivos, cada nodo, por lo general, es capaz de retransmitir un mensaje recibido para así garantizar la conectividad en lo que se conoce como una red mallada inteligente, esquivando así posibles obstáculos en el tránsito de la información. El rango de comunicación medio entre dos nodos alcanza los 30 metros (100 metros máximo), con la capacidad de repetir el mensaje hasta 4 veces o saltos entre nodos.

⁷ (wikipedia, 2017) <https://en.wikipedia.org/wiki/Z-Wave>
 (domodesk, 2018) <http://www.domodesk.com/162-a-fondo-z-wave-sin-cables.html>
 (z-wavealliance, 2018) <https://z-wavealliance.org/>

⁸ (silicon, 2018) <https://www.silabs.com/>

En cuanto a seguridad, Z-Wave usa cifrado S2 de extremo a extremo, lo que hace que los hogares inteligentes sean prácticamente invulnerables. S2 en Z-Wave, elimina por completo posibles vulnerabilidades de la red durante la inclusión del dispositivo. Los posibles ataques a una red Z-wave, se vuelven inútiles por el uso de código PIN y un intercambio de claves. Además, con las habituales tarjetas de radio, no puede detectarse este tipo de red, añadiendo esto, cierta capa de protección frente a la captura de tráfico de estas redes. Para conexiones con otros dispositivos IP, se encamina todo el tráfico Z / IP a través de un túnel seguro TLS 1.1.

Existe una mejora de este estándar que es el Z-Wave Plus. No obstante, Z-Wave Plus es totalmente compatible con los productos Z-Wave de anteriores generaciones, de estas compatibilidades y de la certificación de los dispositivos se encarga la Alliance Z-Wave.

Una tabla comparativa de Z-Wave, WiFi y ZigBee es la siguiente:



Figura 10: Figura comparativa Z-wave, ZigBee y WiFi

2.1.3 Plataformas de desarrollo

En este proyecto, se contemplan dos vías de desarrollo una para la red Z-wave y otra la para la red WiFi.

2.1.3.1 Plataformas de desarrollo Z-Wave

Para desarrollar dispositivos Z-Wave totalmente comercial, se tiene la posibilidad que ofrece su actual propietario Silicon lab⁹, la cual consiste en adquirir dos kits, uno para crear dispositivos (esclavos) y otro para diseñar controladores (maestros). Estos kits contienen todo lo necesario para poder desarrollar elementos que funcionen en una web Z-Wave, pero cabe recordar, que para lograr el sello Z-Wave, estos elementos una vez desarrollados, es la Alliance Z-Wave la cual certificará su funcionamiento y compatibilidad con el resto de elementos del mercado. Esta vía es costosa y escapa a las intenciones de este proyecto.

Por otro lado, la vía que se va a explotar en este proyecto consiste en la adquisición de ciertos elementos como RaZberry, USB Z-Stick y Z-Uno, los cuales se explicarán en el punto [2.2](#). Los dos primeros, son elementos que, conectados a otro, le da las capacidades de un controlador Z-Wave. El último, es un elemento basado en Arduino y el IDE de Arduino que permite desarrollar dispositivos Z-Wave y el cual tiene certificado por la Alliance Z-Wave 10 canales de comunicación, aunque como se verá, se pueden usar hasta tres más.

2.1.3.2 Plataformas de desarrollo WiFi

En cuanto a las plataformas de desarrollo para dispositivos doméstico WiFi existen varias posibilidades, a parte de la comercial:

- Basarse en plataformas Arduino con su IDE de desarrollo Arduino.
- Basarse en la plataforma Raspberry pi.
- Basarse en la plataforma Texas Instruments con sus placas MSP432P401R y CC3100 y su IDE Energia.

En este proyecto, para el desarrollo de dispositivos WiFi, solo se contemplan las dos últimas, Raspberry pi y Texas instruments, ya que se ha considerado que la plataforma Arduino está muy extendida y hay muchos proyectos similares, además el Z-Uno usa el mismo IDE de Arduino por lo que su integración queda asegurada.

⁹ (Silicon, 2018) <https://www.silabs.com/support/getting-started/mesh-networking/z-wave>

2.2 Estudio de mercado.

Este proyecto, se centra en la integración de diversas tecnologías abiertas, con el objetivo de tener mayor flexibilidad de diseño y por lo tanto poder abaratar costes. Este enfoque, es debido a que se piensa en un mercado doméstico, donde el mayor problema de aceptación de los sistemas domóticos, en este mercado, es la necesidad de obra y el coste. Lo primero, queda resuelto implementando sistemas sin hilos y lo segundo, se resuelve logrando sistemas que, gracias a su flexibilidad de diseño, permitan usar los elementos necesarios y del tipo necesario para abaratar el coste.

Así, se va a realizar en las siguientes líneas un estudio de diferentes controladores y dispositivos existentes en el mercado, para ver qué tipo de dispositivos se requeriría para suplir el sistema que se va a desarrollar en este proyecto y cuál es su coste.

2.2.1 Controladores

2.2.1.1 Definición de controlador domótico

El controlador domótico según vendomotica¹⁰ “tiene como finalidad captar toda la información que le brindan los diferentes sensores que haya distribuidos por distintos puntos de la casa. Toda la información recibida es procesada y, posteriormente, genera diferentes órdenes a los actuadores, que son los encargados de ejecutarlas”.

En la actualidad, hay diferentes opciones donde elegir, desde montar un centro domótico con una Raspberry Pi, hasta elegir un centro de control comercial, así tenemos:

2.2.1.2 Controladores en el mercado

Con la información obtenida en zwave.es¹¹ se presenta una comparativa de distintos controladores que están en el mercado:

- **Centro de control Z-Wave Vera Edge**, incorpora una interfaz sencilla, y es compatible con el protocolo Z-Wave y Z-Wave Plus. Es un controlador simple, pero en contra tiene

¹⁰ (vendomotica, 2015) <http://vendomotica.com/blog/definicion-de-controlador-domotico/>

¹¹ (zwave, 2018) <http://zwave.es/>

que no es compatible con algunos periféricos Z-Wave. Su mayor ventaja es su precio 139,95€.

- **Centro de control Z-Wave Zipabox de Zipato**, es un controlador basado en módulos: módulo de respaldo y USB, módulo para protocolo EnOcean, módulo para soporte KNX, módulo para Zigbee, módulo de expansión RF 433Mhz, módulo de control de consumo y módulo de alarma y seguridad. Estos módulos se instalan como conectados en los laterales del controlador, a través de las ranuras. Es uno de los controladores más fiables del mercado a un precio de 199€.
- **Centro de control Z-Wave eedomus plus**, es un controlador muy flexible, compatible con protocolos Z-Wave, EnOcean y la tecnología 433 Mhz, entre otros a través de periféricos de expansión. También es compatible con la mayoría de dispositivos no Z-Wave, con tecnología propietaria. Contiene un motor de creación de escenas domóticas, sencillo y potente, permitiendo crear escenarios complejos con reglas sencillas, evitando scripts. Otra función es la síntesis de voz, que permite interactuar a través de mensajes hablados. Por otra parte, permite diseñar una interfaz de control adaptada y personalizada a necesidades. Tiene un coste de 297,66€.
- **Centro de control Z-Wave Home Center 2 de Fibaro**, permite a través de programaciones, muchas cosas que, otros centros de control no contemplan. Su ventaja, está a la hora de gestionar infinidad de módulos domóticos. Soporta ciertos módulos Z-Wave Plus. Su punto fuerte es la capacidad de programación que tiene a través de scripts en LUA, que permite crear multitud de funcionalidades añadidas. Tiene un precio de 599€.
- **Jeedom**¹², Tal como se indica en domoticadomestica¹³, este controlador más que un controlador físico, es un proyecto domótico completo, se trata de un proyecto código abierto y libre, compatible Z-Wave, que permite que se construya cualquier solución domótica a medida, ofreciendo una interfaz gráfica para su explotación y configuración. Jeedom propone tres controladores domóticos diferentes, este proyecto se centra en Jeedom Mini. El cual, Se trata de una solución basada en Raspberry Pi o máquina virtual, compatible Z-Wave, es una solución muy económica, únicamente necesita un adaptador

¹² (jeedom, 2019) <https://www.jeedom.com/site/fr/>

¹³ (Maestre, 2015) <http://www.domoticadomestica.com/jeedom-controlador-domotico-z-wave-de-bajo-coste-basado-en-la-raspberry/>

RaZberry o un USB Z-Stick o similar conectado al elemento que tenga instalado el software.

Para favorecer el desarrollo de este sistema se ha creado un market¹⁴ de plugins que permitirá prácticamente integrar cualquier objeto. Este controlador tiene una versión gratuita, pero que una vez instalado en una máquina virtual, requiere de un z-Stick GEN5 tiene un coste de 46 €.

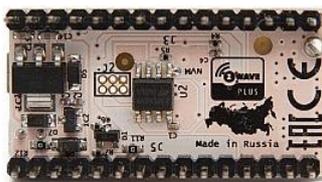
2.2.2 Dispositivos

Actualmente, hay una inmensa oferta de dispositivos en el mercado orientados a la domótica. En esta sección, se van a exponer aquellos con los que se pretende desarrollar el sistema domótico, junto con su contrapartida comercial.

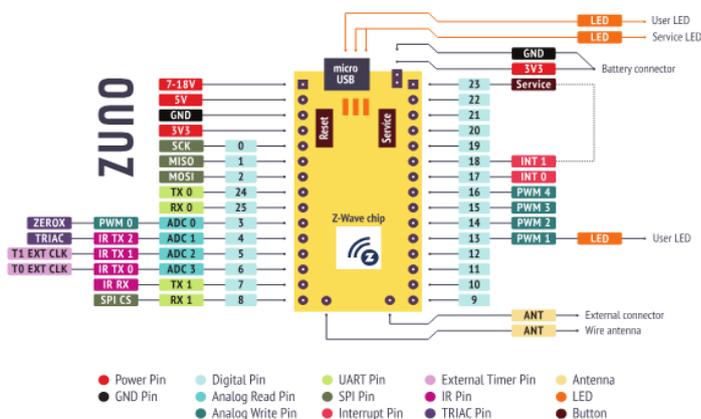
2.2.2.1 dispositivos para desarrollar

Los elementos con los que se pretende desarrollar dispositivos domóticos en este proyecto son:

2.2.2.1.1 Z-Uno



El Z-Uno¹⁵ que permite crear dispositivo Z-Wave sin tener un conocimiento profundo del protocolo o la programación de Z-Wave. Z-Uno es una combinación de la potencia del protocolo de radio Z-Wave y la simplicidad de Arduino. De hecho, Z-Uno



está inspirando en el proyecto Arduino, del cual hereda todos los conceptos y soluciones de hardware y software, manteniendo la máxima flexibilidad. La programación de Z-Uno mediante el lenguaje de programación C y se carga en Z-Uno utilizando IDE Arduino.

Coste 57 €

Figura 11: Z-Uno

¹⁴ (jeedom, 2017) <https://www.jeedom.com/market/index.php?v=d>

¹⁵ (z-wave.me, 2018) <https://z-uno.z-wave.me/>

2.2.2.1.2 Raspberry pi 3 b+

El Raspberry Pi 3 Modelo B +16 es un ordenador de placa simple de tamaño reducido. Es el último producto de la gama Raspberry Pi 3 y algunas de sus características son:

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) SoC de 64 bits a 1,4 GHz
- SDRAM LPDDR2 de 1 GB
- LAN inalámbrica IEEE 802.11.b / g / n / ac de 2.4GHz y 5GHz, Bluetooth 4.2, BLE
- Cabecera GPIO de 40 pines extendida
- Puerto micro SD para cargar su sistema operativo y almacenar datos
- Entrada de corriente continua de 5V / 2.5^a
- Coste 59,49



Figura 12: Raspberry pi

2.2.2.1.3 MSP432P401R+CC3100

El kit de desarrollo LaunchPad MSP432P401R¹⁷ permite desarrollar aplicaciones de alto rendimiento que se benefician de la operación de baja potencia. La MSP432P401R, incluye un ARM® Cortex®-M4F a 48MHz, potencia activa de 80uA / MHz y operación de RTC de 660nA, ADC de precisión SAR con rendimiento de 16 bits y acelerador AES256.



Figura 13: MSP432P401R

MSP-EXP432P401R dispone de una serie de pines (20 y 40) que facilitan la conexión de módulos y sensores que agregan funcionalidad adicional, como Bluetooth de baja energía, conectividad inalámbrica WiFi precisamente a estos módulos pertenece la tarjeta CC3100 que le añade conectividad WiFi. Algunas características son:

- MSP432P401R MCU de bajo consumo y alto rendimiento.
- ARM Cortex M4F de 48 MHz a 32 bits con unidad de punto flotante y aceleración DSP.
- Consumo de energía: 80uA / MHz activo y 660nA RTC en espera.
- Memoria: 256KB Flash, 64KB RAM.
- Temporizadores: 4 x16 bits y 2 x 32 bits.
- Comunicación: Hasta 4 I2C, 8 SPI, 4 UART.
- Conector de 40 pines, y soporte para módulos de 20 pines.

¹⁶ (raspberrypi.org, 2018) <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>

¹⁷ (Texas, 2018) <http://www.ti.com/tool/MSP-EXP432P401R>

- Emulador XDS-110ET incorporado con tecnología EnergyTrace +.
- Y conexión WiFi gracias al módulo CC3100.
- Coste 18,04€

2.2.2.2 dispositivos Comerciales

Los dispositivos comerciales se van a agrupar en dos grupos los WiFi y los Z-Wave. Toda la información está sacada de Amazon y z-wave.es.

Los WiFi son:

2.2.2.2.1 Shelly 2



Figura 14: Shelly 2

Controla una amplia gama de electrodomésticos y equipos de oficina (luces, líneas eléctricas, puertas de garajes, cortinas, sistemas de seguridad, radiadores, aires acondicionados, etc.) desde cualquier lugar con el interruptor de relé doble más pequeño, operado por WiFi, con función de persiana - Shelly S2W-1.0 Coste 19,90 €.

2.2.2.2.2 Shelly humedad y temperatura



Figura 15: Shelly humedad y temperatura

Shelly H&T tiene módulos incorporados para humedad y temperatura y es el primer sensor de humedad y temperatura WiFi del mundo que funciona a más de 1 año con la batería. Coste 24,90 €.

2.2.2.2.3 Neo 2,4 GHz Wi-Fi sensores Movimiento



Figura 16: Neo

El sensor de movimiento avanzado, PIR detecta personas o animales que se mueven, a una distancia de detección de 7 m, ya sea de día o de noche, se puede monitorear en tiempo real. Coste 17,99 €.

2.2.2.2.4 Shelly Sense



Figura 17: Shelly Sense

Shelly Sense es un sensor todo en uno que tiene incorporados módulos para detección de movimiento, temperatura, humedad, intensidad de luz y sonido.

Coste 99 €.

Los dispositivos Z-Wave son:

2.2.2.2.5 USB Z-Stick GEN 5 Aeotec



Figura 18: USB Z-Stick GEN5

Z-Stick Gen5 permite generar una puerta de enlace Z-Wave, es decir, una pasarela para el acceso a una red Z-Wave. Puede comunicarse con más de 230 dispositivos Z-Wave, Z-Stick Gen5 es una antena Z-Wave Plus que funciona con cualquier plataforma. Ha sido diseñado para hacer de interfaz Z-Wave en PC, portátiles, con Mac, Linux o Windows, con Raspberry Pi, e incluso con sistemas NAS seleccionados como Asustor de Asus. Coste 46 €.

2.2.2.2.5 MultiSensor 6 Aeotec



Figura 19: MultiSensor 6

Es un dispositivo Z-Wave que incorpora 6 funcionalidades en un solo dispositivo. Así, este dispositivo es compatible con multitud de plataformas, que contengan el sello de la alliance Z-Wave y dispone de sensor de temperatura, humedad, luminosidad, presencia rayos ultra violeta, y vibraciones anti manipulación. Coste 49 €.

2.2.2.2.6 RaZberry



Figura 20: RaZberry

RaZberry es una tarjeta que se conecta directamente a las GPIO de una Raspberry Pi alimentado directamente a través de éstas. El sistema se basa en un chip Z-Wave de Sigma Designs 5202 y una antena directamente integrada PCB. RaZberry, Permite a la Raspberry Pi convertirse en un centro de control de automatización del hogar compatible Z-Wave. RaZberry se puede utilizar también con otras soluciones como Domotiz, Jeedom, etc. Coste 58,95 €.

2.2.2.2.7 Doble Relé ON / OFF ZIPATO Z-Wave Plus



Figura 21: Doble Relé Zipato Plus

Es un interruptor dual que permite controlar dos luces o dos dispositivos de forma remota utilizando el protocolo Z -Wave Plus, manteniendo los interruptores existentes. Con su tecnología de calibración, se puede utilizar para activar y desactivar muchos tipos de lámparas , incandescente, fluorescente y LED . Además, es capaz

de transmitir el consumo instantáneo de energía (W) y acumulativa (kWh) unido a él . También cuenta con una protección contra la sobrecarga . Coste 59,99 €.

2.2.2.2.8 Detector de Moho, Temperatura y Humedad - POPP



El detector de moho POPP, se puede colocar en cualquier ubicación o estancia , que deba ser supervisada. El detector mide la temperatura y la humedad de su ambiente. Coste 34,99 €.

Figura 22: Detector de Moho POPP

2.2.2.2.9 Detector de Movimiento - Z-Wave Plus - Hank



Figura 23: Detector movimiento Hank

El detector de movimiento de Hank, con un diseño compacto reacciona a lo que ocurre en su casa en tiempo real. Funciona con baterías y además también se puede alimentar por USB. Se puede colocar en una esquina, en las paredes o en el techo gracias a su soporte imantado. Combinado con otros dispositivos de la familia Z-Wave, puede gestionar las luces automáticamente para crear la iluminación adecuada y deseada en su hogar. También puede formar parte de su sistema de seguridad, enviándole una alerta si un intruso se mueve en su hogar. Coste 34,99 €.

2.2.2.2.10 Sensor de luz con pantalla LCD Everspring



Figura 24 Sensor de luz Everspring

El ST815 es un sensor de luz que está diseñado para ser instalado en una mesa o en la pared. Dispone de una pantalla LCD para la luminosidad. 4 botones para cambiar entre valores mínimo, máximo o instantáneo.

El sensor supervisa el entorno de nivel de luz ambiental. La medición de la luminosidad se envía en intervalos regulares a un controlador Z-Wave compatible. Coste 58,90 €.

Capítulo 3. Descripción funcional

Se va a desarrollar un sistema domótico inalámbrico, a través del cual, se automatizará, con hardware comercial y de desarrollado propio, persianas, punto de luz, calefacción eléctrica, mide temperaturas, humedad, presencia, luminosidad y toma decisiones. Esto se realizará, a través de la implementación de dos redes sin hilos, una Z-Wave, otra WiFi, que serán dominadas por un controlador domótico. Dicho controlador, será el que almacene los datos, tome las decisiones y dé las instrucciones, para el control de persianas, luces y demás decisiones. El elemento que recibirá las órdenes y hará de actuador/sensor se ha de conectar a los distintos interruptores, a través de la electrónica necesaria. Dichos elementos, son los que, en instancia final, enviarán los datos de los sensores al controlador y recibirán la orden del éste para, subir y bajar las persianas, apagar y encender luces, etc.

3.1 Diagrama de bloques del sistema

El sistema que se pretende desarrollar responde al siguiente esquema:

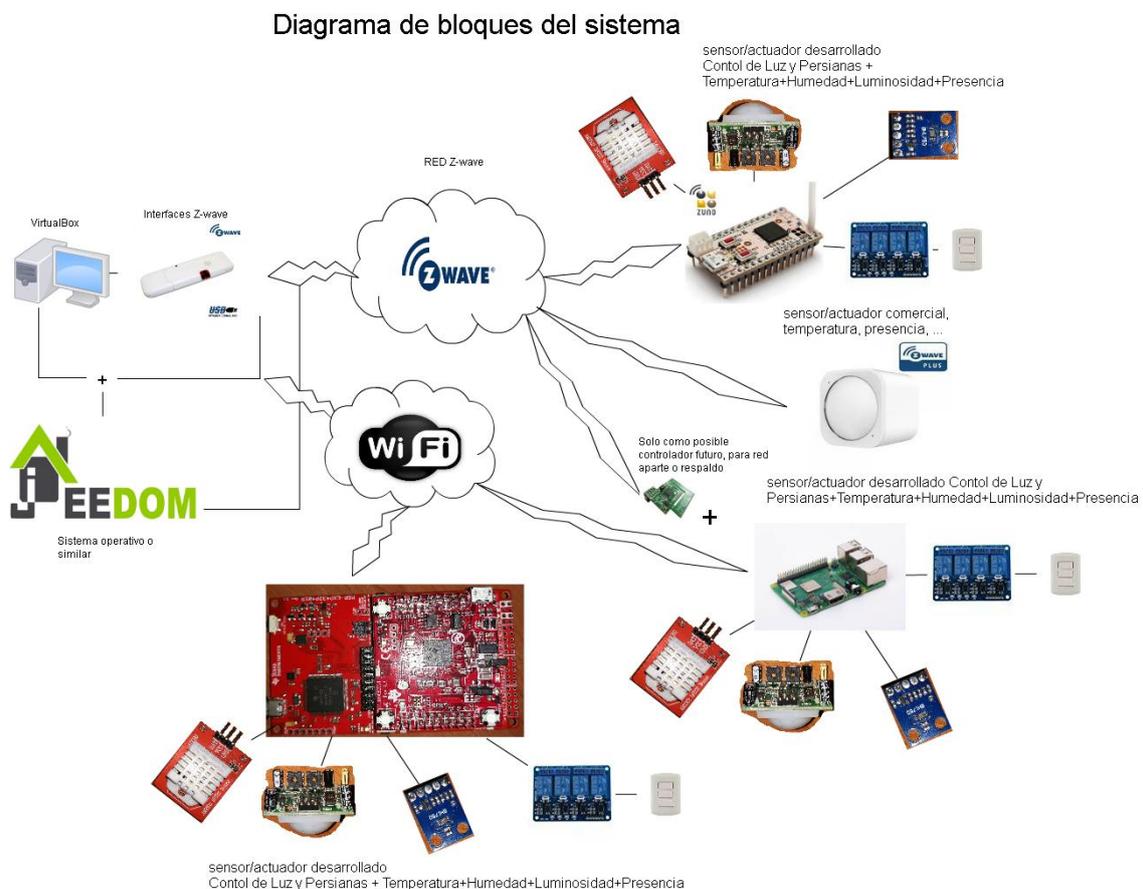


Figura 25: Diagrama de bloques general del sistema

3.2 Propuesta de arquitectura

En este proyecto, de las posibles arquitecturas que se han visto en el punto [2.1.1](#), centralizada, descentralizada, distribuida y mixta, se decide implementar una arquitectura centralizada.

Esta arquitectura, se ha considerado la forma más fácil, de usar diferentes tipos de protocolos de comunicación y diferentes tecnologías. Así, se tiene un nodo central, con la capacidad de comunicarse con todas las tecnologías existentes en el sistema, éste hace de puerta de enlace de las diferentes redes uniéndolas. Además, se facilita la creación de reglas para la toma de decisiones, ya que el sistema está organizado de tal forma que el controlador es el que recibe la información de los sensores, la analiza y envía las órdenes a los actuadores.

3.3 Propuesta de protocolo inalámbrico

Entre las opciones comentadas, Z-Wave, WiFi y ZigBee en el punto [2.1.2](#) se propone inicialmente usar el protocolo Z-Wave.

En principio, Z-Wave tiene peores características que los demás, ya que es más lento y permite menos dispositivos. Pero, a pesar de todo esto, se considera que tiene grandes ventajas, como se ha comentado en el apartado citado, y aunque la velocidad y número de dispositivos que soporta Z-Wave es menor, es suficiente para cualquier aplicación domótica, incluso si esta fuera compleja, cámara, multimedia... Para soportar este protocolo, será necesario que la electrónica use extensiones tipo Z-Stick USB o RaZberry para tener la interfaz Z-Wave.

Finalmente, y debido a los problemas y contratiempos comentados en los puntos [1.5](#), [5](#) y [7](#) también se decide usar WiFi. Esta decisión, es basada en el principalmente pensando en el cumplimiento de los plazos del proyecto, ya que es un protocolo muy conocido e implementado, del cual se puede adquirir electrónica fácilmente y rápidamente. Sin embargo, no cabe olvidar que WiFi es un protocolo altamente activo, ya que sus capacidades en cuanto velocidad y alcance son superiores al resto. Simplemente, el problema es que WiFi no está diseñado para sistemas domóticos, en donde se premia el consumo energético, frente al rendimiento, como si lo están ZigBee o Z-Wave. Por otro lado, su espectro radioeléctrico está

más saturado, pero a pesar de ello tendrá un rendimiento en cuanto a transmisión de información mucho mayor, que los protocolos comentados.

3.4 Propuesta de controladora

Tal como se ha venido diciendo, se busca que el modelo sea de bajo coste altamente flexible y sobre todo escalable. Por ello, la opción elegida como controlador (master) y que se propone en este proyecto es Jeedom, instalado en una en una máquina virtual con un USB z-Stick GEN5, ya que es de código abierto y compatible con múltiples protocolos entre ellos Z-Wave. Jeedom, permitirá diseñar e integrar cualquier solución, protocolo o tecnología domótica, pudiendo diseñar una solución a medida y altamente personaliza. También, cuenta con un github¹⁸ y una documentación¹⁹ con la información necesaria para su instalación (ver [anexo I](#)) y uso. En definitiva, Jeedom cuenta con las funciones de cualquier controlador domótico, y se le puede añadir e instalar infinidad de plugins, de hecho, para demostrar su potencial y dar otra opción de diseño, parte de las soluciones basadas en la Raspberry y MSP432P401R usarán algunos de estos plugins.. Otra cosa a destacar, es que no está basado en la nube, como otros controladores, cosa que se considera positiva para proteger la privacidad.

Respecto a la interfaz gráfica que ofrece, se pretende que este software sea accesible, tanto internamente, como externamente y que por lo menos externamente, use un acceso seguro vía protocolo de transferencia de hipertexto seguro (HTTPS), para lo que habrá que generar certificados y crear un dominio, que resuelva el problema que causa la IP dinámica que suelen distribuir las operadoras a los hogares.

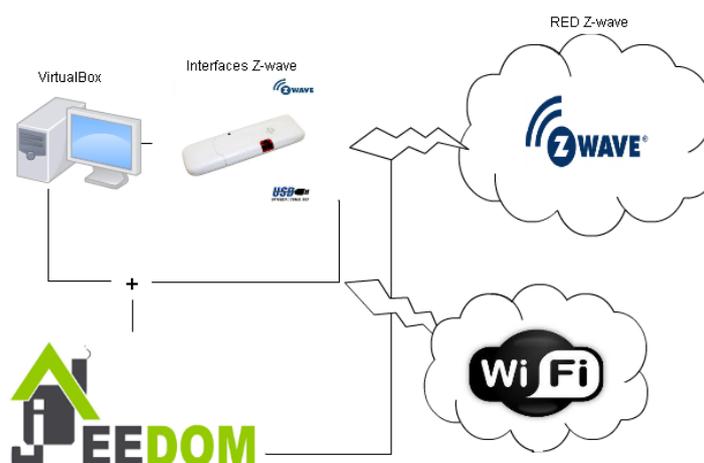


Figura 26: Diagrama controlador domótico

¹⁸ (Jeedom, 2018) <https://github.com/jeedom>

¹⁹ (jeedom, 2019) <https://jeedom.github.io/documentation/>

3.5 Propuesta de dispositivo y desarrollos

Uno de los retos de este proyecto es integrar distintas tecnologías, tanto hardware, como software como de protocolos. Por ello, se decide realizar distintos dispositivos, con funcionalidades parecidas, pero desde un enfoque distinto, para demostrar el amplio abanico de posibilidades que ofrece la filosofía de código abierto y libre, añadiendo al sistema una alta flexibilidad, escalabilidad e incluso un ahorro considerable de costes. Así, a continuación, se expondrán los dispositivos que se van a desarrollar.

3.5.1 Dispositivo comercial

El esquema de funcional de este dispositivo corresponde al siguiente:

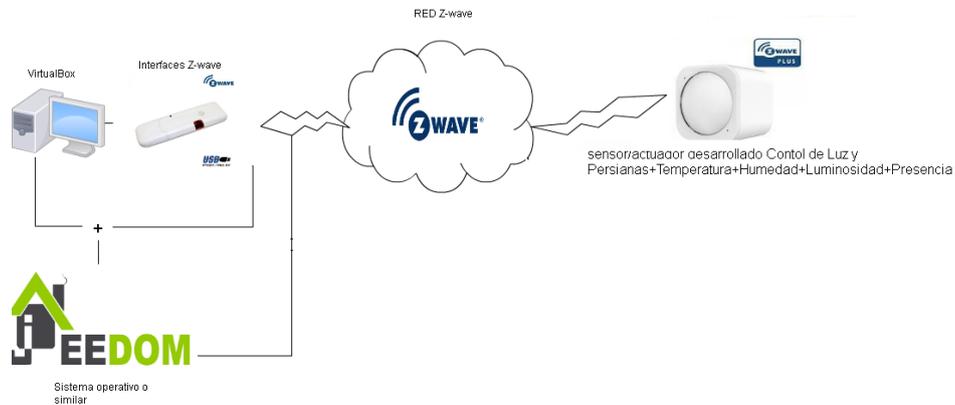


Figura 27: Diagrama funcional sensor comercial MultiSensor 6

Se trata de un sensor comercial MultiSensor 6 de Aeotec modelo ZW100-C, el cual, se va a instalar en el pasillo de la casa y va a controlar la temperatura, humedad, luminosidad presencia, rayos ultravioleta y vibraciones anti manipulación.

3.5.2 Dispositivo Basado en Z-Uno

El esquema funcional de este dispositivo corresponde al siguiente:

Diagrama de bloques del sistema



Figura 28: Diagrama funcional dispositivo desarrollado Z-Uno

Se trata de un dispositivo desarrollado basado en Z-Uno y red Z-Wave, el cual, se va a instalar en el salón y va a controlar los tres motores de las tres persianas de este, la calefacción eléctrica del salón, la luz general del salón, la luz de la zona de mesa y la luz del pasillo. Además, gracias a la electrónica conectada al Z-Uno se controla presencia, temperatura, humedad y luminosidad del salón.

3.5.3 Dispositivo basado en Raspberry pi

El esquema funcional de este dispositivo corresponde al siguiente:

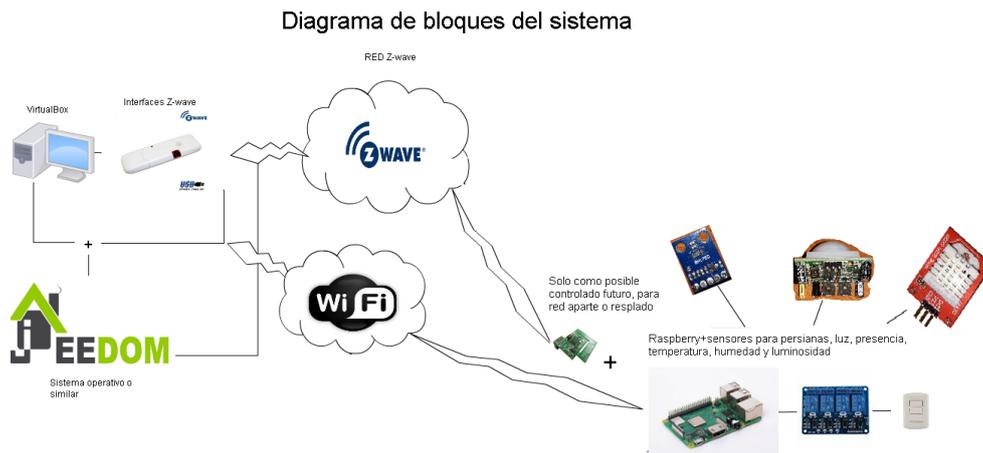


Figura 29: Diagrama funcional dispositivo desarrollado Raspberry

Se trata de un dispositivo desarrollado, basado en Raspberry pi y red WiFi, el cual, se va a instalar en la habitación 1 y va a controlar el motor de la persiana, la calefacción eléctrica y el

punto de luz de ésta. Además, gracias a la electrónica conectada dicha Raspberry se controla presencia, temperatura, humedad y luminosidad de la instancia.

Aclarar, que en el esquema se aprecia como la RaZberry está conectada a la Raspberry, esto pertenece al planteamiento inicial, el cual se modifica como se ha explicado en los puntos [1.5](#), [5](#) y [7](#). Lo cual, no quiere decir que la Raspberry pueda actuar como esclavo en la red Z-Wave, simplemente responde a que se ha instalado la RaZberry en la Raspberry pi, quedando está operativa para hacer de controlador de una nueva red Z-Wave o como controladora de respaldo algo que quedará contemplado en las mejoras futuras, punto [7.3](#).

3.5.6 Dispositivo basado en MSP432 y CC3100

El esquema de funcional de este dispositivo corresponde al siguiente:

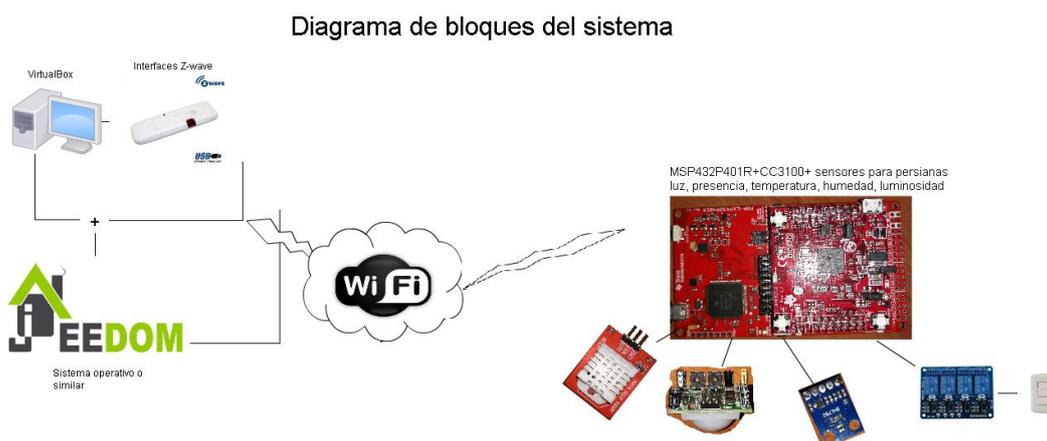


Figura 30: Diagrama funcional dispositivo desarrollado MSP432+CC3100

Se trata de un dispositivo desarrollado, basado en MSP432P401R+CC3100 y red WiFi, el cual, se va a instalar en la habitación 2 y va a controlar el motor de la persiana, la calefacción eléctrica y el punto de luz de ésta. Además, gracias a la electrónica conectada dicho conjunto, se controla presencia, temperatura, humedad y luminosidad de la habitación 2.

3.6 Sistema de tomas de decisiones (Escenas)

El sistema de toma de decisiones pretende a través del controlador Jeedom añadir inteligencia al sistema, permitiendo tomar una serie de decisiones según los valores que los distintos sensores devuelven al controlador. Así, estas reglas son:

- Cuando amanezca todas las persianas de toda la casa han de subir.
- Cuando atardece todas las persianas de la casa han de bajarse.
- Si se detecta presencia en el pasillo y hay poca luminosidad, se enciende la luz del pasillo, durante 10 segundos.
- Si la temperatura individual del salón baja de una cierta cifra, la calefacción del salón se enciende, pero cuando la temperatura del salón sube de otra cierta cifra, la calefacción se apaga.
- Si la temperatura individual de la habitación 1 baja de una cierta cifra, la calefacción de la habitación 1 se enciende, pero cuando la temperatura de la habitación 1 sube de otra cierta cifra, la calefacción se apaga.
- Si la temperatura individual de la habitación 2 baja de una cierta cifra, la calefacción de la habitación 2 se enciende, pero cuando la temperatura de la habitación 2 sube de otra cierta cifra, la calefacción se apaga.
- Si la temperatura media de la casa, entendida como media aritmética de todos los sensores de temperatura de la casa, baja de una cierta cifra, la calefacción de todas las instancias disponibles se enciende, pero cuando la temperatura media de la casa sube de otra cierta cifra, la calefacción de todas las instancias disponibles se apaga.
- Si se activa la alarma y se detecta algún tipo de presencia en cualquiera de los sensores de presencia del sistema, se manda un mensaje vía telegram.

Todas estas reglas, escenas tal como las llama Jeedom, pueden ser activadas o desactivadas por el usuario, vía un botón virtual en Jeedom.

Capítulo 4

Ahora, se expondrá con más detalle todo lo expuesto en el capítulo anterior, es decir, lo realizado y desarrollado en este proyecto.

4.1 Propuesta de controladora y redes Z-Wave WiFi

Para la controladora, se ha creado mediante el programa VirtualBox, una máquina virtual con al menos 1024 Mbyte de RAM, un disco duro de 20 GByte e IP estática.

En dicha máquina virtual se instala la distribución Debian 9. Seguidamente se instala Jeedom siguiendo los pasos de su documentación para la instalación²⁰ en una máquina virtual (ver [anexo I](#)).

Una vez instalado Jeedom, hay que registrarse en su market, e instalar el plugin Z-Wave y asegurar que se conecta el USB Z-Stick GEN5 y que éste es detectado por la máquina virtual. También, es importante asegurar que dicha máquina virtual, tiene conectividad con la red WiFi que proporciona el enrutador del distribuidor de internet pertinente. Una vez logrado esto, se tiene el controlador Jeedom con una red z-wave y otra WiFi, desplegadas y listas para incluir dispositivos.

Una vez listo el controlador, para su acceso desde una red externa (internet) vía HTTPS, primero se ha de resolver que la IP pública, que el proveedor de internet ofrece suele ser dinámica, es decir, va cambiando y después hay que generar un certificado para poder usar en la conexión HTTPS. Para resolver el primer punto, primeramente, hay que indicar en el router correspondiente, que redirija el puerto del protocolo de control de transmisión (TCP) para https (443) a la IP del controlador y hacer los cambios pertinentes en el fichero hosts del Debian. Seguidamente, se crea un dominio en un servidor de resolución de nombres (DNS) público y gratuito como es noip²¹. Tras crear el dominio, se debe generar un certificado válido para este dominio y firmado por una autoridad certificadora (CA) reconocida, para ello se usa Let's Encrypt una entidad de certificación gratuita (ver [anexo II](#)²²). Una vez, finalizado este proceso ya se puede acceder a Jeedom desde el exterior para su explotación y configuración.

Para usos, que se explicarán en apartados posteriores, también se van a instalar los siguientes plugins:

- Virtual: Permite crear dispositivos virtuales.
- Jeedouino: Permite controlar las GPIO de Raspberry pi, Arduino, ...
- Jeelink: Permite controlar un dispositivo de otra instalación de Jeedom, como si estuviera integrado en el Jeedom principal.
- Script: permite crear scripts, consultas HTTP, HTML, ...
- Weather: da acceso a los datos de una web con información meteorológica.

²⁰ (jeedom, 2017) https://jeedom.github.io/documentation/installation/fr_FR/index#tocAnchor-1-9-4

²¹ (noip, 2018) <https://www.noip.com/>

²² (Brunet, 2018) <https://www.domo-blog.fr/securiser-jeedom-https-certificat-ssl-dote-nouveau-challenge-http-01/>

- Telegram: Integra la red social Telegram en Jeedom para poder utilizarla.

También, es recomendable en Jeedom, crear lo que éste llama objetos, ya que estos se asignan a cada dispositivo como padres y facilita el entendimiento y configuración de reglas y parte de la configuración. Así se han creado los objetos Casa, Plata 1, Pasillo, Habitación 1 Habitación 2 y Salón.

4.2 Dispositivo comercial

El sensor comercial es el MultiSensor 6 de Aeotec modelo ZW100-C, el cual, se instala en el pasillo de la casa. Éste, se integra en Jeedom mediante el plugin Z-Wave, que permite su inclusión en la red. Así, Jeedom lo reconoce y le da un identificador (id) de nodo para comunicarse con él. Una vez está incluido en la red, Jeedom es capaz de leer los comandos que este sensor envía, informando de la temperatura (°C), humedad (%), luminosidad (lux), presencia, rayos ultravioleta y vibraciones por manipulación, para los que está preparado.

Todos estos datos, son mostrados en Jeedom prácticamente en tiempo real, si bien, es cierto que tiene una cierta latencia (de milisegundos a segundos) que es configurable variando ciertos parámetros. A la hora de tocar estos parámetros, hay que tener cuidado, ya que este dispositivo, cuando no manda datos está durmiendo, logrando así un considerable ahorro energético, pudiendo funcionar largos períodos con solo una batería. Así, a la hora de configurar el parámetro de tiempo entre actualizaciones de información, hay que llegar a un compromiso entre el ahorro energético y las prestaciones deseadas. En este proyecto se pide que actualice cada 500 milisegundos, ya que se va a hacer uso del detector de presencia para encender la luz del pasillo y la alarma.

- La persiana sube/baja completamente.
- La persiana está subiendo o bajando y sin finalizar el movimiento se le pide que suba o baje una cantidad mayor o menor. El código calcula el nuevo tiempo y le descuenta el tiempo que ha estado subiendo o bajando, y continúa con el relé activo este nuevo tiempo, dejando la persiana correctamente abierta en la última posición dada.
- La persiana sube o baja y se le pide cambiar completamente el sentido. El código para la persiana calcula, considerando la posición actual, el nuevo tiempo de subida o bajada y activa el relé correspondiente dicho tiempo.

Así, el sistema tiene un control total de la persiana, con control de posición, sin anular los interruptores existentes.

También, como se puede apreciar a dicho Z-Uno se le ha añadido una serie de sensores para lo siguiente:



Figura 32: DHT22

- **Control de temperatura y humedad:** DHT22, Sensor digital que utiliza un sensor capacitivo de humedad y un termistor. Sus datos son enviados mediante una señal digital. Este sensor, sólo se puede obtener nuevos datos una vez cada 2 segundos. Y para facilitar su integración, se usa la librería ZUNO_DHT.h.



Figura 33: BH1750

- **Control de luminosidad:** BH1750 Sensor digital de que incorpora un ADC de 16bits que proporciona una resolución de 65535 niveles. tiene una baja influencia al espectro infrarrojo, rechazo al ruido de 50/60 Hz y alta independencia del origen de la fuente de luz. Requiere del bus Circuito inter-integrado (I2C), para su uso. Y para facilitar su integración, se usa la librería ZUNO_BH1750.h

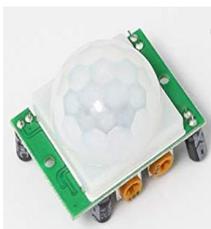


Figura 34: HC-SR501 PIR

- **Control de presencia:** Neuftech HC-SR501 PIR utiliza un sensor infrarrojo pasivo LHI778 y el BISS0001 IC para controlar la detección del movimiento, se le puede ajustar la sensibilidad en un rango de detección de movimiento de 3 metros a 7 metros y también el tiempo de disparo a nivel alto cuando detecta movimiento.

Destacar, que para el control de tiempo, tanto en lo que a las persianas se refiere, como en la actualización de los sensores, no se usan órdenes que pausan la ejecución, perdiendo eficiencia, como haría una orden `delay()`, si no que se ha usado la información que devuelve la orden `millis()` (tiempo de reloj en milisegundos) que añade eficiencia al código.

También, se ha usado uno de los plugins instalados en Jeedom, concretamente el Virtual, el cual, permite crear dispositivos virtuales y con cierta personalización. En este caso, se ha usado este plugin, para crear un dispositivo por cada elemento, es decir, para que el entorno gráfico muestre cada elemento por separado, un icono por cada elemento, tres persianas, tres puntos de luz, calefacción, humedad, temperatura, luminosidad y presencia separadas, en vez de un bloque unido, con todos estos elementos, que mostraría Jeedom por defecto:



Figura 35: Comparación sin usar Virtual (izquierda) usando Virtual (derecha)

También, hay que tener en cuenta para los diseños de Z-Uno que este tiene 10 canales Z-Wave certificados, por la Alliance Z-Wave, pero según el fabricante se pueden usar 3 más, sin estos tres últimos, estar certificados. Como se puede apreciarse en el desarrollo, se han llegado a usar once canales (tres luces, tres persianas, calefacción, temperatura, humedad, luminosidad y presencia) sin ningún problema.

4.4 Dispositivo basado en Raspberry pi

El dispositivo desarrollado con la Raspberry pi responde al siguiente esquema:

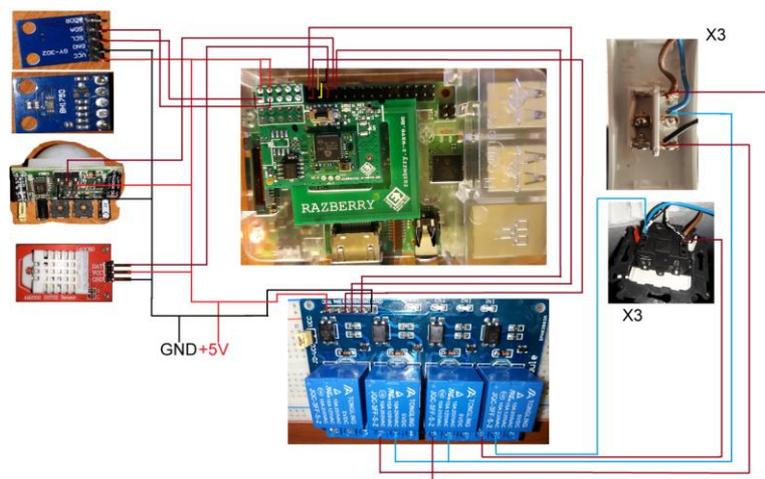


Figura 36: Esquema eléctrico dispositivo desarrollado Raspberry pi (X3 indica que hay dos más)

Este dispositivo basado en Raspberry pi está situado en la habitación 1. Primero destacar que en dicha Raspberry pi es necesario instalar Jeedom según la documentación²⁴ de éste para su instalación en una Raspberry. Una vez instalado Jeedom también hay que instalar los plugin: Virtuel, Jeedouino, Jeelink, Script.

Como se ha comentado en apartados anteriores, se decide implementar las funcionalidades comentadas (persiana, luz, calefacción, humedad, temperatura, presencia y luminosidad) desde un enfoque distinto, usando los plugins que Jeedom facilita. Así, se procede a exponer como se implementa cada parte:

- **Control de luz:** se va a usar el plugin Jeedouino y Virtuel, que facilita el control de los pines GPIO de la Raspberry, incluso remotamente. Por ello, en este caso se van a controlar los pines de la Raspberry, desde la controladora (máquina virtual), para ello es requisito que Jeedouino esté instalado en ambos, controladora y Raspberry. Una vez, se tiene este plugin correctamente configurado, se crea una conexión IP vía WiFi, que permite controlar los pines de la Raspberry y ya simplemente desde la controladora, se elige el pin de la Raspberry al que se ha conectado el relé y se generan las órdenes encender, apagar y estado. Ahora de la mano del plugin Virtuel se crea un dispositivo virtual para la luz, al cual, se le asocia las órdenes comentadas. Tras tener todo bien asociado y configurado, se dispone de un icono en el controlador que maneja a la perfección la luz de la habitación 1.

²⁴ (jeedom, 2017) https://jeedom.github.io/documentation/installation/fr_FR/index#tocAnchor-1-8

- **Control de calefacción:** los pasos son los mismos que para el control de luz, la única diferencia es que esta vez, se usarán los plugins desde la propia Raspberry pi y no desde el controlador principal. Una vez hecho esto, se tiene un icono que controla a la perfección la calefacción, pero este estará en el cuadro de mandos de la Raspberry, no del controlador (máquina virtual). Por ello, para que este icono sea manejado desde el controlador, se usa el plugin Jeelink que debe estar instalado en ambos, Raspberry y máquina virtual. Jeelink, hace una asociación entre el Jeedom de la controladora y el de la Raspberry, permitiendo controlar el icono del control de calefacción situado en la Raspberry, tal como si hubiera sido creado en el controlador principal.
- **Control de persianas:** en este caso, se va a generar exactamente igual que el control de calefacción, haciendo uso de Jeedouino, Virtuel y Jeelink. Pero, esta vez las órdenes creadas en Jeedouino son: subida, parar subida, bajada, parar bajada, la subida y bajada cada una controla un relé distinto. En cuanto, a Virtuel, solo se van a asociar las órdenes de subida, bajada y parada, esta última, no es más que la asociación de las órdenes parar subida y parar bajada. Pero, en este caso también hay que añadir una configuración, para evitar que tanto, la subida, como la bajada, puedan estar activadas a la vez. Así, se configura: que antes de ejecutar la orden subir debe de parar la bajada y antes de ejecutar la orden bajar debe parar la subida. Hasta aquí, se tendría la capacidad de subir la persiana indefinidamente, bajar la persiana indefinidamente y pararla en cualquier momento, algo no aconsejable, ya que, aunque el motor de la persiana suele tener la capacidad de parar cuando llega a su fin de carrera por el solo, supondría tener un relé activo todo el tiempo (si no se le da a parada) debiendo anular los interruptores físicos, para proteger el motor de alimentaciones inversas debido a tener el relé subida activo y pulsar el interruptor bajada. Para solucionar esto, se crea una regla en el Jeedom de la Raspberry, dicha regla se implementa a través de lo que Jeedom llama escenas y esta escena consiste, por un lado, en el disparador, que sería la acción que ejecuta la regla y, por otro, en la regla en sí. En este caso, como disparadores se tiene subida, bajada, parar subida y parar bajada. Y la regla no sería nada más, que dos condiciones, que si se cumplen la persiana ha de estar un cierto tiempo realizando la acción o realizarla hasta que se dé la correspondiente orden parar (diagrama flujo n [anexo III](#)). Decir, que no se implementa control de estado (posición de la persiana actual), si se puede saber si el último movimiento es de subida o bajada. Esto, da una solución menos completa que la del Z-Uno, pero funcional. Esta decisión, responde a la necesidad de seguir implementando más sensores, buscando implementaciones simples pero funcionales. La solución

al control de estado de las persianas quedará propuesta como mejora futura, en el punto [7.3](#).

- **Control de temperatura y humedad:** aquí se usarán los plugin Jeedouino, Virtuel y Jeelink. En este caso, Jeedouino implementa este sensor y solo hay que indicar en qué pin se conecta. Seguidamente, con Virtuel se crea un icono para la humedad y otro para la temperatura. Por último, con Jeelink se asocia al Jeedom del controlador, para poder ver dichos valores en él.
- **Control de luminosidad:** para el control de luminosidad se hará uso de los plugin Script, Virtuel y Jeelink. En este caso, para el control de luminosidad se implantará un código en lenguaje C, en el cual, se hará uso de la librería BH1750. Una vez, este código es funcional, se compilará generando un fichero, el cual, se asociará al Jeedom de la Raspberry, gracias al plugin Script y su opción de script. Una vez hecho esto, Jeedom tiene la capacidad de ejecutar el código C y recuperar el resultado. Para que la ejecución del script se haga cíclicamente, se le activa la opción cron. Por último, con Virtuel se genera un icono con dicha información y con Jeelink, se asocia al Jeedom del controlador para su visualización.
- **Control de presencia:** para el control de presencia se hará uso de los plugin Script, Virtuel y Jeelink. En este caso, se va a proceder exactamente igual que con el control de luminosidad, la única diferencia es que, para demostrar la versatilidad del plugin, esta vez se va a realizar usando la opción de python y, por lo tanto, el código estará escrito en dicho lenguaje. Solo añadir, que en este caso se ajusta el tiempo de activación en alto nivel del sensor Neuftech HC-SR501 PIR, para que cuando detecte movimiento, el nivel alto dure lo suficiente, como para que se ejecute el código. Así, una vez realizado todo, se obtendrá un control de presencia de la habitación 1 visualizado desde el controlador.

Indicar, que uno de los objetivos era instalar la RaZberry en la Raspberry y a pesar de no poder actuar este conjunto como esclavo, se ha instalado y comprobado, su funcionalidad para usarse como controlador de una nueva red Z-Wave o posible respaldo del controlador principal, esto queda propuesto como mejora en el punto [7.3](#).

4.5 Dispositivo basado en MSP432P401R+CC3100

El dispositivo desarrollado con las placas MSP432P401R+CC3100 responde al siguiente esquema:

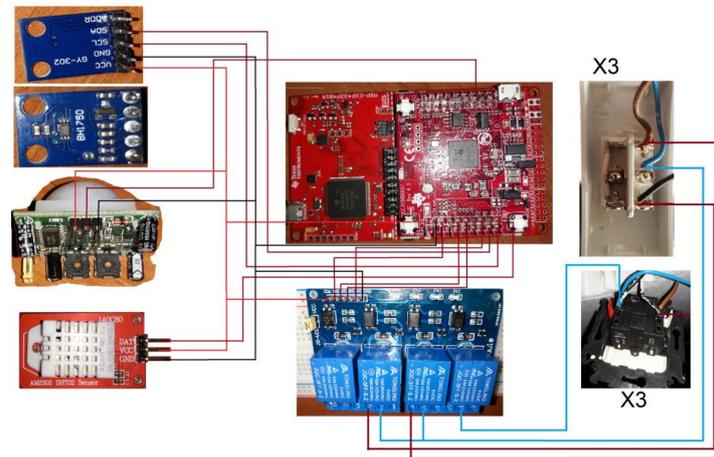


Figura 37: Esquema eléctrico dispositivo desarrollado MSP432P401R+CC3100 (X3 indica que hay dos más)

Este dispositivo basado en el conjunto MSP432P401R+CC3100, está situado en la habitación 2.

La implementación de este dispositivo se realiza de forma similar a la del Z-Uno, pero usando una red WiFi, ayudado de las referencias²⁵ y el IDE energía. Para resolver este desarrollo, se crea un código que ofrezca un servidor web, en el puerto TCP 8080, al que se accede vía WiFi y al cual, Jeedom le pueda hacer consultas http o HyperText Markup Language (html). Así, se toma como punto de partida, el ejemplo de servidor web accesible mediante WiFi que energía ofrece "SimpleWebServerWiFi". Este código, controla el acceso a servidor mediante WiFi, pero hay que modificarlo mediante intrusiones en lenguaje c para que realice las acciones necesarias e incorpore el código html, que genera la web con un cuadro de mando de las funcionalidades a implementar (control de luz, control de calefacción, control de persianas, control de temperatura y humedad, control de luminosidad y control de presencia).

- **Control de luz:** para este, se genera un código que implementa, con html, dos botones para encender y apagar la luz, en el cuadro de mando de la web. Estos botones, son accesibles vía consulta HTTP. Una vez, generado este código, en el

²⁵ (energía, 2017) <https://energia.nu/reference/>

lado del controlador y mediante el plugin script, se usa la opción http de éste, añadiendo las peticiones http necesarias, por ejemplo `http://192.168.1.138:8080/LUZ=ON` y `http://192.168.1.138:8080/LUZ=OFF`. Ya solo queda mediante el plugin Virtuel, crear un icono, al cual, se le asocia dichas peticiones http generadas con script y con el cual, se controla el estado actual de la luz.

- **Control de calefacción:** esta funcionalidad, se implementa de forma idéntica al control de luz, se genera el código html para que muestre dos botones en la web, a los cuales, se le puede hacer consultas http desde Jeedom utilizando Script, creando un icono mediante Virtuel, para poder interactuar con la calefacción.
- **Control de persianas:** Este punto es parecido a los anteriores, se genera un código html para que en la web aparezcan tres botones, subida, bajada y parada. En el resto de código, escrito en C, se controlará el tiempo de subida y bajada y que ambos estados no puedan estar activados a la vez. Por otro lado, desde Jeedom mediante Script y http se generan las tres peticiones para subir, bajar y parar la persiana, a las cuales, se asocia, mediante Virtuel, un icono en el controlador. También, se usa la configuración de Virtuel, para controlar si el último movimiento es de subida o bajada. Funcionalmente, la implementación es exactamente igual que la de la Raspberry, se puede subir, bajar, parar cuando se desee, pero no controla el estado, es decir, la posición actual de la persiana, esto es debido al mismo criterio expuesto en la implementación de la Raspberry. La solución al control de estado de las persianas quedará propuesta como mejora futura, en el punto [7.3](#).
- **Control de temperatura y humedad:** Para el control de temperatura y humedad se ha recurrido a las librerías DHT11 y Adafruit_Sensors.h7 asociada. La solución que se implementa pasa por generar el código necesario para, que mediante el uso de las citadas librerías, se pueda obtener los valores de temperatura y humedad, para seguidamente, haciendo uso de código html, mostrarlos en el panel de control del servidor web. Una vez, realizado esto, simplemente queda mediante el plugin Script y su opción html, hacer la consulta html para que recupere los valores mostrados en la web, por ejemplo, `#valortemperatura` y activar la opción cron, para que los valores se actualicen periódicamente. Aclarar, que estas peticiones html se pueden realizar correctamente, porque en el código html, a estos valores, se le ha

dado una identificación html (id). Ya, solo se usaría el plugin Virtuel, para generar dos iconos (temperatura humedad) que muestren los valores recuperados.

- **Control de luminosidad:** idénticamente al caso de control de temperatura y humedad, se crea el código necesario, tanto en c, como en html, para obtener dicho valor del sensor, eso si ahora la librería usada es BH1750. Y mediante el uso, de la misma forma que en el control de temperatura y humedad, de Script y Virtuel se integra el icono con el valor correspondiente en el controlador.
- **Control de presencia:** se implementa igual que el control de temperatura humedad y luminosidad. Solo añadir que, en este caso, no es necesario usar ninguna librería y que se ajusta el tiempo de activación en alto nivel del sensor Neuftech HC-SR501 PIR, para que cuando detecte movimiento, el nivel alto dure lo suficiente, como para que se ejecute el código. Así, una vez realizado todo, se obtendrá un control de presencia de la habitación 2, visualizado desde el controlador.

4.4 Sistema de tomas de decisiones (Escenas)

El sistema de tomas de decisiones se refiere a una serie de reglas, que según unos valores o unas condiciones, realizan unas acciones determinadas. Para implementar estas reglas, se usa lo que Jeedom llama escenas. Estas escenas, están divididas en dos partes:

- Por un lado, lo que Jeedom llama disparadores, que son las condiciones que han de darse para que la regla de la escena se inicie.
- Por otro lado, la regla en sí, que es la programación que contiene la tomada de decisiones deseada.

Lo primero que se hace, para poder activar y desactivar la escena a gusto del usuario, es mediante el plugin Virtuel, crear por cada escena un icono que permita activar y desactivar dicha escena.



Figura 38: interruptores virtuales para escenas

Ahora, se procede a explicar la implementación de cada una de las escenas contempladas:

- Subir o bajar persianas según si amanece o atardece: para esta regla se van a usar los plugins Virtuel y Wearher. Con el plugin Wearher, entre otras cosas como previsión del tiempo (figura 38), se genera dos órdenes, una que devuelve la hora del amanecer y otra que devuelve la hora del atardecer. Ahora, simplemente queda en las escenas de Jeedom, introducir como disparadores estas órdenes, amanecer, atardecer y las órdenes (Virtuel) activación y desactivación de persianas. Y por otro lado, configurar la regla que diga - si está activo "activar persiana", a la hora de la orden "amanecer" sube todas las persianas (para el caso del Z-Uno hay que poner a 99 todas las barras deslizadoras) y a la hora de "atardecer" baja todas las persianas (para el caso del Z-Uno hay que poner a 0 todas las barras deslizadoras) - (diagrama de flujo [anexo IV](#)).
- Presencia en el pasillo y poca luminosidad enciende la luz 10 segundos: en este caso, simplemente hay que colocar como disparadores las órdenes: activación escena, desactivación escena, se detecta presencia en el pasillo y que la luminosidad del pasillo baje de un cierto grado. Una vez, creados los disparadores simplemente hay que generar la regla que diga - si "escena activa" evalúa si hay "presencia en el pasillo" y la "luminosidad baja de cierto valor", de cumplirse la condición, se activa la luz del pasillo y se espera 10 segundos para luego desactivarla- (diagrama de flujo [anexo V](#)).
- Calefacción instancia, según temperatura instancia: primero de todo, esta misma regla se realizará tres veces, una para el salón, otra para la habitación 1 y otra para la habitación 2. Ahora, como disparadores se ponen las siguientes órdenes, si se activa la escena, desactiva la escena, temperatura instancia baja de cierta cifra, temperatura instancia sube de una cifra. Una vez, se tiene estos disparadores, se crea la siguiente regla - si "escena activa" evalúa si la "temperatura baja de una cifra", si ha bajado se enciende el radiador de la instancia, ahora se evalúa si la

"temperatura sube de cierta cifra" y de ser así apaga la calefacción, por último si se "desactiva la regla" se apaga la calefacción de la instancia (diagrama de flujo [anexo VI](#)).

- Calefacción general, según temperatura general: La única diferencia con la anterior escena (calefacción instancia según temperatura instancia) es que ahora la temperatura se entiende como media aritmética de todos los sensores de temperatura de la casa y los radiadores eléctricos que se encienden y apagan son todos los disponibles en la casa (diagrama de flujo [anexo VII](#)).
- Alarma: para esta escena primero se usa el plugin telegram, luego, hay que crear un bot de telegram con un usuario de telegram válido y tener el acceso https con certificado válido, para que funcione. Una vez, esté funcionando lo anterior, hay que añadir a los disparadores, la activación de escenario, la desactivación de escenario y la detección de presencia de todos los sensores de presencia disponibles. Ya, solo se crea la regla - si esta activado "activación regla" se evalúa si en cualquiera de los sensores se "detecta presencia", de ser así se envía el mensaje "Alarma: detección presencia" (diagrama de flujo [anexo VIII](#)).

Capítulo 5 Viabilidad técnica.

El proyecto, usa tecnología ya madura, cuya electrónica existe y está desarrollada, por lo que inicialmente y técnicamente, es viable. Además, como se explica a lo largo de esta memoria, los elementos usados, tanto software, como hardware, destacan por estar desarrollados siguiendo una filosofía de código abierto o libre, este hecho implica una alta flexibilidad y, por lo tanto, poder lograr cualquier objetivo, si se tienen unos conocimientos avanzados sobre el problema y la solución. A pesar de ellos, ha habido ciertos puntos críticos relacionados con el protocolo Z-Wave, ya que es, el único de los elementos usados, propietario de Silicon Lab y aunque se ha liberado parte del código, no deja de ser propietario.

Uno de los problemas, más críticos, que ha afrontado este proyecto, ha sido precisamente debido a un fallo en los conocimientos técnicos de la electrónica y el protocolo Z-Wave. Inicialmente, en este proyecto solo se contemplaba usar el protocolo Z-Wave, ya que se creía que la RaZberry podría funcionar como esclavo y así se podría desarrollar, el mismo dispositivo que se ha realizado con Raspberry mediante una red WiFi, pero con una red Z-Wave. La

realidad, es que como se ha comentado en el apartado [2.1.3.1](#) para el desarrollo de dispositivos Z-Wave hay dos plataformas de desarrollo distintas, según se quiera desarrollar un dispositivo (esclavo) o un controlador (master). Además, los microcontroladores que incorpora esta electrónica, también varía, dependiendo si es un dispositivo diseñado para ser esclavo, está diseñado para ser controlador (master). El caso, es que la RaZberry pertenece a los dispositivos que usan, tanto microcontrolador, como plataforma de desarrollo, para ser un controlador (master). Por eso, nunca la RaZberry junto a la Raspberry podrá funcionar como un dispositivo esclavo, aunque usando plugin como Jeelink este conjunto pueda tener incluidos dispositivos esclavos Z-Wave los cuales son controlados desde el controlador principal, el conjunto RaZberry y Raspberry, sería igualmente, un controlador que comparte el control de los dispositivos incorporados en su red, red independiente de la del controlador principal y no un dispositivo esclavo en sí.

El último de los problemas de viabilidad, que ha tenido este proyecto, también se ha dado por escasez de los conocimientos de electrónica y las limitaciones de desarrollar en un entorno doméstico. Para el desarrollo con la MSP432P401R, la idea era incorporarle un microcontrolador Z-Wave, como era el módulo ZM5304 de Silicon Labs. Dicho módulo, integrado en la MSP432, podía ofrecer un interfaz Z-Wave, al estilo que el módulo CC3100 es un interfaz WiFi y, por lo tanto, tener la capacidad de desarrollar, el mismo dispositivo desarrollado con la MSP432, pero con una red Z-Wave. El problema, se ha dado al ir avanzando en el estudio de este módulo ZM5304, se comprobó que, por conocimientos y posibilidades técnicas, en un entorno doméstico, su integración y soldadura iba a ser prácticamente imposibles, entre otras cosas debido a su pequeño tamaño.

Debido, a estos dos problemas detectados durante las tareas de documentación del proyecto, fue necesario cambiar los objetivos y la planificación del proyecto. Dejando de lado, las implementaciones Z-Wave y desarrollando los mismos dispositivos, en cuanto a funcionalidades, pero usando una red WiFi más conocida y que iba a implicar menos problemas.

Capítulo 6 Valoración económica

En este capítulo, solo se pretende realizar una valoración económica de lo desarrollado en el proyecto, usando una filosofía de código abierto y libre y la correspondiente valoración económica, de un sistema equivalente, si se hubiera desarrollado usando dispositivos

completamente comerciales. Así, la valoración económica del proyecto, considerando como mano de obra costes una vez diseñado este primer proyecto, es decir, las horas no corresponde con las horas de realización de este proyecto, si no con una planificación futura estimada, donde se amortizan el diseño de este proyecto, se tiene:

Concepto	Cantidad	Precio unitario €	Precio total €
Dispositivos			
Z-Uno	1	57	57
USB Z-Stick GEN5	1	46	46
RaZberry	1	58,95	58,95
MultiSensor 6 Aeotec	1	49	49
DTH22	1	3,1	3,1
BH1750	1	2,66	2,66
Neuftech HC-SR501 PIR	1	2,43	2,43
Relé 4 canales	4	2,78	11,12
		Total Z-Wave	230,26 €
Raspberry pi B 3+ y complementos	1	59,49	59,49
MSP432P401R	1	18,04	18,04
CC3100 BoosterPack	1	20,74	20,74
DTH22	2	3,1	6,2
BH1750	2	2,66	5,32
Neuftech HC-SR501 PIR	2	2,43	5,06
Relé 4 canales	2	2,78	5,56
		Total WiFi	120,41 €
Mano de obra			
Planificación y diseño	16	24	384
Desarrollo	16	24	384
Instalación hogar	4	24	96
Programación	4	24	96
		Total Mano de obra	960 €
		Total	1310,67 €

Tablas 1: Presupuesto proyecto filosofía libre

La valoración económica, de un sistema equivalente, realizado enteramente con componentes comerciales y considerando el mismo controlador basado en código abierto, algo que abarata el coste considerablemente, frente a uno más comercial, como se vio en el apartado 2.2.1.2, es la siguiente:

Concepto	Cantidad	Precio unitario €	Precio total €
Dispositivos			
Doble Relé ON / OFF ZIPATO Z-Wave Plus	5	59,99	299,95
USB Z-Stick GEN5	1	46	46
Detector de Moho, Temperatura y Humedad - POPP	1	34,99	34,99
MultiSensor 6 Aeotec	1	49	49
Detector de Movimiento - Z-Wave Plus - Hank	1	34,99	34,99
Sensor de luz con pantalla LCD Everspring	1	58,90	58,90
		Total Z-Wave	523,83 €
Shelly 2	6	19,90	119,94
Shelly Sense (presencia humedad,...)	2	99	198
		Total WiFi	317,94 €
Mano de obra			
Planificación y diseño	16		384
Desarrollo	16		384
Instalación hogar	4		96
Programación	4		96
		Total Mano de obra	960
		Total	1801,77 €

Tablas 2: Presupuesto equivalente con elementos comerciales

Se puede apreciar, como gracias a la flexibilidad que la filosofía de código abierto y libre proporciona, se ha llegado a un diseño que es un 27,25% más barato, que una solución equivalente, basada en componentes puramente comerciales. Destacar, que en este caso un proceso de industrialización reduciría los costes del diseño, ya que la capacidad de comprar en cantidades mayores abarata el coste de los componentes. Pero, el mismo resultado se obtendría, si dicha industrialización, se le aplicará a la solución comercial. En definitiva, la industrialización, abarataría el coste, pero no la diferenciaría la solución libre, frente a una solución comercial.

Capítulo 7 conclusiones

1. Para finalizar, los objetivos que se han alcanzado en este proyecto son:
2. Despliegue de una red Z-Wave.
3. Despliegue de la plataforma Jeedom.
4. Integración de dispositivos en la plataforma Jeedom.
5. Despliegue dispositivo comercial Z-Wave.
6. Sistema de control de luz, persiana y calefacción basado en el dispositivo Z-Uno y protocolo Z-Wave.
7. Integración del conjunto Raspberry pi 3 y RaZberry.
8. Sistema de control de luz, persiana y calefacción basado en Raspberry pi y WiFi.
9. Monitorización en los puntos de control de temperatura, humedad, presencia y luminosidad.
10. Sistema automático para:
 - 10.1. Activación luz pasillo.
 - 10.2. Persianas
 - 10.3. Calefacción individual por instancias y general.
 - 10.4. Alarma
11. Control de luces, persiana y calefacción, temperatura, humedad, presencia y luminosidad mediante las placas MSP432 y CC3100 usando WiFi.
12. Diseñar un sistema domótico de bajo coste, flexible, escalable y fácil de instalar.
13. Demostrar el potencial y los beneficios que ofrece usar elementos libres o de código abierto.

Debido, a los problemas detectados y explicados en el apartado [5](#), que tienen que ver con la imposibilidad de desarrollar dispositivos (esclavos) trabajando en una red Z-Wave, con la electrónica contemplada al inicio del proyecto, no se logran alcanzar los siguientes objetivos:

- Sistema de control de luz basado en Raspberry pi y Z-Wave.
- Sistema de control de persianas basado en Raspberry pi y Z-Wave.
- Integración del módulo ZM5304 en la placa MSP432
- Integración en la placa MSP432 y el módulo ZM5304 en Jeedom.
- Desarrollar de dispositivo propio con tecnología Z-Wave basado en MSP432.

7.1. Una descripción de las conclusiones

A la finalización del proyecto, se ha logrado implementar, un sistema domótico, que consta:

- **Z-Uno** instalado en el salón, controla tres persianas del salón, dos puntos de luz del mismo, un punto de luz del pasillo, la calefacción eléctrica del este y la temperatura, humedad, luminosidad y presencia de dicha instancia, vía red Z-Wave.
- **MultiSensor 6** (comercial), instalado en el pasillo, controla la temperatura, humedad, luminosidad, rayos ultravioletas, presencia y vibraciones por manipulación de este, vía red Z-Wave.
- **Raspberry pi**, instalada en la habitación 1, controla la persiana, luz, calefacción, temperatura, humedad, luminosidad y presencia de la habitación 1, mediante una red WiFi.
- **MSP432P401R+CC3100**, instalada en la habitación 2, controla la persiana, luz, calefacción, temperatura, humedad, luminosidad y presencia de la habitación 2, mediante una red WiFi.

Destacar, que este proyecto sobre todo ha trabajado en la integración de distintos elementos de naturaleza muy distinta. Esto, ha provocado ser capaz de trabajar con distintas tecnologías, las cuales, se puede dividir según ramas de conocimiento:

Electrónica y hardware: se ha trabajado con Z-Uno de naturaleza basada en Arduino, Raspberry pi, MSP432 y CC3100 de Texas Instruments, sensores electrónicos digitales y analógicos.

Telemática y comunicaciones: se trabaja con Z-Wave, WiFi, I2C, peticiones HTTP, peticiones html, creación de dominios para la resolución DNS y evitar el problema de la IP dinámica, en certificados y sus conceptos para la seguridad y cifrado, en redirecciones de puertos TCP de los router de los proveedores de servicio.

Software y sistemas: se trabaja sobre una plataforma código abierto como es Jeedom, en los ficheros hosts de Debian, en la creación de un servidor web WiFi, se usan distintos lenguajes de programación como es C, Python y html, plugin creador por terceros de código abierto.

El aspecto final, que el cuadro de mando de Jeedom tiene, después de lo desarrollado es el siguiente:



Figura 39: Cuadro de mandos final

Todo ello, de distinta naturaleza y todo ha quedado integrado comunicándose sin problemas, dado una solución funcional, que responde a las expectativas, a la vez, que es muy flexible, escalable y abaratando el coste un 27,25%

7.2. Autoevaluación.

Con este proyecto, se han logrado todos los objetivos principales y transversales. Al final, lo que se pretendía era crear un sistema domótico de bajo coste, flexible y usando una filosofía de código abierto y libre. Para ello, se ha trabajado en la integración de distintas tecnologías tanto hardware como software, tal como se ha expuesto en el apartado anterior, algo muy interesante de este proyecto, es que con él se han trabajado en todas las ramas, para las que

debe estar preparado un graduado en telecomunicaciones, se ha trabajado en las ramas de electrónica, telemática, programación y sistemas, logrando un sistema funcional y apto.

Si bien, es cierto que hay ciertos objetivos, que tienen que ver con Z-Wave, comentados en el apartado [7](#), que no se han alcanzado, algunos por errores de conceptos iniciales, otros por falta de medios. Debido a estos errores, se han tenido que realizar cambios en la planificación y objetivos descritos en el apartado [1.5](#). Pero, la aparición de estos errores, también son importantes y tienen que ver con un graduado en telecomunicaciones. Ya que esté, que debe ser capaz de detectar los posibles problemas y puntos en los que estos pueden aparecer, para buscar alternativas, de tal modo, que la planificación sea lo menos modificada posible, por lo menos en lo que a plazos se refiere. En el caso de este proyecto, se contempló la posible aparición de errores y se sabía que el punto débil era la red Z-Wave, ya que este protocolo es propietario y novedoso. Por ello, la planificación inicial, hacia un seguimiento en cada fase de las posibilidades y además contempla horas de trabajo reservadas, que han sido necesarias usar, tras detectar los problemas y darles una nueva solución usando WiFi.

También, decir que ciertos desarrollos como, por ejemplo, los controles de persianas tanto el basado en Raspberry, como el basado en MPS432, responden a las expectativas y son funcionales, lo que no los salva de que pueden funcionar de una mejor forma. Lo mismo, se puede decir, a no contemplar ningún escenario en caso de fallos de dispositivos. Esto, al final ha sido una decisión tomada, a favor de premiar el desarrollo de más partes, frente a la mejor opción de funcionalidad de los dispositivos, siempre y cuando el dispositivo final cumpla con su función.

Para finalizar, en este proyecto siempre se ha seguido la planificación y cuando está se modificó, se siguió la nueva sin problemas. Además, destacar que su seguimiento y modificación han sido fáciles, ya que, gracias a la metodología de trabajo, en la que la planificación iba de lo más sencillo, a lo más complejo, se iban adquiriendo en una fase previa, los conocimientos necesarios, para la siguiente fase. Así, se crea un ciclo de aprendizaje, en el cual, conforme se va avanzando en el proyecto se va teniendo el desarrollo muy claro y si son factibles los objetivos, en caso de no serlos se aprecia de una mejor forma, como se han de modificar. Permitiendo, lograr poco a poco los conocimientos necesarios, para llevar el proyecto a su conclusión exitosamente, como ha sido el caso de éste.

7.3. Líneas futuras

En este proyecto, en algunas de las implementaciones se ha buscado la simplicidad, con el objetivo de avanzar en otros objetivos, frente a desarrollar un sistema verdaderamente robusto, sin fallos y con todas las posibilidades controladas. En esta línea, hay un buen trabajo por realizar en los controles de persianas, tanto basados en Raspberry, como en MSP432+CC3100. Estos controles, son funcionales, pero no se les ha implementado un control de estado, para saber en cada momento en que posición está la persiana y poder dejarla de una forma exacta en la posición deseada. Mejorar este funcionamiento y lograr reportar el estado de la persiana, es algo muy interesante en lo que se puede trabajar para mejorar este proyecto.

Otro planteamiento, de cara a mejorar el sistema, es como se planteaba desde el inicio, lograr integrar el sistema en una red enteramente Z-Wave. Para ello, habría que trabajar en la integración de los módulos estilo ZM5304, que ofrecen conectividad Z-Wave, en distintas plataformas existente (Arduino Raspberry, Texas Instruments, ...), contemplando su tamaño y posibilidades. Este punto, es la línea más interesante de desarrollo, para mejorar este proyecto, y seguramente tiene material para un proyecto aparte.

Otro punto de mejora interesante es que en ningún punto de este proyecto se plantean escenarios con posibles respaldos como, por ejemplo, respaldo del controlador. Como ya se ha dicho, la Raspberry con el módulo RaZberry, trabajan únicamente como controlador, así que, podría hacer de segundo controlador, el cual, podría apoderarse del control, en caso de que el controlador principal dejará de funcionar.

Glosario

Arduino: plataforma de hardware y software de código abierto que diseña pequeñas placas con pines de entrada analógicos y digitales para su uso.

Bot: es un usuario autónomo de la red social telegram que son controlados por instrucciones.

C: lenguaje de programación.

CA: autoridad certificadora es aquella autoridad que garantiza que un certificado firmado por ella pertenece a quien dice pertenecer.

Certificado: fichero informático firmado por una CA que verifica que las claves que contienen pertenecen a quien dice.

Controlador: dispositivo central de una red, a través del cual, el resto de dispositivos interactúan con la red.

Dns: Sistema de Nombres de Dominio traduce la dirección IP en el nombre del dominio.

Dominio: Nombre asociado a una dirección IP.

Domótica: aquellos sistemas que automatizan una vivienda.

Escenas: reglas utilizadas por Jeedom para tomar una decisión a partir de unos datos.

Esclavo: elemento final, sin poder de decisión o reducido, que debe responder ante otro (master).

GPIO: Entrada/Salida de Propósito General son una serie de pin, cuyo comportamiento programar.

Htmi: Lenguaje de Marcas de Hipertexto lenguaje de programación que se utiliza para el desarrollo de páginas Web.

Http: Protocolo de Transferencia de Hipertexto permite las transferencias de información en la World Wide Web.

Https: Protocolo de Transferencia de Hipertexto Seguro igual que HTTP pero usando cifrado y añadiendo seguridad.

I2C: dos vías de comunicación, entre el maestro y el esclavo para enviarse información.

Integración: convivencia de elementos de distinta naturaleza funcionando correctamente.

IoT: Internet de las Cosas, sistema de dispositivos comunicados e interrelacionados entre sí.

IP: Protocolo de Internet, número que identifica, a una Interfaz en red.

IP dinámica: IP que suministra un proveedor de internet, pero que va cambiando conforme pasa el tiempo.

Jeedom: Software de código abierto orientado al control domótico. Ofrece una interfaz de configuración monitoreo y manejo del sistema domótico.

Let's Encrypt: autoridad de certificación, basada en una iniciativa de software libre, que proporciona certificados.

Librería: códigos hechos por tercero que se pueden usar para facilitar la implementación de un código.

Maestro: dispositivo con capacidad de computación, toma de decisiones y envió de información a los esclavos, después de haber recibido información de estos.

Market: Mercado de Jeedom donde podemos encontrar plugin desarrollados por otros.

Noip: Servicio gratuito para crear un dominio.

Peticiones http: consulta que envía o solicita información a una página web.

Plugin: aplicación o código realizado por terceros, que responde a una ampliación de funcionalidades, de otro código principal y mayor.

Protocolo: reglas que permiten que diferentes sistemas interactúen.

Proveedores de servicios de internet (ISP): empresa que facilita un acceso a internet.

Puerto: número que se asigna a una comunicación, tanto en origen como en destino, para su identificación.

Python: lenguaje de programación interpretado.

Redirección: regla creada para dirigir cierta información a otro lado.

Servidor web WiFi: página web que espera peticiones de los clientes mediante WiFi.

TCP: protocolo de Control de Transmisión, que hace posible la comunicación entre dos sistemas.

WiFi: protocolo de comunicación inalámbrico que funciona en los 2,4GHz.

ZM5304: módulo autónomo me incluye un modem y todo lo necesario para comunicarse vía Z-Wave.

Z-Uno: placa programable basada en Arduino con una interfaz para la comunicación vía Z-Wave.

Z-Wave: protocolo de comunicación inalámbrico propietario que funciona en los 900Hz.

Bibliografía

Brunet, A., 2018. *domo-blog*. [En línea]

Available at: <https://www.domo-blog.fr/securiser-jeedom-https-certificat-ssl-dote-nouveau-challenge-http-01/>

[Último acceso: 09 Enero 2019].

domodesk, 2018. *domodesk*. [En línea]

Available at: <http://www.domodesk.com/216-a-fondo-zigbee.html>

[Último acceso: 8 Enero 2018].

domodesk, 2018. *domodesk*. [En línea]

Available at: <https://www.domodesk.com/162-a-fondo-z-wave-sin-cables.html>

[Último acceso: 08 Enero 2019].

domoticaudem, 2015. *domoticaudem*. [En línea]

Available at: <https://domoticaudem.wordpress.com/arquitectura-de-los-sistemas/>

[Último acceso: 8 Enero 2018].

energia, 2017. *energia*. [En línea]

Available at: <https://energia.nu/reference/>

[Último acceso: 10 Enero 2019].

ieee, 2019. *ieee*. [En línea]

Available at: https://standards.ieee.org/standard/802_15_4-2015.html

[Último acceso: 08 Enero 2018].

jeedom, 2017. *jeedom*. [En línea]

Available at: <https://www.jeedom.com/market/index.php?v=d>

[Último acceso: 08 Enero 2019].

jeedom, 2017. *jeedom*. [En línea]

Available at: https://jeedom.github.io/documentation/installation/fr_FR/index#tocAnchor-1-9-4

[Último acceso: 09 Enero 2019].

jeedom, 2017. *jeedom*. [En línea]

Available at: https://jeedom.github.io/documentation/installation/fr_FR/index#tocAnchor-1-8

[Último acceso: 09 Enero 2019].

Jeedom, 2018. *github*. [En línea]

Available at:

https://github.com/jeedom/documentation/blob/master/installation/en_US/other.asciidoc

[Último acceso: 08 Enero 2019].

jeedom, 2018. *jeedom*. [En línea]

Available at: <https://www.jeedom.com/site/fr/>

[Último acceso: 04 Enero 2018].

jeedom, 2018. *jeedom*. [En línea]

Available at: <https://www.jeedom.com/market/index.php?v=d>

[Último acceso: 08 Enero 2019].

jeedom, 2019. *jeedom*. [En línea]

Available at: <https://jeedom.github.io/documentation/>

[Último acceso: 08 Enero 2018].

Maestre, A., 2015. *domoticadomestica*. [En línea]

Available at: <http://www.domoticadomestica.com/jeedom-controlador-domotico-z-wave-de-bajo-coste-basado-en-la-raspberry/>

[Último acceso: 08 Enero 2019].

Miller, M., 2016. *openzwave*. [En línea]

Available at:

http://www.openzwave.com/home/news/2016/09/01/sigma_designs_releases_z_wave_interoperability_layer_into_the_public_domain/more

[Último acceso: 10 Enero 2019].

noip, 2018. *noip*. [En línea]

Available at: <https://www.noip.com/>

[Último acceso: 21 octubre 2018].

raspberrypi.org, 2018. *raspberrypi.org*. [En línea]

Available at: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>

[Último acceso: 08 Enero 2019].

silicon, 2018. *silabs*. [En línea]
Available at: <https://www.silabs.com/>
[Último acceso: 08 Enero 2019].

Silicon, 2018. *silabs*. [En línea]
Available at: <https://www.silabs.com/support/getting-started/mesh-networking/z-wave>
[Último acceso: 08 Enero 2019].

Texas, 2018. *ti*. [En línea]
Available at: <http://www.ti.com/tool/MSP-EXP432P401R>
[Último acceso: 08 Enero 2019].

vendomotica, 2015. *vendomotica*. [En línea]
Available at: <http://vendomotica.com/blog/definicion-de-controlador-domotico/>
[Último acceso: 08 Enero 2019].

wikipedia, 2017. *wikipedia*. [En línea]
Available at: <https://en.wikipedia.org/wiki/Z-Wave>
[Último acceso: 28 Diciembre 2018].

wikipedia, 2017. *wikipedia*. [En línea]
Available at: <https://es.wikipedia.org/wiki/Dom%C3%B3tica>
[Último acceso: 08 Enero 2019].

wikipedia, 2018. *wikipedia*. [En línea]
Available at: <https://es.wikipedia.org/wiki/ZigBee>
[Último acceso: 2018 Enero 2018].

wikipedia, 2018. *wikipedia*. [En línea]
Available at: <https://es.wikipedia.org/wiki/WiFi>
[Último acceso: 08 Enero 2019].

zigbee, 2018. *zigbee*. [En línea]
Available at: <http://www.zigbee.org/>
[Último acceso: 21 Diciembre 2018].

z-wave.me, 2018. *z-wave.me*. [En línea]
Available at: <https://z-uno.z-wave.me/>
[Último acceso: 08 Enero 2019].

z-wave, 2017. *z-wave*. [En línea]

Available at: <https://z-uno.z-wave.me/reference/>

[Último acceso: 10 Enero 2019].

zwave, 2018. *zwave*. [En línea]

Available at: <http://zwave.es/>

[Último acceso: 08 Enero 2019].

z-wavealliance, 2018. *z-wavealliance*. [En línea]

Available at: <https://z-wavealliance.org/>

[Último acceso: 8 Enero 2019].

Anexo

Anexo I: Instalación de Jeedom

- Iniciar máquina virtual
- Identifíquese con el usuario y la contraseña elegidos durante la instalación.
- Iniciar como root
- su
- Introduzca la contraseña de root establecida durante la instalación
- Obtener el script Jeedom, hacerlo ejecutable, lanzarlo:

```
wget https://raw.githubusercontent.com/jeedom/core/stable/install/install.sh
```
- Darle permisos de ejecución:

```
chmod +x install.sh
```
- Instalar Jeedom:

```
./install.sh
```
- Ejecutar Jeedom
- Para conocer la dirección IP de la máquina virtual:

```
ip -s -c -h a
```
- La dirección IP, tipo 192.168.0.XX hay que introducirla en el navegador para entrar al entorno de configuración y explotación.

[\(Volver texto\)](#)

Anexo II Proceso de creación de certificado

- Registrarse en noip y registrar un dominio.
- En el router redirigir el puerto https 443 a la IP de la máquina virtual con Jeedom.
- Agrega la siguiente línea al archivo hosts:
`ServerName www.DominioRegistrado.x`
- Luego editar el archivo apache2.conf con el siguiente comando:
`sudo nano /etc/apache2/apache2.conf`
- Agrega la siguiente línea en la parte superior del archivo:
`ServerName www.DominioRegistrado.x`
- Reiniciar Apache:
`sudo /etc/init.d/apache2 restart`
- Y ya se puede generar el certificado, para ello, se va al directorio /etc:
`cd /etc`
- Y se descargan unos archivos a través del comando:
`sudo wget https://dl.eff.org/certbot-auto`
- Se le da permiso de ejecución:
`sudo chmod a+x ./certbot-auto`
- Y se genera el nuevo certificado con el comando:
`./certbot-auto --apache -d www.DominioRegistrado.x`
- Se hace una pregunta, responde 2 para que redireccione, luego confirmar.
- Con esto ya se tiene generado el certificado, con las claves pública y privada para el acceso vía https.

[\(Volver texto\)](#)

Anexo III Flujo Espera persiana

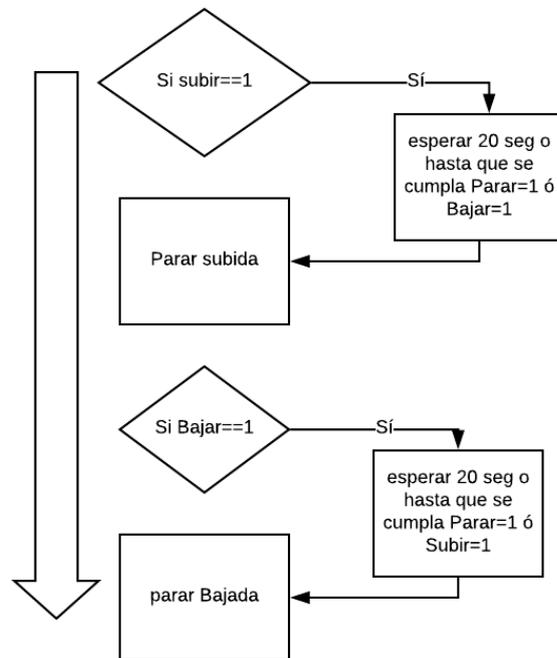


Figura 40: Flujo Espera persiana

[\(Volver texto\)](#)

Anexo IV Flujo persianas amanecer atardecer

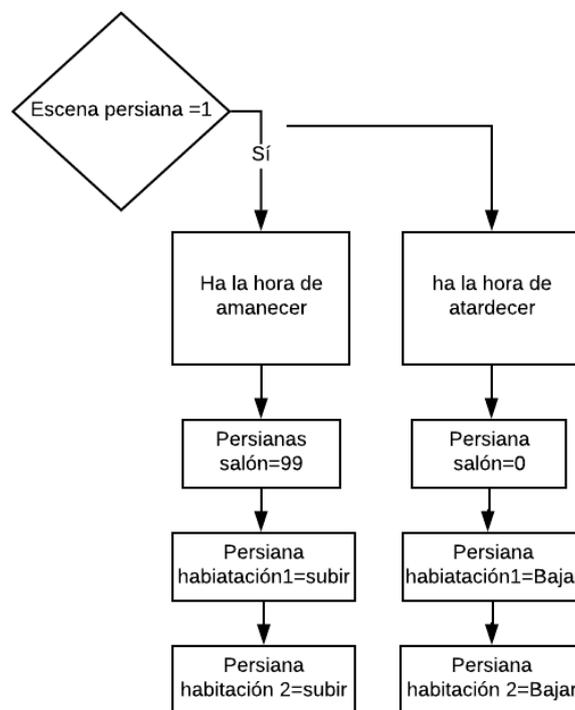


Figura 41: Flujo persianas amanecer atardecer

[\(Volver texto\)](#)

Anexo V Flujo luz pasillo

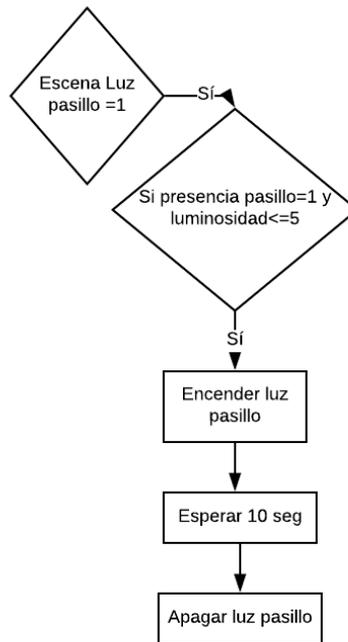


Figura 42: Flujo luz pasillo

[\(Volver texto\)](#)

Anexo VI Flujo calefacción instancia

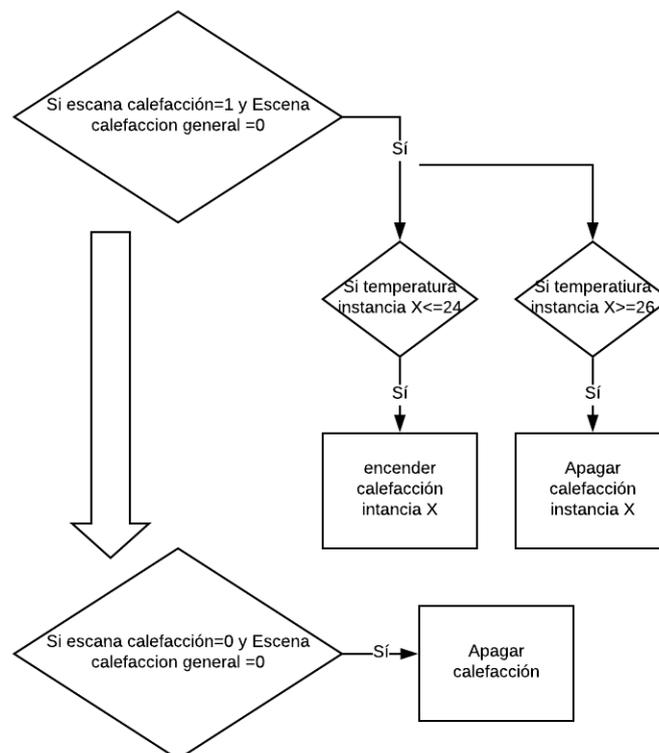


Figura 43: Flujo calefacción instancia

[\(Volver texto\)](#)

Anexo VII Flujo calefacción general

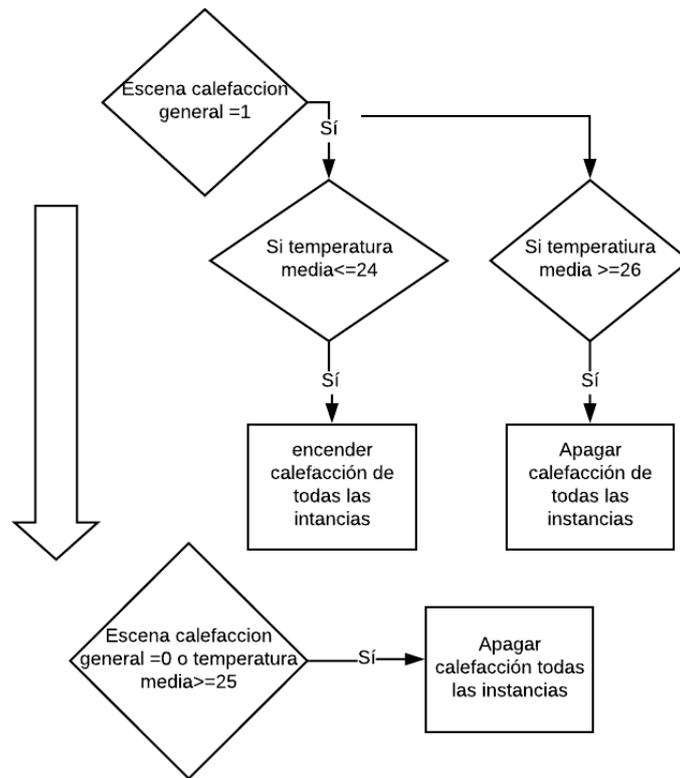
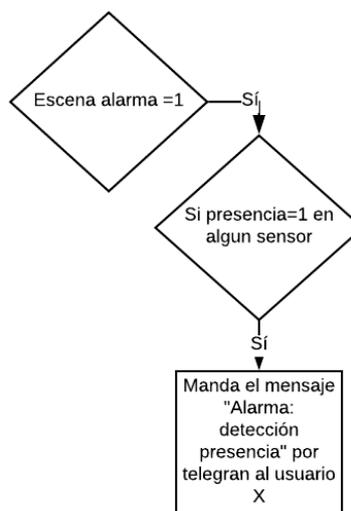


Figura 44: Flujo calefacción general

[\(Volver texto\)](#)

Anexo VIII Flujo Alarma





**Universitat Oberta
de Catalunya**