



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA


Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

**DOMOTIZACIÓN Y CONTROL DE UNA
VIVIENDA**

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

Antonio Andreu Escrivá

TUTORIZADO POR

Houcine Hassan Mohamed

CURSO ACADÉMICO: 2019/2020



RESUMEN

Con este proyecto trato de aplicar un control sobre una vivienda mediante la domotización de ciertos parámetros como la iluminación, climatización y seguridad, que permitan ser controlados mediante una aplicación móvil. De esta manera, se centralizaría todo en un mismo sitio con el objetivo, no sólo de poder interactuar con cada uno de los parámetros de manera convencional y física, sino también poder tener el control de manera remota en cualquier dispositivo en el cual se instale la aplicación móvil, otorgando comodidad al usuario.

RESUM

Amb aquest projecte tracte d'aplicar un control sobre un habitatge mitjançant la domotització de certs paràmetres com la il·luminació, climatització i seguretat, que permeten ser controlats mitjançant una aplicació mòbil. D'aquesta manera, se centralitzaria tot en un mateix lloc amb l'objectiu, no sols de poder interactuar amb cadascun dels paràmetres de manera convencional i física, sinó també poder tindre el control de manera remota en qualsevol dispositiu en el qual s'instal·le l'aplicació mòbil, atorgant comoditat a l'usuari.

ABSTRACT

With this project I try to apply a control on a house through the domotization of certain parameters such as lighting, air conditioning and security, which allow them to be controlled through a mobile application. In this way, everything would be centralized in the same place with the aim of not only being able to interact with each of the parameters in a conventional and physical way, but also being able to have remote control on any device in which the mobile application, giving user comfort.



CONTENIDO

1.	INTRODUCCIÓN.....	5
2.	OBJETO DEL PROYECTO	5
3.	DESARROLLO.....	6
3.1.	LA DOMÓTICA	6
3.2.	PRINCIPALES SERVICIOS DE DOMÓTICA	7
3.2.1	SEGURIDAD.	7
3.2.2	CONTROL ENERGÉTICO.	7
3.2.3	REDES DE TELECOMUNICACIÓN.....	7
3.2.4	CONFORT.....	8
3.3	COMPONENTES PRINCIPALES.....	8
3.4	EDIFICIO INTELIGENTE.....	9
3.5	COMPONENTES DE UN SISTEMA DOMÓTICO.....	11
3.6	TIPOLOGIAS DE REDES DOMÓTICAS.....	12
3.6.1	TIPOLOGÍA EN ANILLO.....	12
3.6.2	TIPOLOGÍA BUS O LINEAL.....	12
3.6.3	TIPOLOGÍA EN ESTRELLA	13
3.6.4	TIPOLOGÍA ESTRELLA EXTENDIDA.....	13
3.6.5	TIPOLOGÍA EN ÁRBOL	14
3.6.6	TIPOLOGÍA EN MALLA.....	14
3.6.7	TIPOLOGÍA MIXTA.....	15
3.7	ARQUITECTURA DE LOS SISTEMAS DOMÓTICOS	16
3.7.1	ARQUITECTURA CENTRALIZADA	16
3.7.2	ARQUITECTURA DESCENTRALIZADA	16
3.7.3	ARQUITECTURA DISTRIBUIDA	17
3.7.4	ARQUITECTURA MIXTA	17
3.8	TRANSMISIÓN DOMÓTICA	18
3.8.1	MEDIOS GUIADOS.....	18
3.8.2	MEDIOS NO GUIADOS	19
3.9	PROTOCOLOS DOMÓTICOS	22
3.9.1	SISTEMAS DOMOMÓTICOS CERRADOS O PROPIETARIOS	22
3.9.2	SISTEMAS ABIERTOS O ESTANDARIZADOS	23
4	SOLUCIÓN ADOPTADA.....	35
5	LA VIVIENDA.....	36
5.1	ESTANCIAS.....	36
5.2	TAREAS QUE REALIZAR	36
5.3	ANÁLISIS DE LA VIVIENDA	37



5.3.1	ENTRADA	37
5.3.2	COCINA	37
5.3.3	GALERÍA	37
5.3.4	COMEDOR.....	37
5.3.5	ARMARIO TRASTERO.....	37
5.3.6	BAÑOS	38
5.3.7	PASILLOS	38
5.3.8	HABITACIONES	38
5.4	ELECCIÓN DE COMPONENTES	38
5.4.1	MICROCONTROLADOR.....	38
5.4.2	CONTROL AUTOMÁTICO DE LUCES	39
5.4.3	DETECTOR DE GAS Y HUMO	41
5.4.4	CONTROL DE TEMPERATURA Y CLIMATIZACIÓN.....	41
5.4.5	CONTROL DE SEGURIDAD.....	42
5.4.6	CONTROL DE PERSIANAS	43
6	CONTROL DOMÓTICO DE LA VIVIENDA	44
6.1	CONEXIÓN ENTRE ARDUINO Y MÓDULO WIFI.....	44
6.2	CONFIGURACIÓN DE LUCES	45
6.2.1	CONTROL LUZ ENTRADA.....	47
6.2.2	CONTROL LUZ ARMARIO TRASTERO.....	48
6.3	CONTROL DE SEGURIDAD FÍSICA	48
6.4	DETECCIÓN DE GAS Y HUMO	49
6.5	CONFIGURACIÓN E INSTALACIÓN DE PERSIANAS	51
6.6	CONFIGURACIÓN DE CLIMATIZACIÓN	53
7	APLICACIÓN MÓVIL ANDROID.....	55
7.1	APP INVENTOR.....	55
7.2	RESUMEN PINES ARDUINO.....	56
7.3	INTERFAZ GRÁFICA.....	57
7.4	FUNCIÓN DE LOS BOTONES.....	59
7.5	CÓDIGO FUENTE DE LA APPLICACIÓN	60
8	PRESUPUESTO.....	62
8.1	CUADRO DE PRECIOS ELEMENTALES.....	62
8.2	CUADRO DE PRECIOS DESCOMPUESTOS.....	63
8.3	COSTE TOTAL CON IMPUESTOS.....	64
9	CONCLUSIONES.....	65
10	BIBLIOGRAFÍA	66
11	ANEXOS.....	68



11.1	ANEXO 1. CÓDIGO COMPLETO	68
11.2	ANEXO 2. CÁLCULO SENSOR DE PROXIMIDAD	77
11.3	ANEXO 3. DECODIFICADOR IR	78
11.4	ANEXO 4. CÓDIGO PROGRAMACIÓN APLICACIÓN DOMÓTICA.....	83
12	PLANO VIVIENDA.....	6886

1. INTRODUCCIÓN

A lo largo de las últimas décadas hemos podido disfrutar de un gran avance a nivel tecnológico que ha supuesto un cambio en la forma de interactuar con el entorno.

Actualmente, gracias a la evolución tecnológica cada vez es más común el concepto de domotización de viviendas con dispositivos sencillos que utilizan la conexión a internet presente en los hogares que permiten controlar la iluminación, la climatización e incluso las alarmas desde el smartphone, así como también su control mediante voz con dispositivos que han adquirido gran popularidad como Alexa o Google Home.

Más allá de la comodidad que pueden aportar a los hogares, resulta evidente que el control de viviendas y edificios va un paso adelante en labores como la seguridad y la eficiencia energética, aportando un beneficio todavía superior del que se podría imaginar.

Es por ello que en este proyecto trataremos de realizar un estudio por los diferentes sistemas domóticos existentes en la actualidad explicando los conceptos teóricos que los caracterizan y, posteriormente, la elección del sistema Arduino como la solución que se adapta mejor, desarrollando la elección de los distintos dispositivos que serán empleados en la instalación.

Finalmente, se incluirán las hojas técnicas necesarias, planos y un presupuesto de este.

2. OBJETO DEL PROYECTO

El siguiente proyecto tiene como objetivo la automatización de una vivienda con la finalidad de aplicar un control mediante la domotización de ciertos parámetros como la iluminación, climatización, control de persianas, cortinas y seguridad.

De esta manera, centralizar en un mismo sitio el sistema para poder interactuar con cada uno de los parámetros, y garantizar la seguridad física tanto de los usuarios como de la vivienda controlando ciertos aspectos como la presencia de intrusos o fugas de gas y humo.

La vivienda está situada en un edificio de cinco plantas en la calle Pintor Renau de Torrent, de 103 m² y el sistema empleado será Arduino, aunque previamente realizaremos un estudio por los distintos sistemas de domotización existentes en la actualidad.

Los principales dispositivos permitirán su encendido, apagado, control de los distintos sistemas de seguridad como alarmas, detectores de humo, iluminación, climatización, etc.

Además, se utilizará el acceso a banda ancha y telefonía existente ya en la vivienda.

3. DESARROLLO

3.1. LA DOMÓTICA

La palabra domótica proviene de la unión de dos palabras: *domus*, que en latín significa casa, y *tica*, que en griego significa autónoma. (www.iecor.com, s.f.)

La domótica se puede definir de manera más cotidiana como la automatización de ciertos parámetros de un lugar con el objetivo de crear independencia a la hora de interactuar con ellos de manera directa, mediante una red de comunicación.

También se puede definir como las tecnologías que permiten de manera inteligente el control y la automatización de un lugar, ya sea una vivienda o un edificio completo, con la finalidad de obtener una gestión óptima de la energía, un control de la seguridad, así como una comunicación directa entre el sistema de control y el usuario, que permitirá ajustarlo según sus necesidades.

Es importante destacar en este ámbito la gestión eficiente de la energía y la seguridad de las personas ya que tanto en una vivienda como en cualquier edificio resultan imprescindibles, así como la facilidad de interacción de los usuarios, buscando el método que permita adaptarse a cualquiera de una manera óptima y, sobre todo, sencilla.

Es común también la utilización del término hogar digital u hogar inteligente para referirse a la domótica, pues ya era utilizado antes de la aparición de este término para referirse a opciones futuristas que no se encontraban todavía en el mercado y que difieren en cierta manera de lo que encontramos actualmente. Sin embargo, hogar digital es un término que se ha estandarizado y que sirve como complementario a la utilización del término domotización.

En resumen, hogar digital o domotización es la integración inteligente de dispositivos y redes de comunicación que permiten el control automático de ciertas actividades de los usuarios ya sea local o remota.

3.2. PRINCIPALES SERVICIOS DE DOMÓTICA

La domótica se fundamenta principalmente en cuatro servicios básicos que constituyen los pilares en los que se desarrollará el proceso.

3.2.1 SEGURIDAD. Consideramos el aspecto de seguridad como uno de los más importantes de los sistemas domóticos ya que su principal objetivo es garantizar el bienestar físico de las personas y los bienes que conforman el lugar.

- Personas: detectores de humo, detectores de fuga de agua o gas, control de presencia para alumbrado, avisos de emergencia, control remoto de enchufes e interruptores, etc. son algunos ejemplos que tratan de garantizar la seguridad de las personas.
- Bienes: alarmas técnicas para la detección de incendios, fugas de gas, etc.; detección de intrusos mediante distintos sensores de apertura de puertas y ventanas, así como roturas de cristales, detectores de presencia, etc.

3.2.2 CONTROL ENERGÉTICO. Este servicio permite la optimización de los recursos para conseguir equilibrar los consumos reduciendo así los costes.

- Iluminación y climatización: utilizar sensores de presencia para activar automáticamente las luces, programación de luces y enchufes para activarlos a determinadas horas del día en función de las horas de luz, programación de la climatización, etc.
- Controles generales: utilización de contadores electrónicos que permitan controlar en todo momento los consumos y posibles fallos, utilizar franjas de tarificación adaptadas para algunos equipos, detección de pérdidas en sistemas de climatización para controlar si hay puertas o ventanas abiertas, control remoto de luces y enchufes para poder activar o desactivar en cualquier momento, etc.

3.2.3 REDES DE TELECOMUNICACIÓN. La evolución tecnológica de la gran variedad de dispositivos que encontramos hoy en día, así como su intercomunicación dan una especial importancia a este punto, pues no podríamos entender la domótica sin tener dispositivos conectados entre sí.

Puntos que destacar:

- Comunicación tanto con el interior como el exterior. En una vivienda destacarían la presencia de intercomunicadores, dispositivos de alarma

acústicos, etc.; y la relación con el exterior con la utilización de internet, redes de telefonía, sistemas de videoconferencia, etc.

- Control remoto de la instalación mediante protocolos TCP/IP, autenticación de acceso remoto, mensajería telefónica SMS, etc.

3.2.4 CONFORT. Este término hace referencia a la capacidad que tiene una vivienda de adaptarse de forma autónoma para mejorar el bienestar de los usuarios.

Tener la capacidad de controlar todo aquello que se quiera o, incluso, que de manera automática sea capaz de realizar ciertas acciones sin el control del usuario, convierten una vivienda en confortable.

En domótica, podemos dividir el control de dispositivos, entre otros, en:

- Control de climatización
- Control de iluminaria
- Control de riego
- Generación de escenas

3.3 COMPONENTES PRINCIPALES

Los distintos componentes que forman la red domótica deben estar conectados entre sí mediante una red interna ya sea inalámbrica o cableada y, además, deben poder trabajar de manera conjunta. Dicha red debe estar a su vez conectada con la red exterior mediante redes públicas como RDSI o RTC.

La red interna se divide en tres partes:

- **Red de control:** Es la encargada de hacer la conexión entre todos los dispositivos de la red domótica. Esta red es la típica de banda ancha que encontramos hoy en día en viviendas particulares, oficinas o escuelas ya que sólo será utilizada para el envío de comandos.
- **Red de datos:** conecta ordenadores y periféricos entre sí de manera independiente a la red de control.
- **Red multimedia:** es utilizada para compartir grandes cantidades de información entre los distintos dispositivos electrónicos inteligentes.

Todas estas redes conviven de manera independiente gracias a la existencia de una pasarela residencial.

3.4 EDIFICIO INTELIGENTE

Para hacer referencia a edificio inteligente tenemos que recurrir al término *inmótica*. Esta palabra hace referencia a la aplicación en edificios no residenciales de las nuevas tecnologías y su uso se debe, principalmente, a que se persiguen objetivos diferentes en la automatización de los que se persiguen con la *domótica*, pues ésta se centra especialmente en la automatización de viviendas.

Los sistemas actuales tratan de mejorar la eficiencia energética de los sistemas eléctricos y electrónicos con el objetivo de poder subsanar las ineficiencias de energía, no sólo para reducir el impacto económico sino también de los recursos naturales. Es por ello que la aplicación de la *inmótica* garantiza un control en la gestión energética, así como también mejorar la seguridad.

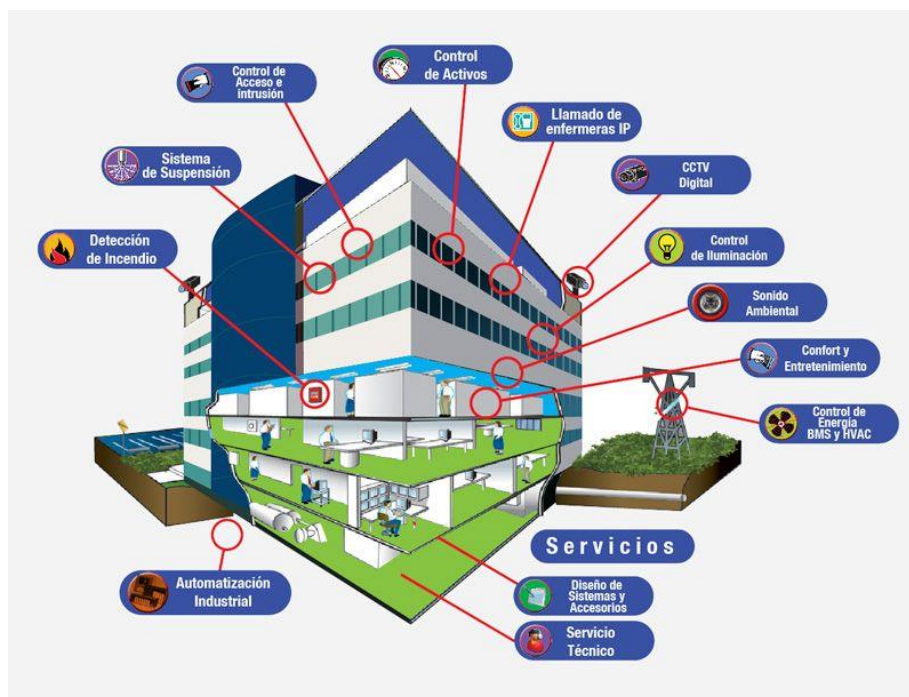


Ilustración 1. Resumen concisos domótica e inmótica

La *inmótica* permite la creación, por tanto, de edificios inteligentes y presenta una serie de ventajas entre las que se pueden destacar:

- Aviso de averías
- Supervisión de consumo eléctrico
- Supervisión en tiempo real de eventos
- Alarmas



- Ahorro energético
- Ahorro en mantenimiento
- Gestión remota

Las aplicaciones que ofrece la inmótica en un edificio son muy diversas y de mucha utilidad. Podemos destacar:

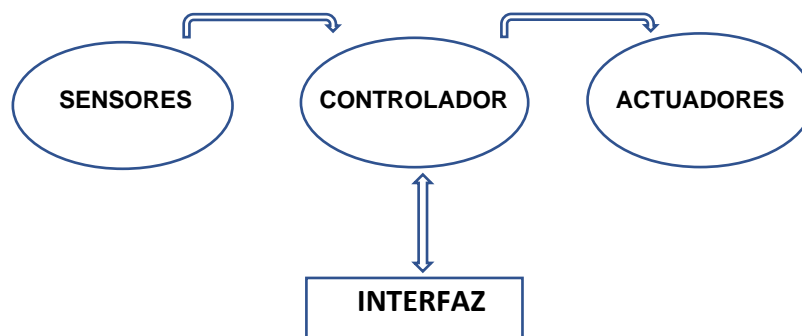
- Control de iluminaria
- Control de acceso
- Sistema de seguridad
- Gestión de la energía
- Control de climatización
- Sistema antiincendios
- Alarmas de fallos
- Control de riego
- Control de ascensores
- Control eléctrico

En resumen, la inmótica se centra en la mejora de la eficiencia energética y económica de un edificio, mientras que la domótica se centra en la mejora de la comodidad y el confort de los usuarios de una vivienda. De esta manera podemos decir que la inmótica integra la domótica interna dentro de una estructura en red.

3.5 COMPONENTES DE UN SISTEMA DOMÓTICO

El sistema domótico tiene como función recopilar información, procesarla y ejecutar órdenes. Estas funciones se realizan con los siguientes componentes básicos:

- **Sensores.** Se trata de dispositivos encargados de detectar acciones externas, ya sea presión, temperatura, etc. para convertirlas en señales eléctricas. Lo sensores más utilizados en domótica son los sensores de presencia, humedad, temperatura o gases.
- **Actuadores.** Su función consiste en modificar ciertas magnitudes del entorno para realizar la acción requerida por el controlador. Se clasifican en analógicos, binarios y digitales, según la señal que utilizan.
- **Controladores.** Su función es la de emitir órdenes a los actuadores según para lo que se hayan programado. En otras palabras, procesan la señal que reciben de los sensores para transmitir a los actuadores las funciones para las cuales se han programado.
- **Interfaces.** Son los mediadores de todo el sistema. Su función es la de comunicar todos los elementos del sistema o, incluso, conectar varios sistemas diferentes entre sí.



Esquema 1. Componentes sistema domótico

3.6 TIPOLOGÍAS DE REDES DOMÓTICAS

Las redes de comunicación son las encargadas de dar vida a los sistemas domóticos que se encargan de conectar el sistema mediante recursos compartidos. Estas redes están formadas por nodos interconectados según el medio de transmisión que se haya escogido.

La topología de red hace referencia a la forma en la que estos nodos están conectados y, según la forma de dicha conexión, las topologías se clasifican en:

3.6.1 TIPOLOGÍA EN ANILLO

Cada estación tiene una única conexión de entrada y otra de salida de anillo. Cada estación tiene un receptor y un transmisor que hace la función de traductor, transmitiendo la señal a la siguiente estación.

Este tipo se da por el paso de un testigo que recoge y entrega paquetes de información, evitando pérdidas de información debidas a colisiones.

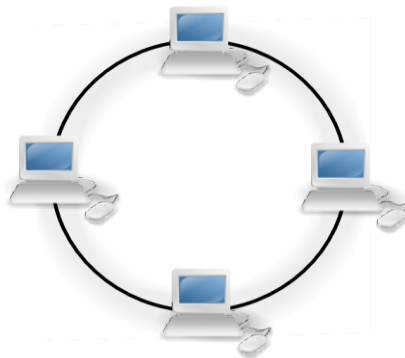


Imagen 1. Tipología en anillo

3.6.2 TIPOLOGÍA BUS O LINEAL

Se caracteriza por tener un único canal de comunicaciones (*bus*) donde se conectan los diferentes dispositivos. Así, todos los dispositivos comparten el mismo canal para comunicarse entre sí. El *bus* es un elemento pasivo ya que no produce regeneración de señal.

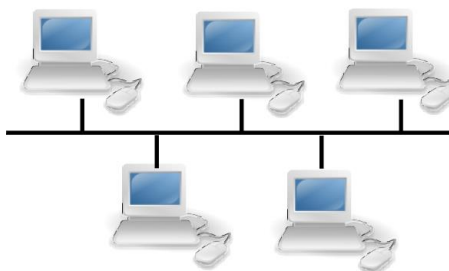


Imagen 2. Tipología bus o lineal

3.6.3 TIPOLOGÍA EN ESTRELLA

Es una red donde todas las estaciones están conectadas directamente a un punto central y todas las comunicaciones se hacen a través de este punto (conmutador, repetidor o concentrador). Los dispositivos no están directamente conectados entre sí, además de que no se permite tanto tráfico de información. Esta tipología es habitual en sistemas centralizados, pero uno de sus principales inconvenientes es que si hay un fallo en el nodo central falla todo el sistema.

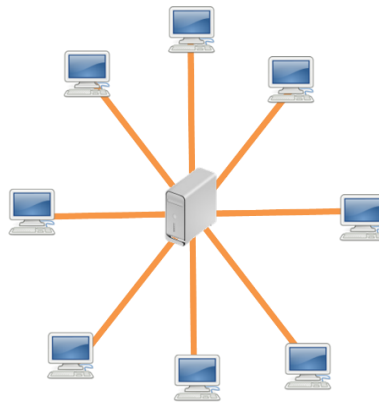


Imagen 3. Tipología en estrella

3.6.4 TIPOLOGÍA ESTRELLA EXTENDIDA

Es igual a la topología en estrella, sólo que cada nodo que se conecta con el nodo central también es el centro de otra estrella. Generalmente el nodo central está ocupado por un *hub* o *switch*, y los nodos secundarios por *hubs*.

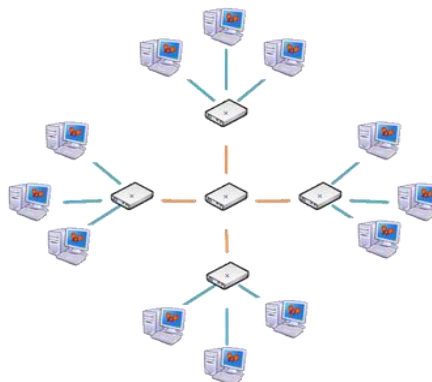


Imagen 4. Tipología estrella extendida

3.6.5 TIPOLOGÍA EN ÁRBOL

Los nodos están colocados en forma de árbol. Es parecida a una serie de redes en estrella interconectadas salvo en que no tiene un nodo central. En cambio, tiene un nodo de enlace troncal, ocupado por un *hub* o *switch*, desde el que se ramifican los demás nodos. Como ventaja presenta que un fallo no implica una interrupción en las comunicaciones.

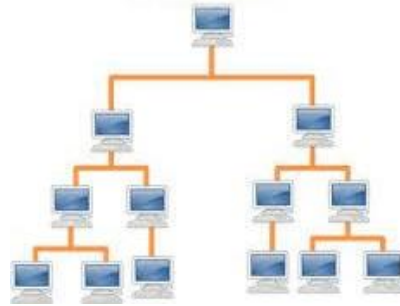


Imagen 5. Tipología en árbol

3.6.6 TIPOLOGÍA EN MALLA

Cada nodo está conectado a todos los nodos. De esta forma es posible llevar los mensajes de un nodo a otro por distintos caminos. Si la red de malla está completamente conectada, no puede existir ninguna interrupción en las comunicaciones. Cada servidor tiene sus propias conexiones con el resto de los servidores.

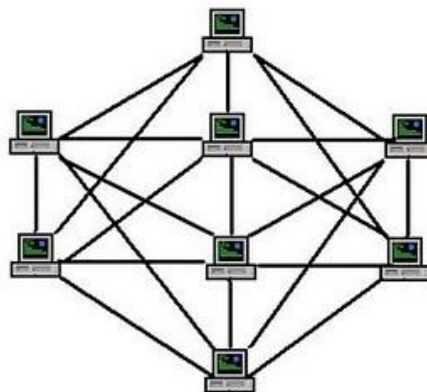


Imagen 6. Tipología en malla

3.6.7 TIPOLOGÍA MIXTA

Es la combinación en una red de varias de las topologías anteriormente descritas.

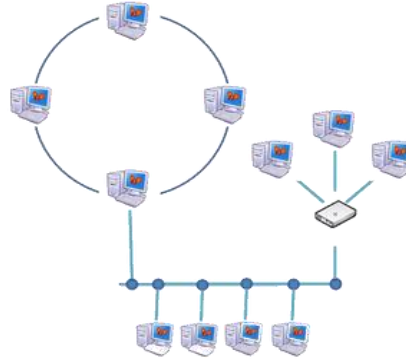


Imagen 7. Tipología mixta

Información de este apartado recopilada e imágenes extraídas de la web:

<https://clasificaciondelasredesblog.wordpress.com>

3.7 ARQUITECTURA DE LOS SISTEMAS DOMÓTICOS

La arquitectura de los sistemas domóticos hace referencia a la forma de distribuir los distintos componentes que forman el sistema, así como la forma de la instalación. Según su distribución, se nombran cuatro tipos de arquitecturas que son:

3.7.1 ARQUITECTURA CENTRALIZADA

En este tipo de arquitectura un controlador centralizado se encarga de enviar la información a los actuadores e interfaces según el programa, la configuración y la información que recibe de los distintos sensores, sistemas interconectados y los usuarios.

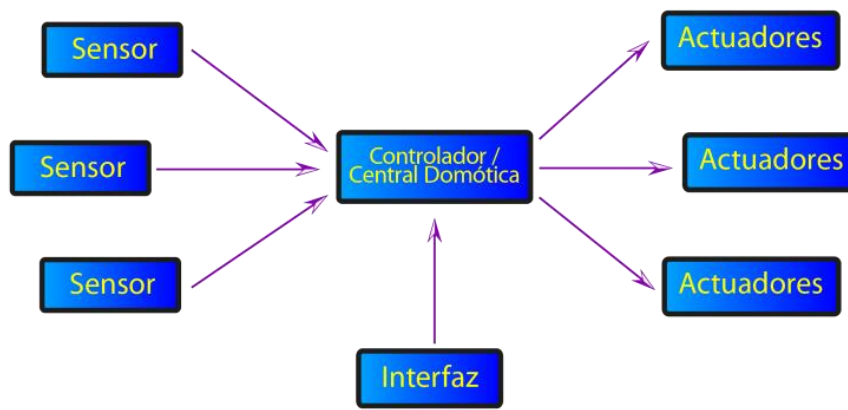


Imagen 8. Arquitectura centralizada

3.7.2 ARQUITECTURA DESCENTRALIZADA

En este tipo de arquitectura hay varios controladores, interconectados por un bus, que envía información entre ellos y a los actuadores e interfaces conectados a los controladores, según el programa, la configuración y la información que recibe de los distintos sensores, sistemas interconectados y los usuarios.

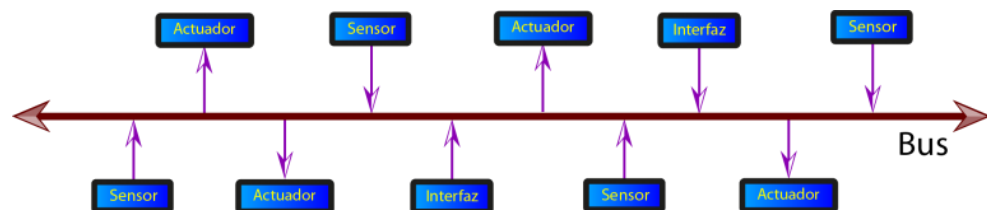


Imagen 9. Arquitectura descentralizada

3.7.3 ARQUITECTURA DISTRIBUIDA

En este tipo de arquitectura cada sensor y actuador es también un controlador capaz de actuar y enviar información al sistema según el programa, la configuración y la información que es capaz de captar por sí mismo y la que recibe de los otros dispositivos.

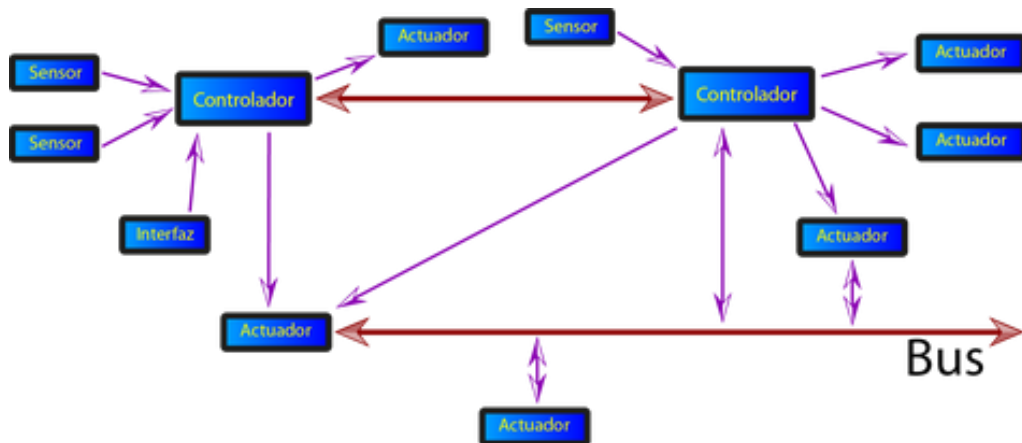


Imagen 10. Arquitectura distribuida

3.7.4 ARQUITECTURA MIXTA

En este tipo de arquitectura se combinan las arquitecturas descritas anteriormente. A la vez que puede disponer de un controlador central o varios descentralizados, los dispositivos de interfaces, sensores y actuadores pueden ser controladores y procesar la información según el programa, la configuración y la información que es capaz de captar por sí mismo y actuar enviándola a otros dispositivos de la red sin que pase por otro controlador.

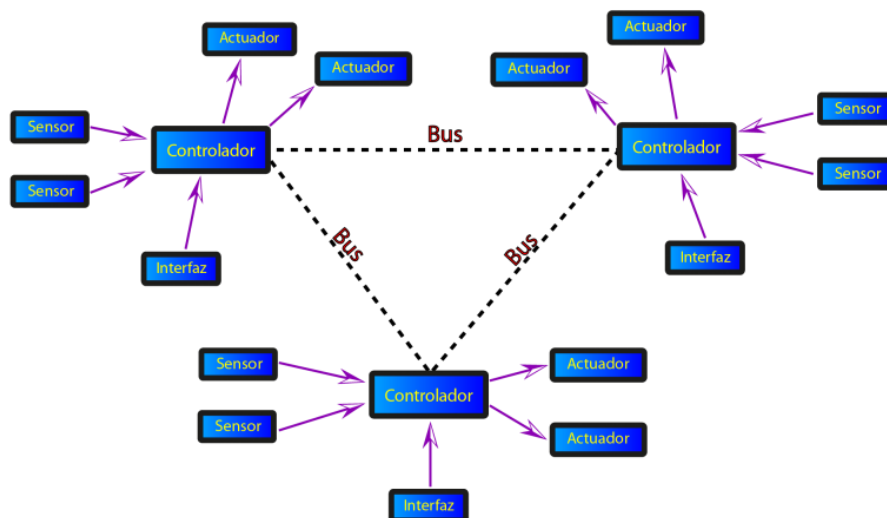


Imagen 11. Arquitectura mixta

Información de este apartado recopilada e imágenes extraídas de la web:

<http://domotica1003.weebly.com/>

3.8 TRANSMISIÓN DOMÓTICA

La transmisión es el medio de comunicación que conecta el emisor y el receptor.

En domótica, la elección de un medio u otro dependerá principalmente de las necesidades que vienen marcadas según la distancia entre emisor y receptor, así como la velocidad de transferencia.

Distinguimos dos tipos principales que son los medios guiados o no guiados.

3.8.1 MEDIOS GUIADOS

Son aquellos que transmiten de manera física como cables de cobre o fibra óptica. Entre los principales utilizados en domótica destacan:

3.8.3.1. PLC (Power Line Communication)

Este medio utiliza la red eléctrica para transmitir datos. Gracias a ello, desde cualquier conexión a la red eléctrica mediante el uso, por ejemplo de enchufes, es posible conectarse a la red.

Esta tecnología convierte la red eléctrica en una línea de datos digital de alta velocidad. Su principal ventaja reside en que es posible acceder a internet y el control del sistema siendo, además, un sistema muy económico debido a la utilización ya existente de la red eléctrica del edificio o lugar donde se quiera domotizar. Su principal desventaja es que presenta grandes problemas de seguridad, así como problemas debido a las interferencias propias producidas por toda red eléctrica.

3.8.1.2 CABLE COAXIAL

Se trata de un medio de transmisión de datos que contiene dos conductores concéntricos donde el núcleo central es el encargado de transmitir la información y la malla que lo envuelve es la derivación a tierra. Entre los dos se encuentra un dieléctrico que se encarga de aislar y todo ello está recubierto por otro aislante.

Su ventaja principal es que es más inmune a interferencias, pero su rigidez lo hace difícil de manejar. Además, su velocidad de transmisión no es muy elevada y por ello, se ha ido sustituyendo en los últimos tiempos por la fibra óptica.

3.8.1.3 PAR TRENZADO

Se trata de un cable compuesto por dos hilos conductores entrelazados y aislados para poder evitar interferencias externas, así como la propia producida por los cables adyacentes.

Se entrelazan los hilos conductores para reducir las interferencias entre ellos mismos. Además, cada uno de estos cables está compuesto por varios pares trenzados que se identifican por colores recubiertos siempre por aislante.

Es uno de los cables más utilizados por ser un estándar en las redes LAN, aunque también está siendo desplazado por la fibra óptica como el cable coaxial. Entre sus principales ventajas destaca su bajo coste y flexibilidad, aunque es más vulnerable ante interferencias.

3.8.1.4 FIBRA ÓPTICA

La fibra óptica es una fibra flexible, transparente, formada a partir de embutir o extruir vidrio o plástico en un diámetro muy fino, ligeramente más grueso que un cabello humano.

Está formada por un núcleo generalmente de óxido de silicio y una corteza exterior. La luz que incide sobre la corteza se refleja hacia el interior puesto que tiene un núcleo de refracción inferior al del núcleo, quedándose concentrada en el mismo y permitiendo una rápida transmisión.

Es el medio de transmisión más avanzado que existe hoy en día a nivel comercial ya que presenta inmunidad ante interferencias electromagnéticas. Su ventaja principal y que lo diferencia ampliamente del resto se debe a que permite enviar grandes cantidades de datos a larga distancia y a velocidades muy elevadas, además es de un peso y tamaño muy reducido. Su menor necesidad de regenerar señal permite utilizar tramos largos de cable evitando así el uso de repetidores y permitiendo ahorrar costes.

3.8.2 MEDIOS NO GUIADOS

Son aquellos que transmiten de manera inalámbrica. Entre los principales utilizados en domótica destacan:

3.8.2.1 RADIOFRECUENCIA

Este medio de transmisión inalámbrica permite el control a distancia del sistema domótico, pero es muy sensible a alteraciones electromagnéticas. Entre los principales sistemas de radiofrecuencia destacan tres: *Bluetooth*, *Zigbee* y *WiFi*.

- **Bluetooth.** Es un medio de transmisión empleado en redes inalámbricas de área personal (WPAN) y permite transferir datos y voz entre dispositivos con un radio corto de hasta 10 metros y utilizando una radiofrecuencia en la banda de 2.4GHz.

En domótica se puede emplear para diversos usos entre los cuales podemos destacar la conexión entre cámaras de seguridad y dispositivos móviles, evitando así la instalación de cableado.

Sin embargo, en domótica existe una alternativa al Bluetooth más utilizada como ZigBee.

- **ZigBee.** Pertenece al grupo ZigBee Alliance y está basado en el estándar del IEEE 802.15.4 de redes WPAN. Su nombre hace referencia al conjunto de protocolos de alto nivel de comunicación inalámbrica utilizados para la radiodifusión digital de bajo consumo. Además, su topología de red en malla hace que sea muy fácil su integración.

Pese a ser una tecnología similar al Bluetooth, presenta ciertas ventajas frente a este:

- Una red ZigBee puede constar de un máximo de 65535 nodos distribuidos en subredes de 255 nodos, frente a los 8 del Bluetooth.
- Menor consumo eléctrico.
- Velocidad de hasta 250 kbits/s frente a los 3000 kbits/s del Bluetooth.
- Estas velocidades hacen que cada uno de los dos sea apropiado para cosas diferentes. El Bluetooth se aplica a aplicaciones como los teléfonos móviles o informática casera mientras que ZigBee es más apropiado para domótica, sensores médicos o artículos de juguetería, entre otros, ya que necesitan una transferencia de datos menor.

- **WiFi.** Permite la conexión inalámbrica de dispositivos electrónicos, que pueden conectarse entre sí o mediante un punto de acceso de red inalámbrica.

Existen varios tipos de wifi basados en un estándar IEEE 802.11, entre los cuales destacan el IEEE 802.11b, IEEE 802.11g e IEEE 802.11n los cuales son los más aceptados internacionalmente gracias a que la banda de 2.4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente.

También existe desde 2013 el estándar IEEE 802.11ac, WiFi 5, que trabaja en la banda de 5 GHz y el cual permite operar a una frecuencia mayor, aunque su alcance es más limitado.

La principal ventaja de este tipo de conexión es la ausencia de cableado y el ahorro de costes, debido a que una vez instalada la conexión se pueden conectar cuantos dispositivos se quiera.

Entre las desventajas destacan la menor velocidad frente a una red cableada, la seguridad, ya que algunos programas son capaces de capturar paquetes pudiendo calcular la contraseña de red y acceder a ella. Además, no es compatible con otro tipo de conexiones sin cable como Bluetooth y la potencia del wifi se ve directamente afectada por los agentes físicos como paredes, árboles, montañas, etc.

3.8.2.2 INFRARROJOS

Es un estándar de conexión que permite la transmisión y recepción de datos mediante rayos en el espectro infrarrojo. Es muy común su uso en mandos a distancia, aunque actualmente se encuentra más en desuso debido a tecnologías de radiofrecuencia pese a tener la ventaja de ser inmune a interferencias electromagnéticas por tratarse de un medio óptico.

Información de este apartado recopilada en webs y libro:

<http://domotica1003.weebly.com>

<https://es.wikipedia.org>

Instalaciones domóticas Edición 2020

Escrito por Luis Miguel Cerdá Filii, Manuel Gas Bueno

3.9 PROTOCOLOS DOMÓTICOS

En la actualidad existen gran variedad de protocolos domóticos los cuales se diferenciarán principalmente por el formato de los mensajes intercambiados por los elementos del sistema para comunicarse entre ellos.

Debido a la gran variedad de protocolos existentes surge el problema que todos no son compatibles entre sí, limitando la elección de estos ya que no es posible interconectar dispositivos de diferentes fabricantes. Además, la reducción de posibles ampliaciones futuras obliga a recurrir siempre al mismo fabricante.

Podemos diferenciar entre dos tipos de protocolos domóticos, sistemas abiertos o estandarizados y sistemas cerrados o propietarios, siendo este último más numeroso.

3.9.1 SISTEMAS DOMÓTICOS CERRADOS O PROPIETARIOS

Se trata de protocolos específicos de una marca en concreto. Se denominan cerrados porque sólo son utilizados por la propia marca y son los que pueden realizar mejoras y fabricar dispositivos que puedan interconectarse. Esto limita la evolución tecnológica de los sistemas domóticos y, por esta razón, conforme evolucionan los sistemas con protocolo estándar van abarcando más cuota de mercado que los propietarios.

A continuación, un ejemplo de sistema propietario.

3.9.1.1 EnOcean

Este sistema transforma energía que se encuentra en el entorno del periférico o que se genera cuando se activa en energía eléctrica.

Cuando se pulsa un interruptor domótico EnOcean, la energía producida al pulsar la tecla es transformada en energía eléctrica gracias a generadores magnetorresistivos o piezogeneradores. Otros dispositivos funcionan con células fotoeléctricas, termo pares u otros sistemas de conversión de energía.

La principal ventaja que tiene es que permite tener dispositivos sin pilas y, por tanto, sin necesidad de mantenimiento. Aun así, algunos dispositivos sí admiten pilas para seguir funcionando en condiciones extremas, cuya vida útil de la misma puede ser superior a 3 años.

La energía que consiguen estos módulos domóticos es suficiente para enviar datos al controlador vía inalámbrica teniendo un alcance teórico de unos 300 metros trabajando en la frecuencia de 868 Mhz en Europa. Por ello, según la propia marca recoge en su eslogan, EnOcean funciona *sin cables, sin pilas y sin límites*.

La diferencia principal que encontramos entre EnOcean y Z-Wave es que los primeros no están en permanente contacto con el controlador ya que solo informan de su estado cuando reciben la energía suficiente.

Los módulos EnOcean son muy económicos y se pueden integrar en sistemas Z-Wave, pudiendo encontrar entre ellos interruptores, sensores de apertura o temperatura, etc.



enocean® alliance

No Wires. No Batteries. No Limits.

Imagen 12. EnOcean

3.9.2 SISTEMAS ABIERTOS O ESTANDARIZADOS

Se trata de protocolos definidos entre varias compañías con el fin de unificar criterios. Son protocolos abiertos, por tanto, no existen patentes sobre el protocolo, así cualquier fabricante puede desarrollar sus aplicaciones y productos llevando implícito este protocolo de comunicación.

La principal ventaja que tienen este tipo de sistemas es que no repercute si una empresa desaparece o deja de desarrollar productos ya que existen otros fabricantes que trabajan sobre el mismo sistema.

A continuación, resumiremos tres de los principales sistemas abiertos o estandarizados centrándonos en el sistema KNX.

3.9.2.1 Z-Wave

Se trata de un sistema de comunicaciones interoperable e inalámbrico, basado en la comunicación RF diseñado para aplicaciones de control, monitorización y lectura de estado en ambientes residenciales y comerciales. Es líder mundial en control inalámbrico, con productos económicos, fiables y fáciles de usar.

Entre sus principales ventajas destacan:

- Es inalámbrico, por tanto, no necesita obra ni cableado.
- Utiliza tipología de mallado y puede incluir hasta 232 equipos con una separación de 20 metros interiores y 30 exteriores.
- La red es posible ampliarla con el uso de repetidores.
- Utiliza la banda 800-900 Mhz, lo cual evita la interferencia con WiFi.
- Se accede a los dispositivos por internet.
- Buena integración con dispositivos de audio y vídeo.
- Precio económico.

Muchos dispositivos pueden ser compatibles con la red de Z-Wave lo cual permite un amplio abanico de posibilidades con el uso de este sistema domótico.



Imagen 13. Z-Wave

3.9.2.2 Modbus

Este sistema trabaja mediante la relación Maestro-Esclavo. Se trata de un sistema unilateral y recíproco ya que, para que un dispositivo funcione, primero debe iniciar un mensaje y después esperar que el otro equipo envíe una respuesta.

Por ello, el Maestro es el principal responsable de iniciar una actividad que se realiza normalmente mediante un interfaz HMI o sistema SCADA. El Esclavo, puede ser un sensor o dispositivo que responde a través de un controlador PLC o PAC.

Las solicitudes y respuestas de este sistema son enviadas entre ambas partes por lo que se conoce como capas de protocolo, aunque Modbus utiliza una base a serial con la cual no puede ser dividida en múltiples capas.

Utiliza las redes TCP/IP y UDP para separar protocolo principal y poder definir la capa de red y controlar la unidad de datos de aplicación.



Imagen 14. Modbus

3.9.2.3 BACnet

Se trata de uno de los protocolos más potentes del mercado. Es utilizado mayoritariamente en edificios ya que comprende funciones de climatización, seguridad y ahorro energético.

Fue creado por ASHRAE en el año 2003 y funciona bajo el estándar abierto ISO 16484-5 desarrollado para la automatización de un edificio.

A diferencia de otros protocolos domóticos, este incluye servicios Who-Is, I-am, Who-Has y I-Have, generalmente usados para detectar objetos en habitaciones.

Además, también incorpora funciones como ventilación, refrigeración, iluminación, control de puertas y cámaras.



Imagen 15. BACnet

3.9.2.4 X-10

Es uno de los protocolos más antiguos pues apareció entre 1976 y 1978 en Escocia. Su principal característica es la transmisión de datos a muy baja velocidad utilizando las líneas de baja tensión y a muy bajo coste.

La principal ventaja de este sistema es la no utilización de nuevos cables, pues utiliza los propios de la línea eléctrica lo que lo convierte en una opción muy económica para la instalación de sistemas domóticos no muy complejos.

Cualquier fabricante puede crear y distribuir dispositivos basados en X-10 pero debe utilizar los circuitos del fabricante escocés que creó la tecnología, lo cual no supone un problema pues no significa que se trate de un protocolo propietario y los circuitos integrados son de muy bajo coste en comparación con otros.

Emplea la arquitectura tipo bus debido a la utilización de la línea eléctrica como medio de transmisión. Los principales dispositivos de esta tecnología son actuadores, sensores, controladores, transceptores, repetidores y filtros.

Los módulos X-10 solo llevan dirección los de aplicación que son los únicos receptores de órdenes lo que supone una ventaja, pero presenta el inconveniente de que no se puede verificar que equipo envía una orden.



Imagen 16. X-10

3.9.2.5 LonWorks

Creado en la década de los 90 por la compañía Echelon, es un protocolo muy aceptado en la actualidad homologado en gran cantidad de países. Está también homologado por el estándar europeo de electrodomésticos CEDEC AIS y por la red de gasolineras (IFSF) para el control y la comunicación, siendo también de los más utilizados en el control de viviendas y edificios, así como para el control industrial, transporte ferroviario, naval y aeroespacial, e incluso para la monitorización remota de contadores y el Street-lighting. Hoy en día, es el más popular en domótica en los Estados Unidos y compite directamente con KNX por liderarlo en Europa.

Es utilizado por multitud de empresas para automatización y redes de control distribuidas existiendo millones de dispositivos de este protocolo, lo que hace que se

utilice, sobre todo, en grandes edificios, ya que para viviendas existen tecnologías más económicas.

LonWorks emplea un sistema descentralizado y de extremo a extremo, permitiendo distribuir la instalación entre todos los dispositivos tanto a nivel físico como a nivel de aplicación en redes de control. Es un sistema abierto donde cualquier fabricante puede aplicar y mejorar la tecnología desarrollando productos nuevos.

Es capaz de admitir cualquier medio físico ya que utiliza equipos que se conectan mediante transceptores, permitiendo adaptar la señal a un medio en concreto. Por ello, existen multitud de canales con diferentes tasas de transmisión de bits, un determinado número de equipos y distintas distancias de alcance.

LonWorks permite el uso de todas las topologías de red disponibles siendo las más comunes de usar las de bus, estrella y anillo. La tipología bus se utiliza principalmente para redes industriales gracias a que se puede ampliar el sistema fácilmente añadiendo nuevos dispositivos. En cambio, la tipología de estrella resulta más adecuada para espacios más reducidos.

Los dispositivos LonWorks están formados principalmente por dos componentes:

- **Neuron Chip.** Creado por la propia marca, está formado por tres microprocesadores; dos se encargan de ejecutar el protocolo de comunicaciones y otro maneja las aplicaciones, dando la posibilidad de crear programas complejos sin que una ralentización pueda provocar problemas al resto de la red.
- **Transceptor.** Permite tanto enviar como recibir mensajes y su función es la de conectar el dispositivo con la red, convirtiendo las señales del Neuron Chip para que puedan ser adaptadas a lo que se necesite. Por ello, los transceptores pueden comunicarse entre sí si son iguales y mediante *routers* o puentes si no lo son.

Las principales ventajas que podemos destacar de este protocolo son:

- Tecnología abierta e interoperable.
- Inteligencia en cada punto.
- Independencia del medio de transmisión.
- Lenguaje optimizado y rápido.
- Robustez en la red.

- Seguridad de la comunicación y de los propios datos.
- Capacidad de la red.
- Flexibilidad de elección: ya que posee una amplia variedad de ofertas competitivas con una gran cantidad de fabricantes a nivel mundial.

Su principal desventaja es su mayor coste respecto al de su competencia y la necesidad del hardware específico y el Neuron Chip para el movimiento de la red de protocolo.



Imagen 17. LonWorks

3.9.2.6 KNX

Se trata de un estándar de protocolo de comunicaciones de red basado en OSI para edificios inteligentes. Procede de la convergencia de tres estándares previos:

- *European Home Systems Protocol (EHS)*
- *European Installation Bus (EIB o Instabus)*
- *BatiBUS*

Este estándar está gestionado por la *Asociación KNX*.

Hasta enero de 2016 para poder utilizar este protocolo era necesario adquirirlo mediante previo pago a una tarifa, pero, desde entonces, las especificaciones son de libre acceso siempre teniendo una cuenta gratuita registrada en la web de *KNX Association*.

Los principales objetivos de esta fusión fueron:

- Crear un estándar único que pueda cubrir todas las necesidades de las instalaciones tanto domóticas como inmóticas ya sea para uso profesional como residencial dentro del ámbito europeo.
- Mejorar las prestaciones de los dispositivos utilizados para la comunicación especialmente los utilizados por radiofrecuencia.

- Abarcar un mayor número de modos de funcionamiento.
- Utilización de buses domóticos en climatización.
- Fomentar la gestión técnica de los dispositivos mediante el contacto con proveedores autorizados.

El protocolo KNX define varios medios de comunicación física:

- **Cableado de par trenzado.** El cual fue heredado de los protocolos *BatiBUS* y *EIB*)
- **Red eléctrica.** Heredado de *EIB* y *EHS* y muy similar al que encontramos en el protocolo X-10.
- **Radiofrecuencia.** (*KNX-RF*)
- **Ethernet.**

El cableado se realiza sobre par trenzado a 9600 bps donde, por los dos hilos se suministra 30 Vdc para la telealimentación de los dispositivos KNX utilizando la técnica CSMA/CA con arbitraje positivo del bus que evita colisiones y, en consecuencia, los reintentos, maximizando el ancho de banda disponible.

En la red eléctrica se utilizan corrientes portadoras sobre 230 Vac/50 Hz a 1200/2400 bps. Usa la modulación SFSK y la distancia máxima que se puede conseguir sin repetidor es de 600 metros. Además, este sistema no es compatible con sistemas de alimentación eléctrica a 110V/60 Hz.

Para la conexión Ethernet se utiliza generalmente como *backbone* entre segmentos, líneas y áreas KNX permitiendo la transferencia de datos a través del protocolo IP a ubicaciones remotas.

Por último, la comunicación por radiofrecuencia permite conseguir una conexión inalámbrica utilizando el aire como medio de transmisión, permitiendo comunicar distancias de hasta 300 metros en campo abierto. Es posible utilizar repetidores para mayores distancias o edificios con múltiples estancias, pero es recomendable, sobre todo, para comunicaciones de baja prioridad.

En cuanto a los dispositivos físicos utilizados por el sistema KNX podemos destacar 4 grupos principales:

- **Actuadores.** Son los elementos que conectan físicamente los elementos a controlar en el edificio como, por ejemplo, las luces, electroválvulas, motores, etc., y hacen la traducción de las instrucciones del protocolo KNX, regulando o accionando los dispositivos controlados.

- **Sensores.** Son los elementos del sistema encargados de recoger los datos o interpretar órdenes del usuario como, por ejemplo, pulsadores, botones, detector de movimiento, termostato, etc. Muchos sensores incorporan visualizadores donde se puede controlar y monitorizar el sistema, como las pantallas táctiles.
- **Pasarelas.** Son los conocidos como *Gateway* o *routers* y se encargan de enlazar otros sistemas con otros protocolos de comunicación con KNX, por ejemplo, de BACnet, LonWorks o X10. Permiten interactuar con proyectores, otros sistemas inteligentes o comunicarse de manera remota con el sistema.
- **Acopladores.** Realizan una separación física dentro del bus consiguiendo agrupar los dispositivos en un segmento de características según sus características, ubicación o funciones para tener una mayor eficiencia en el envío de datos, alcanzar mayores distancias.

Además, podemos distinguir dos tipos de software en este sistema:

- **Software de gestión.** Será el encargado de configurar los dispositivos y ponerlos en marcha. Actualmente es el ETS, el cual es un programa que funciona en el sistema operativo Windows y nos permite relacionar actuadores con sensores y traducir las comunicaciones a través de las pasarelas. Este software es el único que permite realizar estas acciones y su distribuidor es KNX Association.
- **Software de control.** Es el encargado de otorgarnos la posibilidad de interactuar con el control y la visualización del sistema mediante un dispositivo de cómputo. Con él podremos realizar distintas funciones como visualizar el estado de los elementos, controlar la instalación, registrar eventos, crear funciones lógicas, etc.

Entre sus principales ventajas podemos destacar que el estándar KNX permite contar con un sistema que elimina los problemas originados por la coexistencia de varios dispositivos independientes, lo que nos ayudará a elegir todos los componentes con independencia del fabricante y permitiendo la ampliación futura sin tener que acometer grandes obras.

Será posible tener una mayor facilidad para gestionar de manera unitaria los equipos logrando satisfacer los principios clave como la seguridad, el ahorro y la optimización energética.

Por último, soporta un gran número de modos de configuración y es independiente de cualquier clase de plataforma de software o hardware que podamos encontrar en el mercado.



Imagen 18. KNX

3.9.2.7 ARDUINO

Se trata de una compañía de desarrollo de software y hardware de código libre, así como una comunidad internacional que se dedica a diseñar y manufacturar placas de desarrollo de hardware para construir dispositivos digitales e interactivos que puedan detectar y controlar objetos de la vida real.

Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darle diversos tipos de uso.

Su característica principal es el hardware y software libre; esto quiere decir en cuanto al hardware que son dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. Esto significa que cualquier empresa puede desarrollar sus propias placas con la misma funcionalidad debido a la utilización de una base común.

El software libre son programas informáticos creados con código accesible por cualquiera con la particularidad que cualquiera puede utilizarlo y modificarlo. Cuenta con una plataforma llamada Arduino IDE (Entorno de Desarrollo Integrado), el cual es un entorno de programación en el que cualquiera puede crear aplicaciones para las placas.

Se creó en el año 2003 por estudiantes del Instituto de Diseño Interactivo de Ivrea (Italia) para facilitar el acceso y uso de la electrónica y la programación de manera más económica debido a que lo existente por aquel entonces era muy poco económico.

El resultado fue una placa que cuenta con todos los elementos necesarios para conectar periféricos a entradas y salidas de un microcontrolador y que permite la programación en los distintos sistemas operativos actuales, ya sea Windows, macOS y GNU/Linux.

Su funcionamiento se basa en una placa con un microcontrolador ATMEL. Los microcontroladores son circuitos integrados programables, es decir, que se pueden grabar instrucciones mediante un lenguaje de programación utilizando el entorno Arduino IDE. Con estas instrucciones se pueden crear programas para poder interactuar con los circuitos de la placa.

Este microcontrolador tiene una interfaz de entrada que permite la conexión de diversos tipos de periféricos, cuya información recibida por estos es trasladada al microcontrolador que se encarga de procesar los datos que recibe.

Existen gran variedad de periféricos con multitud de usos como sensores de proximidad, ultrasonidos, infrarrojos, radiofrecuencia, cámaras, etc. Además, posee una interfaz de salida que permite llevar la información procesada por la placa a otros periféricos como relés, pantallas, altavoces, etc.

Existen gran variedad de placas Arduino con características diferentes, pero con la misma base programable. Esto quiere decir que encontraremos en el mercado placas con diferentes números de entradas y salidas o con módulos ethernet y WiFi para poder interactuar con ellas de manera inalámbrica, entre otros. Esto permite la existencia de placas con distintos tamaños y especificaciones, lo que permite la elección de una placa u otra según las características que se necesiten utilizar.

Entre las distintas placas Arduino, resumimos en la siguiente tabla algunas de sus características diferenciales.

	UNO	MEGA 2560	LEONARDO	DUE	ADK	NANO	PRO MINI	ESPLORA
Microcontrolador	ATmega328	ATmega2560	ATmega32u4	AT91SAM3X8E	ATmega2560	ATmega168 (versão 2.x) ou ATmega328 (versão 3.x)	ATmega168	ATmega32u4
Portas digitais	14	54	20	54	54	14	14	-
Portas PWM	6	15	7	12	15	6	6	-
Portas analógicas	6	16	12	12	16	8	8	-
Memória	32K (0,5K usado pelo bootloader)	256K (8K usado pelo bootloader)	32K (4K usado pelo bootloader)	512K disponível para aplicações	256K (8K usado pelo bootloader)	16K (ATmega168) ou 32K (ATmega328) (bootloader: 2K)	16K (2K usado pelo bootloader)	32K (4K usado pelo bootloader)
Clock	16Mhz	16Mhz	16Mhz	84Mhz	16Mhz	16Mhz	8Mhz (modelo 3.3v) ou 16Mhz (modelo 5v)	16Mhz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Serial/Módulo USB externo	Micro USB
Conector para alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não	Não
Tensão de operação	5V	5V	5V	3.3V	5V	5V	3.3 ou 5V, dependendo do modelo	5V
Corrente máxima portas E/S	40mA	40mA	40mA	130mA	40mA	40mA	40mA	-
Alimentação	7-12Vdc	7-12Vdc	7-12Vdc	7-12Vdc	7-12Vdc	7-12Vdc	3.3-12V (modelo 3.3v) ou 5-12V (modelo 5v)	5V

Tabla 1. Tipos de placas Arduino <https://www.filipeflop.com/>

Entre las ventajas que proporciona Arduino frente a otros sistemas destacan:

- Un bajo coste.
- Fácil de utilizar.
- Plataforma abierta.
- Alta flexibilidad.
- Gran variedad de placas.
- Hardware y software ampliable y de código abierto.
- Gran variedad de aplicaciones y usos.



Imagen 19. Arduino

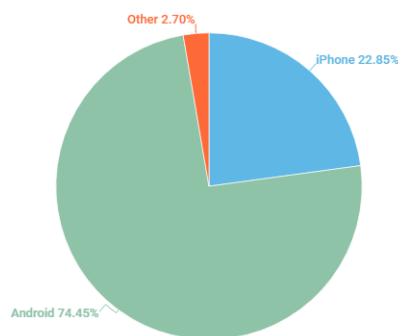
3.9.2.8 ANDROID

Android es un sistema operativo de código abierto basado en Linux presente en multitud de teléfonos inteligentes, tabletas, televisores, relojes inteligentes y automóviles. Fue creado por la empresa Android Inc., que años más tarde sería comprada por Google. Su principal ventaja es que al ser de código libre cualquiera puede tener acceso a la programación de aplicaciones para los dispositivos en los que se encuentra instalado, lo que produce una gran versatilidad en el desarrollo de software para dispositivos móviles.

En el desarrollo de software para el control domótico resulta una alternativa muy eficiente debido a la gran cantidad dispositivos que utilizan Android. Esto facilita el llevar el control domótico siempre encima de nosotros gracias a los teléfonos inteligentes, lo cual nos permite interactuar de manera remota con el sistema, dando la posibilidad de múltiples opciones sin necesidad de tener que ser únicamente controlado desde el mismo sitio donde se encuentre instalado el sistema domótico.

Existe gran variedad de software para la programación en Android como, por ejemplo, App Inventor, un sencillo entorno de programación web que permite desarrollar aplicaciones para este sistema sin necesidad de tener un programa instalado, ya que se accede mediante su página web.

Su principal ventaja es que al ser de código libre cualquiera puede desarrollar aplicaciones de manera gratuita. Además, su gran popularidad hace que cualquiera disponga de algún dispositivo con este sistema operativo.



Figures covering Jan 2018 - Jan 2019 supplied by Statcounter

Imagen 20. Gráfico porcentaje de la utilización de Android



Información de este apartado recopilada e imágenes extraídas de la web:

<https://domoticasistemas.com/>

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

<https://micasainteligente.site/>

<https://es.wikipedia.org/>

4 SOLUCIÓN ADOPTADA

Tras analizar los distintos tipos de sistemas domóticos que encontramos en la actualidad y viendo las ventajas y desventajas que estos ofrecen, el proyecto lo realizaremos utilizando Arduino debido a su bajo coste y a la facilidad que ofrece a la hora de acceder a diferentes periféricos compatibles. Se trata de un sistema de código libre y existen multitud de placas programables que elegiremos según las necesidades de nuestra vivienda.

Además, completaremos el proyecto con el desarrollo de una sencilla aplicación basada en Android que nos permitirá el control del sistema domótico mediante cualquier dispositivo que soporte este sistema operativo, en nuestro caso un teléfono inteligente.

5 LA VIVIENDA

La vivienda se encuentra en la Calle Pintor Renau 42, 10 en Torrent, Valencia y se trata de una vivienda construida en el año 1991. Cuenta con 103 m² construidos y tras una reforma actualmente cuenta con un recibidor, una cocina, una galería, un salón comedor, tres habitaciones, dos baños, pasillo y un armario/trastero.

5.1 ESTANCIAS

A continuación, detallaremos en una tabla las dimensiones de estos.

ESTANCIA	SUPERFICIE (m ²)
Recibidor	4
Cocina	10
Galería	2
Salón	25
Habitación 1	10
Habitación 2	10
Habitación 3	15
Baño 1	5
Baño 2	8
Armario/trastero	5
Pasillo	4

Tabla 2. Estancias de la vivienda

5.2 TAREAS QUE REALIZAR

El objetivo del proyecto es la domotización de la vivienda mediante el sistema Arduino como hemos estudiado previamente, por ello, en primer lugar, detallaremos los procesos a domotizar que se van a llevar a cabo.

1. Control automático de luces.
2. Detección de gas y humo.
3. Control de temperatura y climatización.
4. Control de seguridad física.
5. Control de persianas.

Por ello, debemos realizar un estudio sobre los componentes que van a ser utilizados minimizando el coste por lo que se utilizará como base la instalación eléctrica ya

existente, modificando únicamente lo estrictamente necesario para dotar la vivienda de inteligencia.

5.3 ANÁLISIS DE LA VIVIENDA

Realizaremos un estudio de los elementos a domotizar de las diferentes estancias de la vivienda.

5.3.1 ENTRADA

En la entrada encontramos un interruptor para activar las luces del techo. En esta estancia automatizaremos dicho interruptor para que pueda ser accionado mediante la colocación de un sensor de movimiento y un sensor de luminosidad para el encendido automático de éstas.

También se colocará un sensor de proximidad en la puerta que se pondrá en funcionamiento cuando el usuario desee mediante la aplicación móvil para la detección de intrusos emitiendo una señal acústica si se abre la puerta.

5.3.2 COCINA

En la cocina encontramos un interruptor para la luz, que automatizaremos para que pueda ser controlado desde la aplicación móvil. Además, instalaremos un detector de humo para que envíe una notificación si se produce algún incendio.

5.3.3 GALERÍA

En la galería encontramos únicamente un interruptor de luz que automatizaremos para que pueda ser controlado mediante la aplicación móvil.

5.3.4 COMEDOR

El comedor cuenta con dos interruptores de luz que automatizaremos para ser controlado desde la aplicación móvil. También encontramos dos persianas, a las cuales se va a instalar un motor para poder controlarlas mediante interruptores de manera física, así como de manera remota.

También se encuentra un aire acondicionado el cual automatizaremos para ser controlado remotamente, tanto su encendido y apagado como la variación de temperatura.

5.3.5 ARMARIO TRASTERO

Se automatizará el interruptor de luz mediante un sensor de movimiento para que se active siempre que se acceda a él.

5.3.6 BAÑOS

Tenemos dos baños con dos interruptores de luz, uno en cada baño, que serán automatizados para controlarlos remotamente.

5.3.7 PASILLOS

Tenemos un pasillo dividido en dos en forma de L, con dos puntos de luz independientes que serán automatizados y, a su vez, se instalarán dos sensores de movimiento para su encendido automático.

5.3.8 HABITACIONES

Tenemos tres habitaciones que cuentan con tres interruptores de luz, uno en cada habitación, que serán automatizados para controlarlos remotamente. Además, cuentan con tres persianas que serán automatizadas con la elección de un motor y para los cuales se hará una instalación física para su control, así como para poder accionarlas remotamente.

5.4 ELECCIÓN DE COMPONENTES

5.4.1 MICROCONTROLADOR

Tras analizar previamente en una tabla (*tabla 1*) las diferentes placas Arduino que encontramos en el mercado la elección es la Arduino MEGA 2560. Los microcontroladores Arduino tiene la particularidad que sus pines pueden utilizarse indistintamente como entradas o salidas, por ello, la elección de este en concreto se debe a que cuenta con un gran número de pines, así como un puerto de comunicación serie. También soporta comunicación I²C.

Tener el mayor número de pines posibles facilita la ampliación del sistema domótico con la inclusión de nuevos periféricos sin necesidad de tener que adquirir una nueva placa, lo cual incrementaría el coste.



Imagen 21. Arduino MEGA 2560

Esta placa no cuenta con un módulo WiFi incorporado, pero lo solucionaremos con un periférico ESP8266 ESP01 el cual resulta muy económico y cumple con las necesidades para dotar a nuestro sistema domótico de conexión inalámbrica. No contar con una placa que incorpore directamente un módulo de conexión red presenta la ventaja que, ante cualquier fallo de algunos componentes, estos se puedan sustituir de manera independiente reduciendo el coste de la reparación.



Imagen 22. Módulo WiFi ESP8266

5.4.2 CONTROL AUTOMÁTICO DE LUCES

Para el control de las luces se utilizará el módulo de relé simple que permite controlar un dispositivo de corriente alterna hasta de 10 A a 250 Vac y 30 Vdc. Resulta muy útil este tipo de relé ya que cuenta con las conexiones ya instaladas para la corriente alterna además de ser un producto muy económico. En el mercado se encuentran módulos de hasta 16 relés en una misma placa lo que aumenta las posibilidades de conexión y reduce los costes.

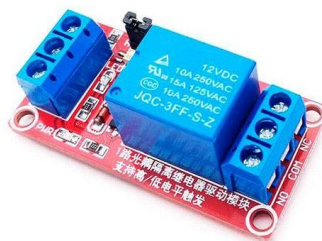


Imagen 23. Módulo de relé

Se utilizará también un sensor de movimiento PIR HC-SE501 el cual funciona con un rango de tensión entre 5V-20V y cuenta con unas dimensiones reducidas, 3,2 x 2,4 cm. Además de contar con un coste muy bajo, sus dimensiones permiten colocarlo en cualquier sitio sin ocupar apenas espacio y reduciendo el impacto visual.



Imagen 24. Sensor de movimiento

Se incluirá un detector de luminosidad o fotorresistencia LDR GL5516 que será utilizado juntamente con el sensor de movimiento antes mencionado, con el objetivo de encender luces únicamente cuando el nivel de luz recibida sea bajo y se detecte movimiento. Sus dimensiones son reducidas, 2 x 4 x 5 mm, admite una tensión máxima de 150 V y una disipación de potencia máxima de 100 MW.



Imagen 25. LDR

Se cambiarán las bombillas de toda la vivienda utilizando halógenos GU10 Led de 4 W para las zonas donde se utilizan este tipo de bombillas, de 350 lúmenes sin parpadeos ni deslumbramiento. Este tipo de halógenos equivalen a uno convencional de 50W permitiendo un ahorro energético de más del 90%.

GU10 LED BULBS



Imagen 26. Halógeno GU10 LED

Se utilizarán bombillas de rosca para el resto de las estancias que no llevan halógenos como la Bombilla LED Esférica E27, de 14 W de color blanco cálido, que proporciona

1521 lúmenes de luz de manera inmediata siendo el equivalente a una bombilla convencional de 100W, produciendo un ahorro de más del 86% de energía.



Imagen 27. Bombilla LED Esférica E27

5.4.3 DETECTOR DE GAS Y HUMO

Se empleará el módulo MQ-2 para la detección de gases y humo el cuál puede detectar GLP, Butano, Propano, Metano, Alcohol, Hidrógeno y humo. Funciona con un voltaje de 5 VDC, con una salida analógica de 0 a 5 V y salida digital TTL. Es de tamaño reducido y muy económico.



Imagen 28. Detector de gas y humo MQ-2

5.4.4 CONTROL DE TEMPERATURA Y CLIMATIZACIÓN

Se controlará el aire acondicionado mediante la clonación del mando a distancia para poder controlarlo de manera remota desde Arduino. Se utilizará un módulo de emisión infrarroja IR que funciona a 5 V y tiene una longitud de onda de 940 nm, siendo sus dimensiones reducidas de 13.7 x 27.8 mm.

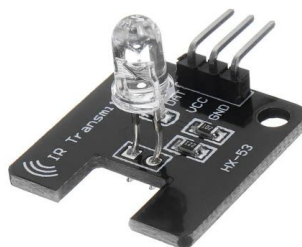


Imagen 29. Emisor IR

5.4.5 CONTROL DE SEGURIDAD

Se controlará la entrada de intrusos a la vivienda mediante un sensor de proximidad de ultrasonidos HC-SR04, con un voltaje de funcionamiento de 5 VDC, una distancia de detección de 2 a 450 cm y una precisión de hasta 3 mm.

Además, se utilizará un zumbador activo que emita un sonido constante cuando detecte presencia, RS Pro, con un nivel de sonido de 96 dB, frecuencia resonante de 2.8 kHz y tono continuo.



Imagen 30. Sensor de proximidad



Imagen 31. Zumbador

5.4.6 CONTROL DE PERSIANAS

Para el control de persianas se instalarán motores tubulares dentro del eje de las persianas de modo que permitan la subida y bajada de las mismas de manera mecánica. Se emplearán motores estándar de 230 V, utilizando finales de carrera para controlar si la persiana está abajo o arriba del todo.

El accionado se utilizará mediante la actuación de módulos relé mencionados anteriormente que serán controlados mediante Arduino.



Imagen 32. Motor tubular persiana

Imágenes extraídas de <https://www.google.es/imghp?hl=es&tab=wi&oqbl>

6 CONTROL DOMÓTICO DE LA VIVIENDA

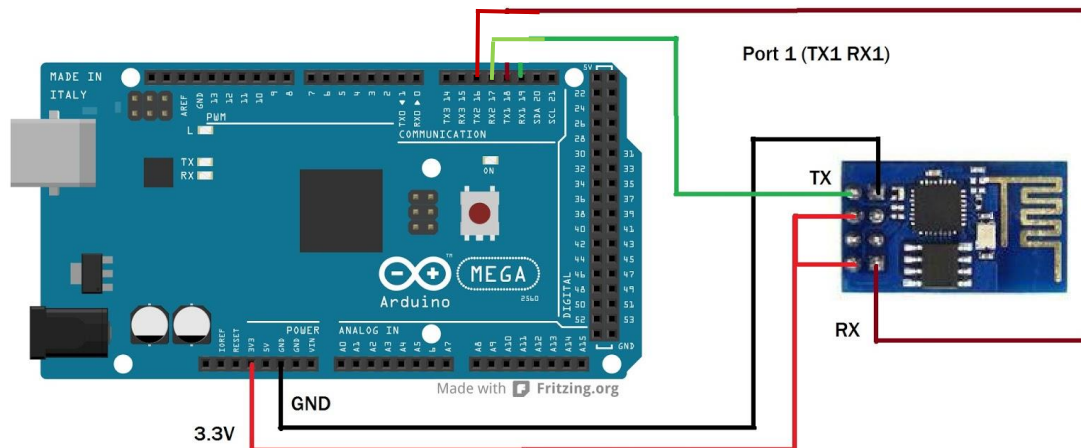
En este apartado describiremos detalladamente cómo se va a realizar cada proceso de domotización por secciones, cómo se realizan las conexiones y qué parámetros son necesarios para la programación del código en Arduino, que serán añadidos al final del proyecto en el *Anexo*.

6.1 CONEXIÓN ENTRE ARDUINO Y MÓDULO WIFI

El módulo wifi ESP8266 se comunica con el Arduino mediante puerto serie utilizando los pines RX y RT de la placa. Cabe mencionar que para otra placa de Arduino la conexión a los pines puede variar. En nuestro caso, dado que hemos utilizado la placa Arduino Mega 2560 la conexión se realiza en los pines 18 TX2 y 19 TX2 y de manera intercambiada, es decir, el RX del Arduino con el TX del módulo ESP8266 y el RT con el RX.

El resto de los pines del módulo, pin CH_PD apaga o enciende el módulo, por tanto, debe conectarse a la tensión de 3.3 V, así como también el módulo Vcc. Por último, el GND se conecta a un GND de la placa.

Es importante recalcar que el módulo funciona a 3.3 V y que conectarlo por error a 5 V lo dejaría inutilizable.



Conexión Arduino Mega 2560 + ESP8266

Imagen 33. Conexión módulo ESP8266. Fuente: pdacontroles.com

Como hemos mencionado, el módulo se comunica por puerto serie y funciona mediante comandos AT. Por ello, para realizar la comunicación debemos declarar las variables que conforman el puerto serie del Arduino.

```
void setup ()  
{  
Serial.begin(115200)  
Serial2.begin(115200)  
}
```

La velocidad de transmisión de datos la hemos fijado en 115200 ya que el fabricante determina que pueden ser 9600 o 115200 y, en nuestro caso, funciona con la segunda.

Los comandos AT son los que mandan las funciones al módulo. Mediante ellos debemos establecer la conexión entre el módulo y nuestra red wifi. Para ello se utiliza el comando AT+CJWAP = 'NOMBRE DE LA RED', 'CONTRASEÑA'. Cuando se realiza la conexión se le asigna automáticamente una dirección IP que será de vital importancia para poder mandar las instrucciones a ejecutar con Arduino, y también es importante asignar un puerto de comunicación, el cual se asigna mediante el comando AT+CIPSERVER=1, 80.

Así, ya queda establecida la conexión del módulo con la red wifi, mediante una IP y un puerto de conexión. Ejemplo: <http://192.168.1.141/80>.

Para enviar el accionamiento de pines del Arduino se deberá utilizar esta dirección asignada para la comunicación que explicaremos posteriormente mediante la utilización de una aplicación Android que nos permitirá controlar el sistema domótico desde un smartphone.

6.2 CONFIGURACIÓN DE LUCES

El control de las luces se realizará actuando de manera directa sobre los interruptores existentes en la casa de manera que puedan ser accionados de manera manual o remotamente.

Para ello conmutaremos el interruptor de pared con el módulo de relé de la siguiente manera.

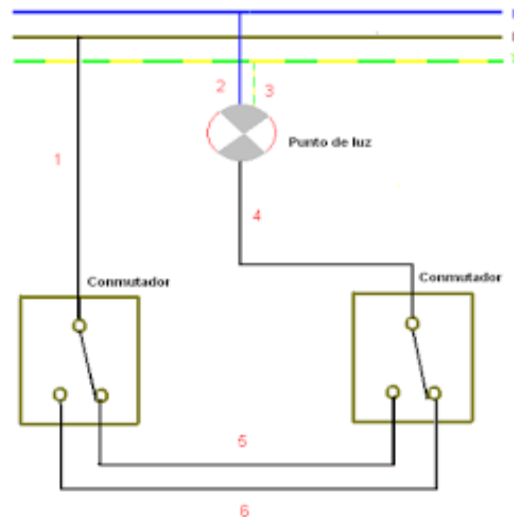


Imagen 34. Esquema de conexión de dos conmutadores.

Como se observa, se conecta la fase al común del interruptor y el neutro al común del relé, el contacto cerrado del interruptor con el contacto abierto del relé y el contacto abierto del interruptor con el contacto cerrado del relé.

De esta manera, si hubiera una pérdida de conexión a internet o algún problema con el sistema domótico la instalación seguiría funcionando de manera convencional.

La conexión del módulo de relé a la placa se hace conectando el pin Vcc a 5 V, el GND con un GND, el pin IN a una salida digital del Arduino.

Para el código, se inicializa un pin y se le da valor de salida (**OUTPUT**). Como hemos estudiado, la ventaja que presenta Arduino es que sus pines pueden actuar tanto de entrada como de salida. También se pondrá inicialmente en estado alto (**HIGH**).

Es importante destacar que el módulo de relé que vamos a utilizar en este proyecto funciona de manera inversa, es decir, se activa cuando está en estado bajo.

```
void setup()
{
pinMode(23, OUTPUT);
digitalWrite(23,HIGH);
}
```

Para recibir la acción enviada desde el smartphone se debe de enviar una serie de comandos AT que sirvan para recibir la orden. Así pues:

```
void loop()
{
```

```
if(Serial2.available()) // Verifica si el módulo está disponible
{
if(Serial2.find("+IPD,")) //Busca el campo de texto +IPD
{
delay(1000); //Espera un segundo para leer todos los datos del puerto serie
int connectionId = Serial2.read()-48;
Serial2.find("pin="); // avanza el cursor a "pin="
int pinNumber = (Serial2.read()-48)*10; // conseguir primer número, si el pin es 23,
entonces el primer número es 1, entonces se multiplican para obtener 10
pinNumber += (Serial2.read()-48); // consigue el segundo número, si el número de
pin es 13, entonces el segundo número es 3, a continuación, le añade al primero
digitalWrite(pinNumber, !digitalRead(pinNumber));

}}}
```

Mediante la aplicación del smartphone, presionamos un botón que configuramos para que envíe a nuestra dirección IP el texto "pin=23", Arduino capta el número y mediante el código **digitalWrite** asigna el pin 23 y con el código **digitalRead** cambia el estado de dicho pin. De esta forma, cuando recibe una orden dependiendo de los botones que hayamos configurado sabe que para un determinado pin debe cambiar su estado, en este caso, de HIGH a LOW, y viceversa, dando como resultado el encendido o apagado de luces.

En la vivienda tenemos un total de 11 interruptores distribuidos a lo largo de la casa, por tanto, declararemos 11 pines de la misma manera que hemos realizado antes.

6.2.1 CONTROL LUZ ENTRADA

El encendido y apagado de la luz de la entrada se realizará de manera automática mediante la utilización del sensor de movimiento y el LDR. Sólo cuando la luminosidad sea baja y se detecte movimiento se encenderá la luz. Esto funcionará de manera independiente y no se realizará ningún control remoto de la misma debido a que la luz de la entrada sólo se necesitará cuando se accede o se sale de la vivienda.

Para su funcionamiento se conecta una pata del LDR a la entrada analógica 0 que será la encargada de recibir los datos de luminosidad, el pin de salida del sensor de movimiento se conecta al pin digital 4 declarado como entrada y el pin de entrada del relé se conecta al pin digital 33 declarado como salida.

El código lo vemos en el ANEXO 1.

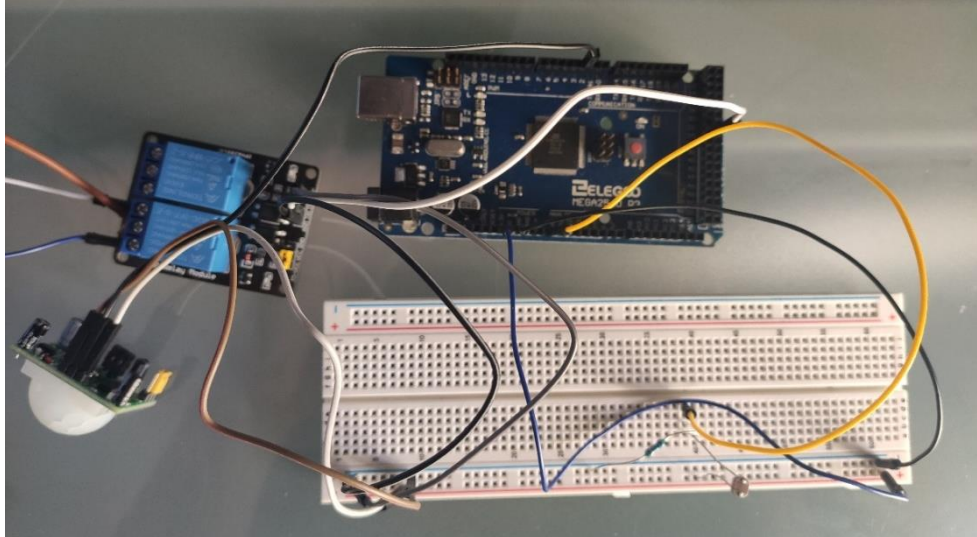


Imagen 35. Esquema conexión luz entrada

6.2.2 CONTROL LUZ ARMARIO TRASTERO

La vivienda cuenta con un armario trastero al cual dotaremos de luz con un sensor de movimiento y un relé que actúe sobre el interruptor que acciona la luz de manera manual. Utilizaremos el mismo sistema empleado en la luz de la entrada a excepción de eliminar el detector LDR para la luminosidad, ya que al tratarse de un armario es necesario encender siempre la luz.

Por tanto, el pin del sensor de movimiento irá conectado al pin 5 del Arduino que será declarado como entrada (*INPUT*) y el relé al pin 47 declarado como salida (*OUTPUT*). El código lo vemos en el *ANEXO*.

6.3 CONTROL DE SEGURIDAD FÍSICA

Se instalará una alarma anti-intrusos cuya activación o desactivación se realizará mediante la aplicación móvil. Esto permitirá su utilización únicamente cuando se desee, por ejemplo, cuando la vivienda esté vacía.

Se realizará mediante un sensor de proximidad ubicado en el techo de la entrada a pocos centímetros de la puerta, para cuando la puerta se abra detecte la proximidad de esta y accione un zumbador que emitirá un pitido y, al mismo tiempo, encienda todas las luces de la vivienda.

Para su conexión, en primer lugar, se conectan los pines del sensor de proximidad, VCC a 5 V, GND a masa y TRIG y ECHO a los pines 3 y 2, que serán definidos en el

código como salida y entrada, respectivamente.

El zumbador se conecta el negativo a masa y el positivo al pin digital 48, el cual se configura como salida.

El esquema quedaría montado de la siguiente manera.

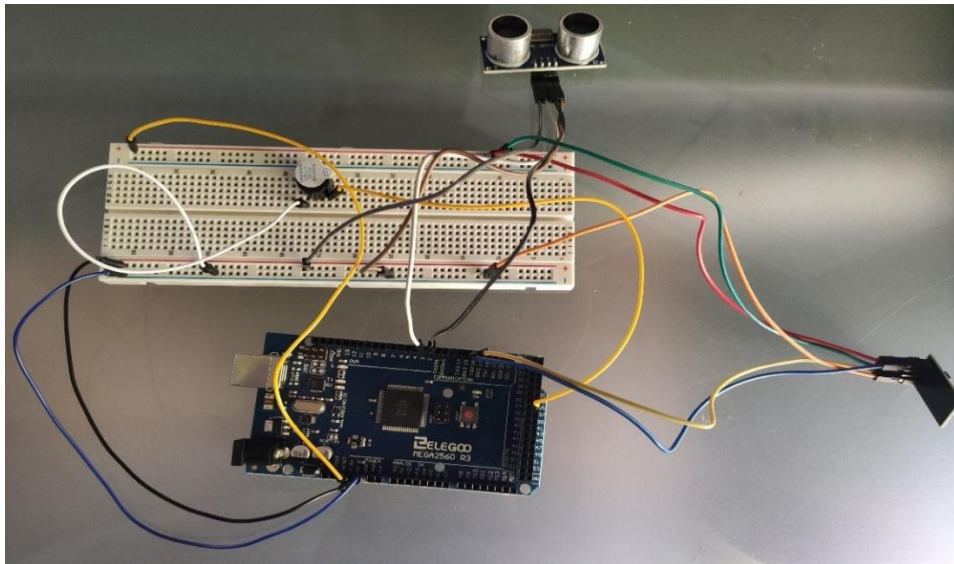


Imagen 36. Esquema conexión alarma

Para el código de programación es importante la configuración del sensor de proximidad. Al tratarse de un sensor de ultrasonidos debemos hacer los cálculos correspondientes para medir la distancia a la que configuraremos su funcionamiento.
ANEXO 2.

Destacar la función ***pulseIn()*** que se encargará de escuchar el pin que le pasemos y buscará una señal que pase de LOW a HIGH, contando el tiempo que tarda en volver a bajar desde que sube.

6.4 DETECCIÓN DE GAS Y HUMO

Para la detección de gas y humo utilizaremos un sistema independiente de la aplicación móvil que permita la detección constante de gases y humos, avisando en caso de que se detecte con una señal acústica.

El detector estará situado en la cocina debido a que es el lugar de la vivienda donde se puede producir una fuga de gas o es más probable que se produzca un incendio debido a la naturaleza de los electrodomésticos.

La conexión con Arduino será de manera similar a la ya mencionada para el control de la luz de entrada de la vivienda.

Es importante destacar que el sensor de gas tiene la posibilidad de conectarse de manera analógica o digital. Por una parte, la entrada analógica nos permitiría trabajar con diferentes niveles de gas y escalarlo de acuerdo con la necesidad de nuestra aplicación mientras que la entrada digital actuaría de manera más rápida y sencilla, puesto que no sería necesario la parametrización de los distintos niveles de gases que puede detectar este sensor.

En nuestro caso, utilizaremos la entrada digital para leer los datos del sensor y ajustaremos la sensibilidad de este a través de la resistencia variable que incorpora; girando a la derecha se hace más sensible y necesitamos menos presencia de gas para activar la salida y, de igual forma, girando hacia la izquierda el umbral de detección aumenta siendo necesaria más cantidad de gas para su activación.

Debido a la variedad de gases que es capaz de detectar este sensor y ya que su función va a ser únicamente el aviso acústico para el usuario no es necesario parametrizar de manera exhaustiva los parámetros. Aun así, siempre queda la posibilidad de realizar alguna modificación futura si el usuario lo necesitara.

Por ello, la conexión se realizará conectando la alimentación a 5VCC y tierra como en los otros sensores estudiados en este proyecto y la entrada digital al pin 48, el cual se declarará como *INPUT*.

Se conectará un zumbador a la salida digital 49 el cual se declarará como *OUTPUT*, para que en el momento se detecte un gas se accione y emita el sonido.

El detector funciona de tal manera que cuando está en estado alto, es decir, estado lógico 1, no está detectando ningún gas y cuando pasa a estado lógico 0 es cuando se encuentra detectando.

6.5 CONFIGURACIÓN E INSTALACIÓN DE PERSIANAS

El control de persianas se realizará primero automatizando éstas con la instalación de un motor y dos interruptores que permitan la subida y la bajada.

El motor de la persiana se conecta a los interruptores siguiendo el siguiente esquema:

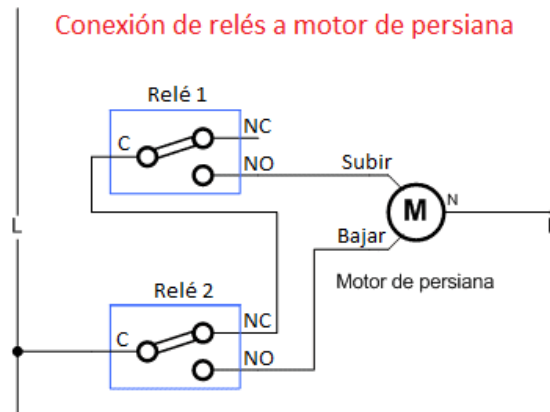


Imagen 37. Esquema conexión motor-persiana

Donde realizamos la conexión en serie entre ambos interruptores conectando los cables de subir y bajar persiana a los contactos normalmente abiertos de cada interruptor y se conecta el contacto normalmente cerrado del interruptor 2 al común del interruptor 1. Finalmente, la línea se conecta al común del relé 2 y el neutro directamente al neutro del motor.

Es importante el ajuste de los finales de carrera que vienen determinados en este tipo de módulos de motorización de persiana. El final de carrera tiene como función que el motor se detenga cuando llega a un determinado punto, evitando que siga girando constantemente de manera ininterrumpida.

Su ajuste se realiza de manera manual mediante el giro de dos tornillos que vienen instalados en los módulos, tanto para la subida como la bajada.

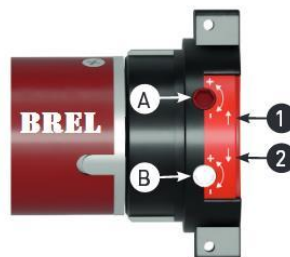


Imagen 38. Ajuste final de carrera

El proceso de control remoto de las persianas mediante Arduino y la aplicación móvil se realizará de manera similar a la ya estudiada con el control de luces.

De esta forma, el proceso consistirá en la instalación de dos módulos de relé conectados de forma paralela a los interruptores de subida y bajada de las persianas y que, a su vez, estará conectados al Arduino. Así, se consigue actuar de forma directa sobre los interruptores evitando tener que parametrizar otros aspectos y, además, permitiendo la utilización de ambos sistemas sin tener que prescindir de ninguno. Si el sistema domótico con Arduino sufriera algún fallo nos permitirá seguir haciendo uso del sistema de persianas de forma manual con los interruptores de pared.

Para la conexión se realizará el siguiente esquema:

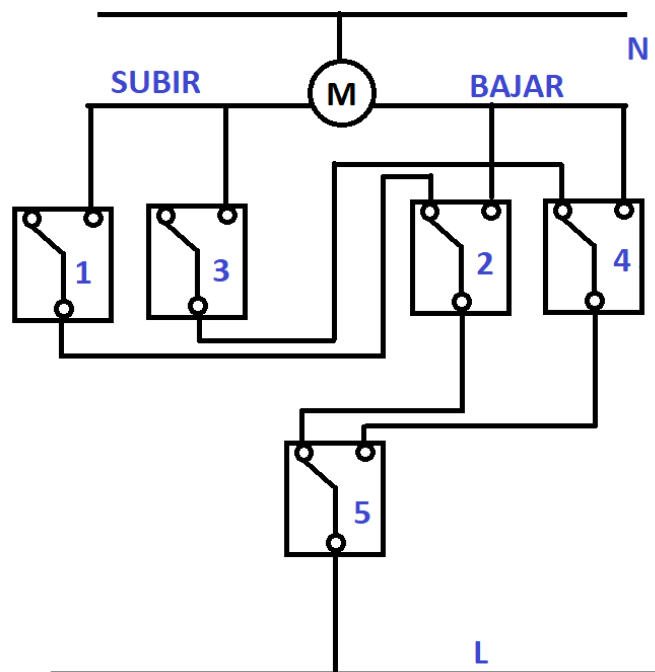


Imagen 39. Esquema conexión persiana Arduino

Como se observa, se han colocado 2 relés conectados de manera paralela a los interruptores anteriormente descritos. De esta forma, el relé 1 y 2 son los interruptores de pared y los relés 3 y 4 son los interruptores accionados desde Arduino. Esta conexión nos permite controlarlos de forma manual o de forma remota.

El relé 5 se ha instalado para decidir cuáles son los que funcionaran debido a que, si tenemos accionado el interruptor de pared y, al mismo tiempo, accionamos el interruptor remoto, provocaría una sobrecarga de tensión en el motor de la persiana el cual quedaría inutilizable.

La solución ante este problema que hemos optado es incluir el relé 5 que será accionado de manera remota desde la aplicación móvil y que actuará como selector, de manera que cuando se accione permitirá la subida y bajada de la persiana desde la aplicación y mientras no lo esté, sólo se podrá mediante los interruptores de pared.

El código de Arduino para el control de las persianas será el mismo utilizado para el control de luces, donde utilizaremos los pines correspondientes definiéndolos como *OUTPUTs*, uno para la subida y otro para la bajada de la persiana, y otro que servirá como selector para activar la opción de accionarlos de manera remota.

6.6 CONFIGURACIÓN DE CLIMATIZACIÓN

El control de climatización se va a realizar actuando sobre el aire acondicionado ya instalado en la vivienda mediante la clonación del mando a distancia de este. Esto nos permitirá ubicar un emisor IR próximo al dispositivo y mediante Arduino y la aplicación móvil controlar su encendido y apagado, así como subir y bajar la temperatura.

Se utilizará un receptor IR y un código previo el cual permita descifrar los códigos infrarrojos que emite el mando del aire acondicionado para, posteriormente, poder emplearlos en el código de programación de nuestro sistema domótico en Arduino.
Anexo 3.

Una vez descifrado el código del mando a distancia del aire acondicionado ya podemos realizar la implementación del código Arduino de nuestro sistema domótico.

La conexión del emisor de IR se realiza conectando la alimentación y masa, como en los demás sensores empleados en este proyecto, y la patilla *out* al pin 9 del Arduino.

Para la elaboración del código es importante declarar las variables que accionaran los comandos que enviará a Arduino nuestra aplicación móvil y que, posteriormente, ejecutará enviando la orden al emisor IR.

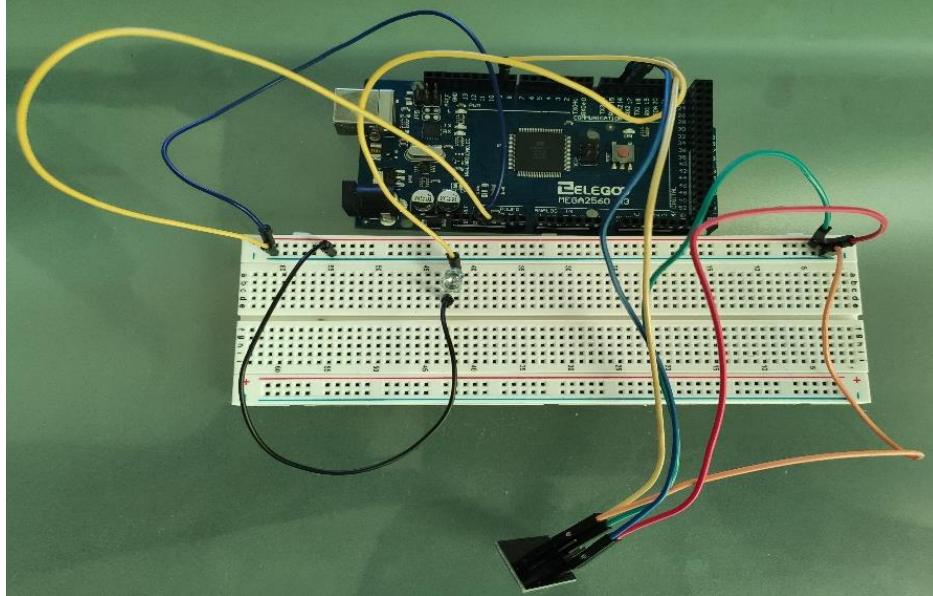


Imagen 40. Esquema conexión emisor IR

La solución adoptada para resolver el control de climatización no es muy habitual, pero resulta muy práctica y el impacto económico sobre el coste de la domotización es mínimo.

7 APLICACIÓN MÓVIL ANDROID

El sistema de domotización realizado en este proyecto está diseñado para ser controlado vía wifi mediante una aplicación móvil.

La aplicación la realizamos mediante el uso de App Inventor.

7.1 APP INVENTOR

App Inventor es un entorno de desarrollo por software creado por Google Labs que permite el desarrollo de aplicaciones para el sistema operativo móvil Android.

La particularidad que tiene este entorno de desarrollo es su interfaz gráfica, ya que permite el desarrollo de aplicaciones mediante el uso de bloques que se van enlazando entre sí. Esto simplifica su uso además que permite desarrollar aplicaciones teniendo simplemente nociones básicas de programación.

Se trata de una plataforma gratuita y accesible vía web, lo cual permite la creación de aplicaciones sin la necesidad de instalar ningún programa en nuestro ordenador. De esta forma, es posible acceder a nuestros proyectos desde cualquier plataforma y desde cualquier navegador web.

Nuestro proyecto precisa de una sencilla aplicación que permita el envío de comandos para accionar los diversos pines de la placa Arduino Mega 2560 los cuales será enviados por WiFi gracias a la implementación del módulo ESP8266. Éste permitirá conectarse a la red de la vivienda y mediante una dirección IP y un puerto que estableceremos nosotros podremos hacer el intercambio de datos entre nuestra aplicación y Arduino.



Imagen 41. App Inventor. <https://www.wikipedia.org/>

7.2 RESUMEN PINES ARDUINO

Para el desarrollo de la aplicación móvil, primero es necesario tener bien definidos y recogidos los pines de Arduino sobre los cuales ésta va a tener que actuar. La siguiente tabla muestra el resumen de los pines utilizados.

REFERENCIA	PIN
Activar alarma	22
Luz pasillo 1	23
Luz pasillo 2	24
Luz comedor 1	25
Luz comedor 2	26
Emisor IR	27
Activar persianas comedor	28
Subir persiana comedor	29
Bajar persiana comedor	30
Luz cocina	31
Luz galería	32
Luz baño 1	33
Luz baño 2	34
Luz habitación 1	35
Activar persiana habitación 1	36
Subir persiana habitación 1	37
Bajar persiana habitación 1	38
Luz habitación 2	39
Activar persiana habitación 2	40
Subir persiana habitación 2	41
Bajar persiana habitación 2	42
Luz habitación 3	43
Activar persiana habitación 3	44
Subir persiana habitación 3	45
Bajar persiana habitación 3	46

Tabla 3. Resumen pines Arduino para aplicación móvil

7.3 INTERFAZ GRÁFICA

La interfaz gráfica de la aplicación cuenta con una apariencia sencilla pero funcional. Cuenta con una única pantalla principal que recoge todas las estancias de la vivienda que se pueden controlar de manera remota.

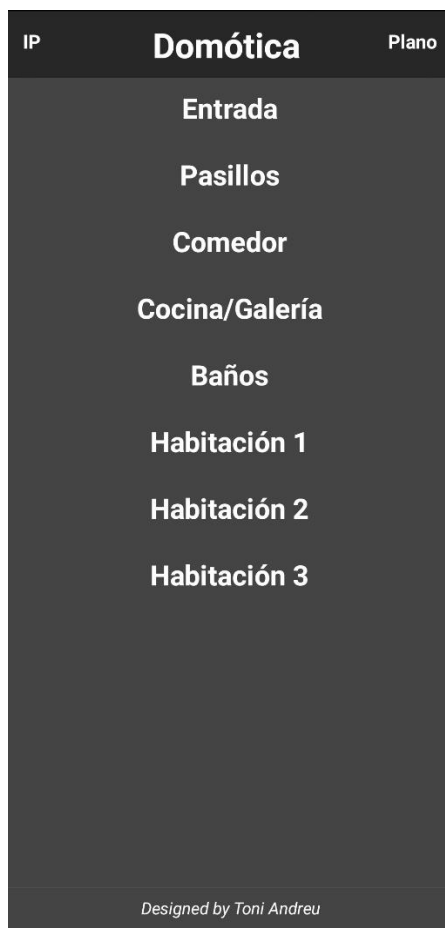


Imagen 42. App domótica pantalla principal

Como se observa en la imagen, queda de manera recogida para poder acceder rápidamente a la estancia que se desee controlar. Al presionar sobre una de ellas, se despliega un submenú con botones que permiten interactuar con los distintos dispositivos domotizados.

Así mismo, en la cabecera de la aplicación hay dos submenús adicionales que aparecen siempre que se desee, aunque estén todas las estancias habilitadas.

Por una parte, encontramos el botón IP el cual se ha añadido para poder modificar la dirección con la que la aplicación se comunica con el módulo wifi a través del *router*. De normal, una vez se ha hecho la primera conexión a red del módulo wifi esta dirección se mantiene, pero si se cambiara de *router* implicaría modificar el código de la aplicación

para poder comunicarlos entre ellos. De esta manera no sería necesario ya que simplemente deberíamos cambiar la dirección IP directamente desde este apartado.



Imagen 43. IP App domótica

El siguiente botón principal daría acceso a una visualización del plano de la vivienda mostrando donde se encuentran los elementos domotizados. De esta manera, resulta sencillo para el usuario poder conocer sobre qué elementos quiere actuar.

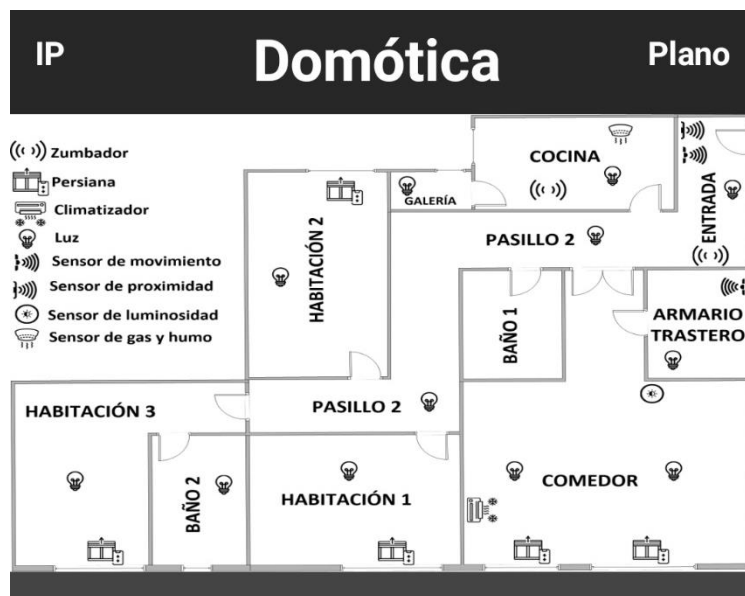


Imagen 44. Plano App domótica

Tanto el apartado de IP como del Plano, así como las distintas estancias de la casa se despliegan haciendo una pulsación sobre ellas y se recogen haciendo una pulsación más larga. Esto otorga fluidez y comodidad a la hora de interactuar con ella, pues si, por ejemplo, sólo se desea encender alguna luz en concreto o subir o bajar una persiana, no tenemos que buscar de manera confusa sobre toda la interfaz.



Imagen 45. App domótica desplegada

7.4 FUNCIÓN DE LOS BOTONES

Como se observa, el despliegue de las estancias muestra los elementos sobre los cuales se puede interactuar mediante botones o *switchs*.

Los botones y *switchs* sirven para cambiar de estado los pines del microcontrolador Arduino. De esta forma, cuando se activan o desactivan la aplicación envía el número del pin sobre el cual estamos actuando el cual hemos definido previamente y explicado el proceso en apartados anteriores. Así, podemos encender las luces, activar la alarma o encender el aire acondicionado.

La sección de climatización interactúa de manera directa sobre un emisor IR que transmite las ordenes como si fuera el propio mando a distancia del aire acondicionado. En este caso, cuando se enciende y se sube o baja la temperatura, la aplicación llama a la función enviando el nombre que le hemos definido previamente y el Arduino emite el código por el emisor IR, comportándose de la misma forma que si se tratase de un mando a distancia.

Por último, para poder actuar sobre las persianas primero debemos activar el *switch* de Activar Remoto y posteriormente presionar los botones subir o bajar. Esto se ha hecho para permitir también la interacción de manera física sobre las persianas mediante los interruptores de pared como hemos explicado anteriormente.

Hay que destacar que, como los botones de subir y bajar actúan sobre relés controlados, si se desea bajar o subir la persiana a una cierta altura sin tener necesariamente que llegar hasta el final, estos botones cambian su color a verde simulando que están presionados. En el momento se desee parar, simplemente hay que volver a presionar el botón que volverá a su estado inicial.



Imagen 46. Persianas App domótica

7.5 CÓDIGO FUENTE DE LA APLICACIÓN

Como hemos mencionado, la aplicación se ha desarrollado con la plataforma App Inventor. Su código sirve para enviar comandos desde la aplicación a la dirección IP asignada por el *router* al módulo wifi conectado a Arduino. El teléfono móvil debe estar conectado a la misma red que el módulo wifi.



Imagen 47. Código botones App domótica.



La función actúa sobre los botones o *switchs* de la siguiente forma:

Cuando se pulsa sobre el botón o cambia el estado del *switch* se ejecuta la función *llamar* la cual conecta con el visor web y manda a la URL un mensaje. La URL la coge de la dirección IP escrita en el campo de texto definido *TextBox1* donde al final de dicha dirección añade el mensaje *pin=22*.

De esta manera, escribe el mensaje <http://192.168.1.141/pin=22> en el servidor web que hace que el Arduino reciba el código *pin=22*, haciendo referencia en nuestro caso a poner en estado alto o bajo el pin 22 del Arduino como se ha visto en apartados anteriores.

Esto se repetiría para cada uno de los botones y *switchs* de la aplicación.

Se puede observar el código completo de la aplicación de domótica en el *ANEXO 4*.

8 PRESUPUESTO

A continuación, se plantea el presupuesto orientativo del desarrollo del proyecto. Incluye tanto la elaboración técnica del proyecto como la instalación de esta en la vivienda.

8.1 CUADRO DE PRECIOS ELEMENTALES

En este apartado incluimos el coste de los materiales necesarios para el desarrollo del proyecto.

CUADRO DE PRECIOS ELEMENTALES DE MATERIALES					
Ref.	Ud.	Descripción	Precio	Cantidad	Total
m1	ud	Arduino Mega 2560 R3	36,58 €	1	36,58 €
m2	ud	Módulo Wifi ESP8266	7,29 €	1	7,29 €
m3	ud	Modulo 8 relés 5V 220V	3,47 €	3	10,41 €
m4	ud	Sensor Movimiento PIR HC-SR501	2,20 €	2	4,40 €
m5	ud	Sensor LDR GL5516	0,11 €	1	0,11 €
m6	ud	Bombilla halógena GU10 LED pack 10 uds	36,99 €	2	73,98 €
m7	ud	Bombilla LED esférica E27 14W pack 2 uds	12,49 €	2	24,98 €
m8	ud	Módulo detector gas y humo MQ-2	4,99 €	1	4,99 €
m9	ud	Emisor infrarrojo IR KY-005	1,09 €	1	1,09 €
m10	ud	Sensor Proximidad HC-SR04	4,99 €	1	4,99 €
m11	ud	Zumbador activo RS-Pro	1,79 €	2	3,58 €
m12	ud	Interruptor conmutador subir/bajar persiana	2,35 €	4	9,40 €
m13	ud	Motor de persiana 40Nm	49,95 €	4	199,80 €
m14	ud	Resistencia 10kΩ	0,04 €	1	0,04 €
m15	ud	Fuente alimentación Arduino Mega 9V	3,95 €	1	3,95 €
m16	ud	placa multiperforada	0,25 €	3	0,75 €
		TOTAL			386,34 €

Tabla 4. Coste de materiales

CUADRO DE PRECIOS ELEMENTALES DE MANO DE OBRA			
Ref.	Ud.	Descripción	Precio
h1	h.	Ingeniero Electrónica	14,35 €
h2	h.	Especialista electrónica y sistemas informáticos	20,75 €
h3	h.	Oficial 1ª de electrónica	12,52 €

Tabla 5. Costes mano de obra

8.2 CUADRO DE PRECIOS DESCOMPUESTOS

En este apartado se descomponen los costes tanto de materiales como de mano de obra contabilizando las horas y medios auxiliares.

CUADRO DE PRECIOS DESCOMPUESTOS					
Ref.	Ud.	Descripción	Precio	Cantidad	Total
m1	ud	Arduino Mega 2560 R3	36,58 €	1	36,58 €
m2	ud	Módulo Wifi ESP8266	7,29 €	1	7,29 €
m3	ud	Modulo 8 relés 5V 220V	3,47 €	3	10,41 €
m4	ud	Sensor Movimiento PIR HC-SR501	2,20 €	2	4,40 €
m5	ud	Sensor LDR GL5516	0,11 €	1	0,11 €
m6	ud	Bombilla halógena GU10 LED pack 10 uds	36,99 €	2	73,98 €
m7	ud	Bombilla LED esférica E27 14W pack 2 uds	12,49 €	2	24,98 €
m8	ud	Módulo detector gas y humo MQ-2	4,99 €	1	4,99 €
m9	ud	Emisor infrarrojo IR KY-005	1,09 €	1	1,09 €
m10	ud	Sensor Proximidad HC-SR04	4,99 €	1	4,99 €
m11	ud	Zumbador activo RS-Pro	1,79 €	2	3,58 €
m12	ud	Interruptor conmutador subir/bajar persiana	2,35 €	4	9,40 €
m13	ud	Motor de persiana 40Nm	49,95 €	4	199,80 €
m14	ud	Resistencia 10kΩ	0,04 €	1	0,04 €
m15	ud	Fuente alimentación Arduino Mega 9V	3,95 €	1	3,95 €

M15	ud	Placa multiperforada	0,25 €	3	0,75 €
M.O.D					
h1	h.	Ingeniero Electrónica	14,35 €	240	3.444,00 €
h2	h.	Oficial 1ª de electrónica	12,52 €	56	701,12 €
h3	%	Medios auxiliares	5	%	226,54 €
Coste total sin impuestos					4.757,25 €

Tabla 6. Costes descompuestos

8.3 COSTE TOTAL CON IMPUESTOS

COSTE TOTAL		PRECIO
Coste total sin impuestos		4.757,25 €
Coste total con impuestos		5.756,27 €

Tabla 7. Coste total proyecto

El coste total del proyecto de domotización y control de una vivienda es:

**CINCO MIL SETECIENTOS CINCUENTA Y SEIS EUROS CON VEINTISIETE
CENTIMOS**

El elevado precio del proyecto pese a la utilización de materiales realmente económicos reside principalmente en las horas empleadas para su desarrollo completo. Si este proyecto se quisiera implementar en otras viviendas con similares condiciones el coste final se reduciría considerablemente.

9 CONCLUSIONES

Con la realización de este proyecto podemos comprobar la importancia que están adquiriendo en la actualidad las nuevas tecnologías ya que nos ayudan a tener mayor comodidad y seguridad en el hogar. La evolución del *Internet of Things* (IoT) y de la inteligencia artificial está en auge, no sólo por las comodidades físicas que aportan sino también porque son capaces de dotar de inteligencia cualquier ámbito donde se apliquen.

La domotización de viviendas se ha ido desarrollando desde hace muchos años. Además, gracias a su evolución, cada vez resulta más sencillo y económico poder aplicarla a las viviendas, no sólo a las nuevas construcciones, sino que también ofrece la posibilidad de dotar de automatización cualquier vivienda ya existente.

Ha resultado muy satisfactoria la realización de este proyecto porque hemos juntado tanto la parte técnica electrónica como la parte informática de programación, consiguiendo una fusión entre ambas otorgando como resultado la domotización real de una vivienda. Esto ha servido para conocer de primera mano los distintos protocolos de domotización y poder llevar a cabo uno en concreto: Arduino. Para su realización, hemos puesto en práctica gran cantidad de conocimientos adquiridos a lo largo del grado y al mismo tiempo hemos conseguido más experiencia en la materia.

10 BIBLIOGRAFÍA

Para el desarrollo de este proyecto se ha recopilado información de distintos sitios web, así como de algún libro.

WEBS:

Desarrollo estudio sistemas domóticos, protocolos, conexiones.

- <https://bibinq.us.es/>
- <https://www.iecor.com/>
- <https://clasificaciondelasredesblog.wordpress.com>
- <http://domotica1003.weebly.com>
- <https://domoticasistemas.com/>
- <http://www.serconint.com/>
- <https://micasainteligente.site/>
- <https://es.wikipedia.org/>
- <https://arduinoodhtics.weebly.com/>
- <https://www.knx.org/>

Desarrollo técnico

- <https://www.xataka.com/>
- <https://www.zonamaker.com/>
- <https://www.thegreenmonkey.es/>
- <https://www.pcworld.es/>
- <https://forum.arduino.cc/>

Cálculo costes material

- <https://www.pccomponentes.com/>
- <https://www.amazon.es/>
- <https://es.aliexpress.com/>
- <https://www.robotshop.com/>
- <https://es.rs-online.com/>



LIBROS:

Instalaciones domóticas Edición 2020

Escrito por Luis Miguel Cerdá Filiu, Manuel Gas Bueno

Arduino: Manual de Programación

Traducido y adaptado: José Manuel Ruiz Gutiérrez

11 ANEXOS

11.1 ANEXO 1. CÓDIGO COMPLETO

```
#define DEBUG true

#include "SPI.h"

#include "WebServer.h"

#include <IRremote.h>

const int LuzEntrada = 47; //Pin para accionar el relé

const int PIRPin = 2; //Pin para detectar movimiento

const int LDRPin = 0; //Pin de lectura del LDR

const int DetecGas = 48; //Pin para detectar gas

const int Zumbador = 3; //Pin accionar zumbador

const int LuzTrastero = 49; //Pin para accionar el relé

const int PIRPinTrastero = 4; //Pin para detectar movimiento

const int trigPin = 4; const int echoPin = 2; long duration, cm;

int pin11 = 52;

int pin = 13;

int buttonState = 0;

template<class T>

inline Print &operator <<(Print &obj, T arg)

{

    obj.print(arg);

    return obj;

}

//Variable IR

IRsend irsend;
```



```
#define PREFIX ""
```

```
WebServer webserver(PREFIX, 80 ); //Asignamos el puerto
```

```
void setup()
```

```
{
```

```
  Serial.begin(115200);
```

```
  Serial2.begin(115200); // velocidad de transmisión podría ser diferente, aunque  
  generalmente es la que usa el módulo modulo
```

```
  pinMode(22, OUTPUT); //configuramos los pines como salidas
```

```
  digitalWrite(22, LOW);
```

```
  pinMode(23, OUTPUT); //configuramos los pines como salidas
```

```
  digitalWrite(23, LOW);
```

```
  pinMode(24, OUTPUT); //configuramos los pines como salidas
```

```
  digitalWrite(24, LOW);
```

```
  pinMode(25, OUTPUT); //configuramos los pines como salidas
```

```
  digitalWrite(25, LOW);
```

```
  pinMode(26, OUTPUT); //configuramos los pines como salidas
```

```
  digitalWrite(26, LOW);
```

```
  pinMode(27, OUTPUT); //configuramos los pines como salidas
```

```
  digitalWrite(27, LOW);
```

```
  pinMode(28, OUTPUT); //configuramos los pines como salidas
```

```
  digitalWrite(28, LOW);
```

```
  pinMode(29, OUTPUT); //configuramos los pines como salidas
```

```
  digitalWrite(29, LOW);
```

```
  pinMode(30, OUTPUT); //configuramos los pines como salidas
```

```
  digitalWrite(30, LOW);
```

```
  pinMode(31, OUTPUT); //configuramos los pines como salidas
```



digitalWrite(31, LOW);

pinMode(32, OUTPUT); //configuramos los pines como salidas

digitalWrite(32, LOW);

pinMode(33, OUTPUT); //configuramos los pines como salidas

digitalWrite(33, LOW);

pinMode(34, OUTPUT); //configuramos los pines como salidas

digitalWrite(34, LOW);

pinMode(35, OUTPUT); //configuramos los pines como salidas

digitalWrite(32, LOW);

pinMode(36, OUTPUT); //configuramos los pines como salidas

digitalWrite(36, LOW);

pinMode(37, OUTPUT); //configuramos los pines como salidas

digitalWrite(37, LOW);

pinMode(38, OUTPUT); //configuramos los pines como salidas

digitalWrite(38, LOW);

pinMode(39, OUTPUT); //configuramos los pines como salidas

digitalWrite(39, LOW);

pinMode(40, OUTPUT); //configuramos los pines como salidas

digitalWrite(40, LOW);

pinMode(41, OUTPUT); //configuramos los pines como salidas

digitalWrite(41, LOW);

pinMode(42, OUTPUT); //configuramos los pines como salidas

digitalWrite(42, LOW);

pinMode(43, OUTPUT); //configuramos los pines como salidas

digitalWrite(43, LOW);

pinMode(44, OUTPUT); //configuramos los pines como salidas

```
digitalWrite(44, LOW);

pinMode(45, OUTPUT); //configuramos los pines como salidas

digitalWrite(45, LOW);

pinMode(46, OUTPUT); //configuramos los pines como salidas

digitalWrite(46, LOW);

pinMode(LuzEntrada, OUTPUT); //LuzEntrada como salida

pinMode(PIRPin, INPUT); //PIRPin como entrada

pinMode(DetecGas, INPUT); //Entrada

pinMode(Zumbador, OUTPUT); //Salida

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(pin11, OUTPUT);

digitalWrite(pin11, LOW);

pinMode(LuzTrastero, OUTPUT); //LuzEntrada como salida

pinMode(PIRPinTrastero, INPUT); //PIRPin como entrada

webserver.begin();

webserver.addCommand("acon", &acon); //Encendido y apagado

webserver.addCommand("up", &up); // Subir temperatura

webserver.addCommand("down", &down); //Bajar temperatura

//sendData("AT+RST\r\n",2000, DEBUG); // resetea el modulo

//sendData("AT+CWMODE=3\r\n",1000, DEBUG); // configure el módulo con AP y
Estación al mismo tiempo

//sendData("AT+CWJAP='SSIDWIFI','CONTRASEÑAWIFI'\r\n",1000,DEBUG);//
cambiar estos datos por los de la red

sendData("AT+CIFSR\r\n", 5000, DEBUG); // Este comandos nos indica la
configuración ip que ha tomado nuestro módulo

sendData("AT+CIPMUX=1\r\n", 1000, DEBUG); // Configura multiples conexiones
```



```
sendData("AT+CIPSERVER=1,80\r\n", 1000, DEBUG); // Inicializa el servidor web
}

void loop()
{
  bool state = digitalRead(DetecGas);

  if (!state)
  {
    digitalWrite(Zumbador, HIGH);

    delay(10000); //Espera 10 segundos
  }
  else {
    digitalWrite(Zumbador, LOW); //Sino se desactiva
  }

  if (analogRead(LDRPin) > 150 && PIRPin == HIGH ) { //Si el valor del LDR es menor a
150 y el sensor detecta movimiento se activa LuzEntrada

    digitalWrite(LuzEntrada, HIGH);

    delay(10000); //Espera 10 segundos
  }
  else {
    digitalWrite(LuzEntrada, LOW); //Sino se desactiva
  }

  if ( buttonState == HIGH) {

    digitalWrite(trigPin, LOW);

    delayMicroseconds(2);
```

```
digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

cm = (duration / 2) / 29.1;

if (cm < 20)
{
    digitalWrite(pin11, HIGH);
}

}

else
{
    buttonState == LOW;

    digitalWrite(pin11, LOW);
}

webservice.processConnection();

if (Serial2.available()) // Verifica si el módulo está disponible
{
    if (Serial2.find("+IPD,") //Busca el campo de texto +IPD
    {
        delay(10); //Espera un microsegundo para leer todos los datos del puerto serie

        int connectionId = Serial2.read() - 48;

        Serial2.find("pin="); // Avanza el cursor a "pin="

        int pinNumber = (Serial2.read() - 48) * 10; // Conseguir primer número, si el pin es
13, entonces el primer número es 1, entonces se multiplican para obtener 10
```

pinNumber += (Serial2.read() - 48); // consigue el segundo número, si el número de pin es 13, entonces el segundo número es 3, a continuación, le añade al primero

```
digitalWrite(pinNumber, !digitalRead(pinNumber));
```

```
buttonState = digitalRead(pin);
```

```
String closeCommand = "AT+CIPCLOSE=";
```

```
closeCommand += connectionId;
```

```
closeCommand += "\r\n";
```

```
sendData(closeCommand, 10, DEBUG); // cierra la conexión
```

```
}
```

```
}
```

```
}
```

```
String sendData(String command, const int timeout, boolean debug)
```

```
{
```

```
String cadena = "";
```

```
Serial2.print(command); // enviar el carácter leído al modulo
```

```
long int time = millis();
```

```
while ( (time + timeout) > millis())
```

```
{
```

```
while (Serial2.available())
```

```
{
```

```
// Muestra los Datos de salida del módulo por el seguimiento de la serie
```

```
char c = Serial2.read(); //Lee el próximo carácter
```

```
cadena += c;
```

```
}
```

```
}
```

```
if (debug)
```



```
{  
  
    Serial.print(cadena);  
  
}  
  
return cadena;  
  
}  
  
//Funciones que enviarán el código IR  
  
void acon(WebServer &server, WebServer::ConnectionType type, char *url_tail, bool  
tail_complete)  
  
{  
  
    if (type == WebServer::POST)  
  
    {  
  
        server.httpFail();  
  
        return;  
  
    }  
  
    server.httpSuccess();  
  
    irsend.sendNEC(0x9CC72A2E, 32);  
  
    delay(40);  
  
}  
  
void up(WebServer &server, WebServer::ConnectionType type, char *url_tail, bool  
tail_complete)  
  
{  
  
    if (type == WebServer::POST)  
  
    {  
  
        server.httpFail();  
  
        return;  
  
    }  
  
}
```



```
}  
  
server.httpSuccess();  
  
irsend.sendNEC(0x9CC72A2E, 32);  
  
delay(40);  
  
}  
  
void down(WebServer &server, WebServer::ConnectionType type, char *url_tail, bool  
tail_complete)  
{  
    if (type == WebServer::POST)  
    {  
        server.httpFail();  
  
        return;  
    }  
  
    server.httpSuccess();  
  
    irsend.sendNEC(0xFB437BF6, 32);  
  
    delay(40);  
  
}
```

11.2 ANEXO 2. CÁLCULO SENSOR DE PROXIMIDAD

El sensor de proximidad de ultrasonidos para calcular la distancia a la que se encuentra un objeto en centímetros necesita una serie de parámetros que calcularemos a continuación.

Como hemos mencionado anteriormente, el primer paso es calcular el tiempo en μs que tarda en cambiar de estado el pin que hemos seleccionado. Una vez este tiempo está calculado, como la velocidad del sonido es de 343 m/s, calculamos el tiempo que necesitamos para recorrer un metro y, para realizar esta medida más cómoda, pasaremos la medida de metros a centímetros.

Tiempo que tarda en recorrer un metro:

$$T = \frac{1}{343} = 0,00291 \text{ s}$$

Medida en microsegundos por centímetro:

$$0,00291 \times \frac{1000000 \mu s}{segundo} \times \frac{1 \text{ metro}}{100 \text{ cm}} = 29,1 \mu s/cm$$

Como nuestra función mide el tiempo que tarda el pulso en ir y venir la distancia recorrida será la mitad:

$$Distancia = \frac{Duración}{29,1} \times \frac{1}{2}$$

Donde *duración* es el tiempo que tarda en cambiar de estado el pin definido.

11.3 ANEXO 3. DECODIFICADOR IR

Para decodificar el mando a distancia es necesario el uso de un receptor IR y el siguiente código:

```
#include <IRremote.h>

int RECV_PIN = 19;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()

{

  Serial.begin(9600);

  irrecv.enableIRIn();

}

void dump(decode_results *results) {

  int count = results->rawlen;

  if (results->decode_type == UNKNOWN) {

    Serial.print("Unknown encoding: ");

  }

  else if (results->decode_type == NEC) {

    Serial.print("Decoded NEC: ");

  }

  else if (results->decode_type == SONY) {

    Serial.print("Decoded SONY: ");

  }

  else if (results->decode_type == RC5) {

    Serial.print("Decoded RC5: ");

  }

  else if (results->decode_type == RC6) {

    Serial.print("Decoded RC6: ");

  }

}
```

```
}  
  
else if (results->decode_type == PANASONIC) {  
  
    Serial.print("Decoded PANASONIC - Address: ");  
  
    Serial.print(results->panasonicAddress, HEX);  
  
    Serial.print(" Value: ");  
  
}  
  
else if (results->decode_type == JVC) {  
  
    Serial.print("Decoded JVC: ");  
  
}  
  
Serial.print(results->value, HEX);  
  
Serial.print(" (");  
  
Serial.print(results->bits, DEC);  
  
Serial.println(" bits");  
  
Serial.print("Raw (");  
  
Serial.print(count, DEC);  
  
Serial.print("):                                     ");  
  
for (int i = 0; i < count; i++) {  
  
    if ((i % 2) == 1) {  
  
        Serial.print(results->rawbuf[i]*USECPERTICK, DEC);  
  
    }  
  
    else {  
  
        Serial.print(-(int)results->rawbuf[i]*USECPERTICK, DEC);  
  
    }  
  
    Serial.print(" ");  
  
}  
  
Serial.println("");  
  
}
```



```
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    dump(&results);
    irrecv.resume();
  }
}
```

El siguiente código muestra por puerto serial los códigos según el botón que presionemos. En nuestro caso, estos son: encendido, apagado, subir temperatura y bajar temperatura.

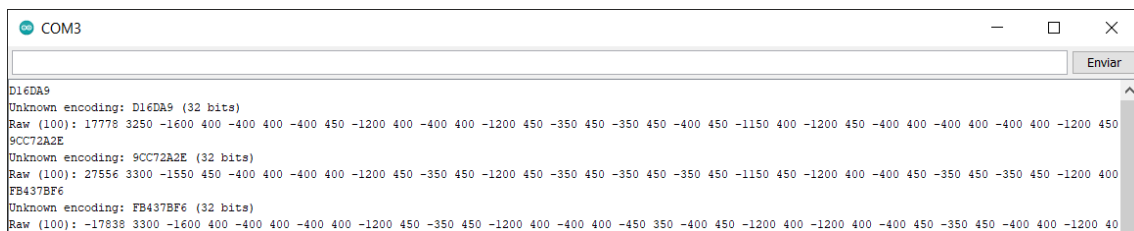


Imagen 48. Códigos descifrados mano aire acondicionado

Con estos códigos, desarrollamos el código Arduino de nuestro sistema domótico que permitirá tanto el encendido y apagado como subir y bajar temperatura.

```
#include <IRremote.h>

template<class T>

inline Print &operator <<(Print &obj, T arg)
{ obj.print(arg); return obj; }

IRsend irsend;

#define PREFIX "192.168.1.141"

WebServer webserver(PREFIX, 80 ); //Asignamos el puerto

//Funciones
void acon(WebServer &server, WebServer::ConnectionType type, char *url_tail, bool
tail_complete)
```



```
{
  if (type == WebServer::POST)
  {
    server.httpFail();
    return;
  }
  server.httpSuccess();

  irsend.sendNEC(0xD16DA9, 32); //

  delay(40);
}

void up(WebServer &server, WebServer::ConnectionType type, char *url_tail, bool
tail_complete)
{
  if (type == WebServer::POST)
  {
    server.httpFail();

    return;
  }

  server.httpSuccess();

  irsend.sendNEC(0x9CC72A2E, 32); //irsend.send(COMPANYIR)(0x(IRCODE), (BITS));

  delay(40);
}

void down(WebServer &server, WebServer::ConnectionType type, char *url_tail, bool
tail_complete)
{
  if (type == WebServer::POST)
  {
    server.httpFail();

    return;
  }
}
```

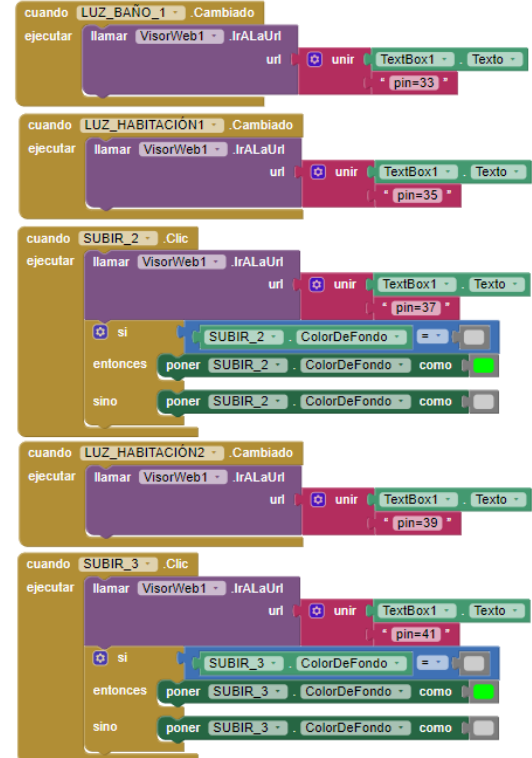
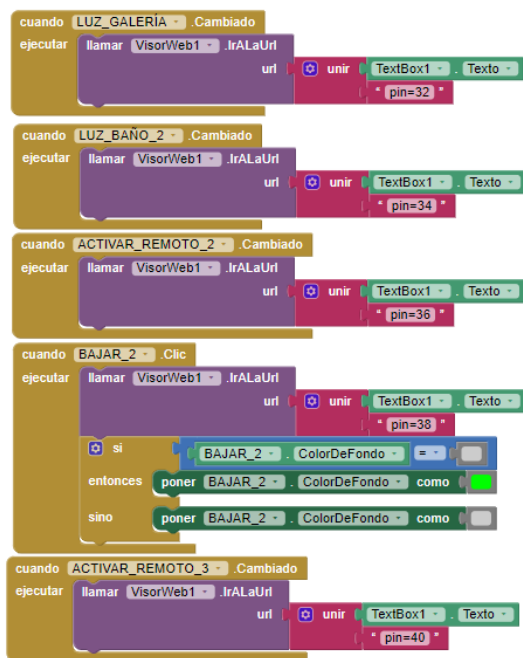
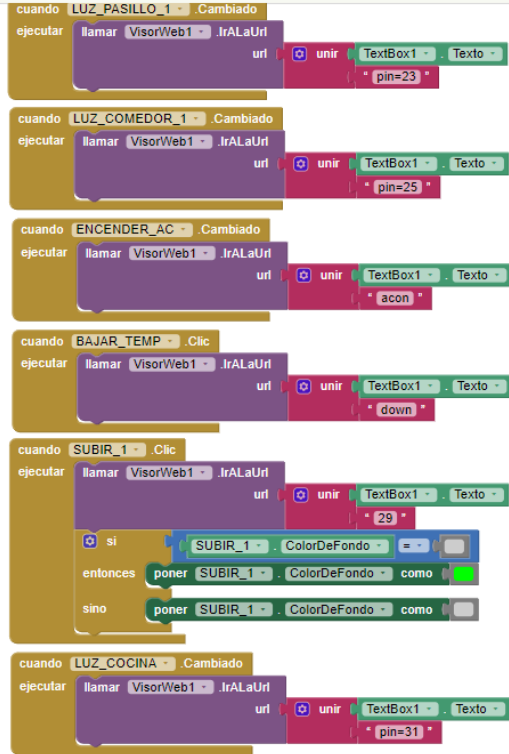
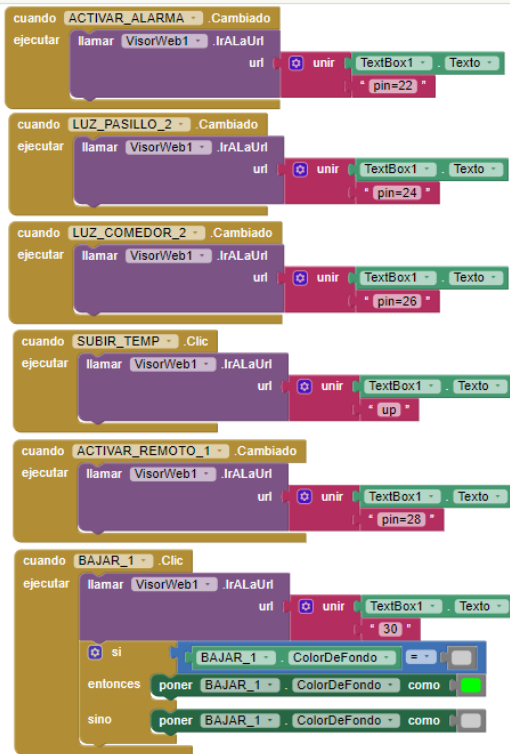


```
}  
  
server.httpSuccess();  
  
irsend.sendNEC(0xFB437BF6, 32); /  
  
delay(40);  
  
}  
  
void setup()  
{  
  
  Serial.begin(115200);  
  
  Serial2.begin(115200);  
  
  Serial2.addCommand("acon", &acOn); //Encendido y apagado  
  
  Serial2.addCommand("tempup", &tempUp); //Subir temperatura  
  
  Serial2.addCommand("tempdown", &tempDown); //Bajar temperatura  
  
}  
  
void loop()  
{  
  
  webserver.processConnection();  
  
}  
  
}
```

Finalmente, este código se implementará al código general de todo el sistema domótico que permitirá su accionamiento desde la aplicación móvil.

11.4 ANEXO 4. CÓDIGO PROGRAMACIÓN APLICACIÓN DOMÓTICA

<p>cuando IP - .Clic ejecutar poner TextBox1 - .Visible - como cierto -</p>	<p>cuando IP - .ClicLargo ejecutar poner TextBox1 - .Visible - como falso -</p>
<p>cuando ENTRADA_1 - .Clic ejecutar poner ACTIVAR_ALARMA - .Visible - como cierto -</p>	<p>cuando ENTRADA_1 - .ClicLargo ejecutar poner ACTIVAR_ALARMA - .Visible - como falso -</p>
<p>cuando PASILLOS_1 - .Clic ejecutar poner PASILLOS_2 - .Visible - como cierto -</p>	<p>cuando PASILLOS_1 - .ClicLargo ejecutar poner PASILLOS_2 - .Visible - como falso -</p>
<p>cuando COMEDOR_1 - .Clic ejecutar poner LUMINARIA_COMEDOR - .Visible - como cierto - poner CLIMATIZACIÓN_0 - .Visible - como cierto - poner PERSIANAS_COMEDOR - .Visible - como cierto - poner LUCES_0 - .Visible - como cierto - poner CLIMATIZACIÓN_1 - .Visible - como cierto - poner PERSIANAS_COMEDOR_1 - .Visible - como cierto -</p>	<p>cuando COMEDOR_1 - .ClicLargo ejecutar poner LUMINARIA_COMEDOR - .Visible - como falso - poner CLIMATIZACIÓN_0 - .Visible - como falso - poner PERSIANAS_COMEDOR - .Visible - como falso - poner LUCES_0 - .Visible - como falso - poner CLIMATIZACIÓN_1 - .Visible - como falso - poner PERSIANAS_COMEDOR_1 - .Visible - como falso -</p>
<p>cuando COCINA_1 - .Clic ejecutar poner COCINA_3 - .Visible - como cierto -</p>	<p>cuando COCINA_1 - .ClicLargo ejecutar poner COCINA_3 - .Visible - como falso -</p>
<p>cuando BAÑOS_1 - .Clic ejecutar poner LUCES_BAÑOS - .Visible - como cierto -</p>	<p>cuando BAÑOS_1 - .ClicLargo ejecutar poner LUCES_BAÑOS - .Visible - como falso -</p>
<p>cuando HABITACIÓN1_1 - .Clic ejecutar poner LUMINARIA_HABITACION1 - .Visible - como cierto - poner LUZ_HABITACION1 - .Visible - como cierto - poner PERSIANAS_HABITACION1 - .Visible - como cierto - poner PERSIANAS_HABITACION1_1 - .Visible - como cierto -</p>	<p>cuando HABITACIÓN1_1 - .ClicLargo ejecutar poner LUMINARIA_HABITACION1 - .Visible - como falso - poner LUZ_HABITACION1 - .Visible - como falso - poner PERSIANAS_HABITACION1 - .Visible - como falso - poner PERSIANAS_HABITACION1_1 - .Visible - como falso -</p>
<p>cuando HABITACIÓN1_1 - .Clic ejecutar poner LUMINARIA_HABITACION1 - .Visible - como cierto - poner LUZ_HABITACION1 - .Visible - como cierto - poner PERSIANAS_HABITACION1 - .Visible - como cierto - poner PERSIANAS_HABITACION1_1 - .Visible - como cierto -</p>	<p>cuando HABITACIÓN1_1 - .ClicLargo ejecutar poner LUMINARIA_HABITACION1 - .Visible - como falso - poner LUZ_HABITACION1 - .Visible - como falso - poner PERSIANAS_HABITACION1 - .Visible - como falso - poner PERSIANAS_HABITACION1_1 - .Visible - como falso -</p>
<p>cuando HABITACIÓN2_1 - .Clic ejecutar poner LUMINARIA_HABITACION2 - .Visible - como cierto - poner LUZ_HABITACION2 - .Visible - como cierto - poner PERSIANAS_HABITACION2 - .Visible - como cierto - poner PERSIANAS_HABITACION2_1 - .Visible - como cierto -</p>	<p>cuando HABITACIÓN2_1 - .ClicLargo ejecutar poner LUMINARIA_HABITACION2 - .Visible - como falso - poner LUZ_HABITACION2 - .Visible - como falso - poner PERSIANAS_HABITACION2 - .Visible - como falso - poner PERSIANAS_HABITACION2_1 - .Visible - como falso -</p>
<p>cuando HABITACIÓN3_1 - .Clic ejecutar poner LUMINARIA_HABITACION3 - .Visible - como cierto - poner LUZ_HABITACION3 - .Visible - como cierto - poner PERSIANAS_HABITACION3 - .Visible - como cierto - poner PERSIANAS_HABITACION3_1 - .Visible - como cierto -</p>	<p>cuando HABITACIÓN3_1 - .ClicLargo ejecutar poner LUMINARIA_HABITACION3 - .Visible - como falso - poner LUZ_HABITACION3 - .Visible - como falso - poner PERSIANAS_HABITACION3 - .Visible - como falso - poner PERSIANAS_HABITACION3_1 - .Visible - como falso -</p>
<p>cuando Plano - .Clic ejecutar poner PLANO - .Visible - como cierto -</p>	<p>cuando Plano - .ClicLargo ejecutar poner PLANO - .Visible - como falso -</p>



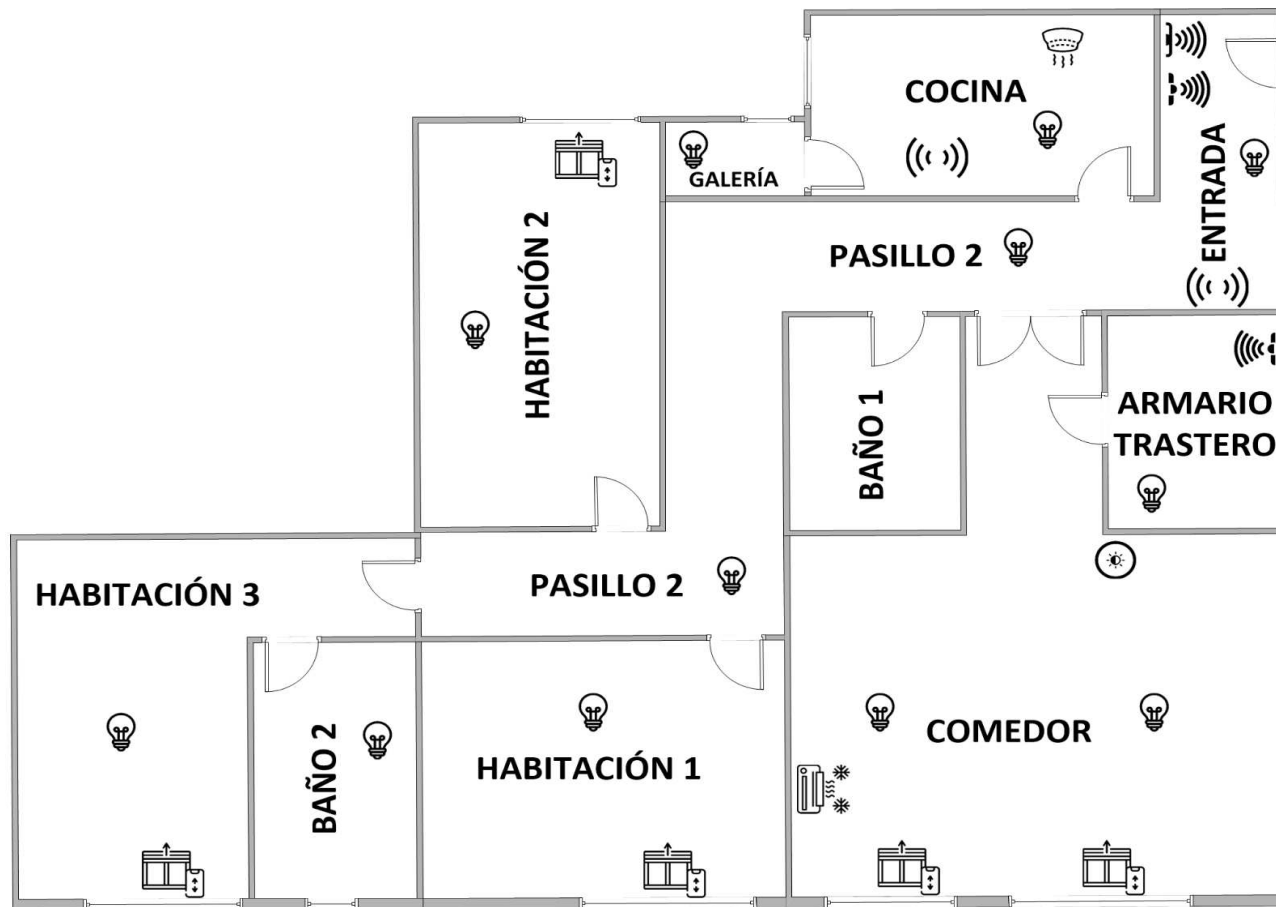
```
cuando BAJAR_3 .Clic
ejecutar
  llamar VisorWeb1 .IrALaUrl
  url unir TextBox1 .Texto
  " pin=42 "
  si BAJAR_3 .ColorDeFondo =
  entonces poner BAJAR_3 .ColorDeFondo como
  sino poner BAJAR_3 .ColorDeFondo como

cuando ACTIVAR_REMOTO_4 .Cambiado
ejecutar
  llamar VisorWeb1 .IrALaUrl
  url unir TextBox1 .Texto
  " pin=44 "

cuando BAJAR_4 .Clic
ejecutar
  llamar VisorWeb1 .IrALaUrl
  url unir TextBox1 .Texto
  " pin=46 "
  si BAJAR_4 .ColorDeFondo =
  entonces poner BAJAR_4 .ColorDeFondo como
  sino poner BAJAR_4 .ColorDeFondo como
```

```
cuando LUZ_HABITACION3 .Cambiado
ejecutar
  llamar VisorWeb1 .IrALaUrl
  url unir TextBox1 .Texto
  " pin=43 "

cuando SUBIR_4 .Clic
ejecutar
  llamar VisorWeb1 .IrALaUrl
  url unir TextBox1 .Texto
  " pin=45 "
  si SUBIR_4 .ColorDeFondo =
  entonces poner SUBIR_4 .ColorDeFondo como
  sino poner SUBIR_4 .ColorDeFondo como
```



-  Zumbador
-  Persiana
-  Climatizador
-  Luz
-  Sensor de movimiento
-  Sensor de proximidad
-  Sensor de luminosidad
-  Sensor de gas y humo

	Fecha	Nombre	Firma:	 Escuela Técnica Superior de Ingeniería del Diseño
Dibujado	05/05/2020	Antonio Andreu Escrava		
Comprobado	08/05/2020	Houcine Hassan Mohamed		
id.s.normas	-	-		
Escala:	1:50			Lamina n. 1
				N. Alumno: -
				Curso: 4º