



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas

**Análisis para un sistema domótico con la arquitectura Arduino y
Raspberry Pi, sobre TCP/IP.**

Monografía previa a la obtención de título de ingeniero de sistemas

Autor

Orlando Patricio Chacón Molina

Director

Ing. Diego Chacón. Mdhd

Cuenca-Ecuador

2014

Dedicatoria

Dedicatoria.

Dedico esta monografía a mi familia. A mis padres, Carlos y Esther, por haberme brindado su apoyo durante todo mi tiempo de estudio. A mis hermanos, Milton y Julio, fuente de inspiración y confianza. A mi esposa Gladys e hija Kristel, porque de ellas he robado el tiempo que me ha faltado.

Agradecimientos

Quiero agradecer, con toda sinceridad, a todos los profesores que con franqueza han compartido su conocimiento, especialmente a aquellos que me han “enseñado a aprender”.

Agradezco puntualmente a:

Ingeniero Marcos Orellana, por su trabajo como director de escuela.

Ingeniero Esteban Crespo, quien se ha tomado el tiempo para realizar varias revisiones del diseño.

Ingeniero Kenneth Palacio, quien ha aportado con ideas y observaciones muy valiosas.

Ingeniero Diego Chacón, quien en calidad de director de monografía me ha brindado todo su apoyo y soporte desde la concepción de la idea hasta su finalización.

A mi hermano Julio, que ha dedicado su tiempo para realizar revisiones, sugerencias y varios de los diseños gráficos para la aplicación.

Índice

Contenido

Dedicatoria	i
Agradecimientos.....	ii
Índice.....	iii
Índice de ilustraciones	vi
Índice de tablas	vii
Resumen.....	viii
Abstract.....	ix
Introducción.....	x
Objetivos	xi
Objetivo Principal	xi
Objetivos Específicos:.....	xi
1 Fundamentos teóricos	1
1.1 Introducción a la domótica	1
1.2 Componentes de una red domótica.....	1
1.2.1 Sensores	1
1.2.2 Actuadores	3
1.2.3 Controladores	4
1.2.4 Aparatos electrónicos inteligentes	4
1.2.5 Sistemas informáticos.....	5
1.3 Redes Domésticas	5
1.3.1 Red de Control	5
1.3.2 Red de Datos.....	5
1.3.3 Red Multimedia.....	5
1.4 Introducción a la plataforma Arduino	5
1.5 Introducción a la plataforma Linux (Raspbian OS sobre Raspberry Pi)	7
2 Definición de la arquitectura	9
2.1 Arquitectura de Controladores Terminales (CT)	9

2.1.1 Configuración y programación de tarjetas Arduino como Controlador Terminal.....	10
2.1.2 Sensores	22
2.1.3 Actuadores	23
2.2 Dispositivos de Control o interacción con el sistema	25
2.2.1 Control Físico – Pulsadores.....	25
2.2.2 Control por Software.....	26
2.3 Circuitos Soportados.....	26
2.4 Arquitectura de Software.....	27
2.4.1 Configuración y programación de Raspberry Pi como Controlador Central.....	27
2.4.2 Estructura de Base de Datos (DB - Data Base).....	31
2.4.3 Estructura de directorios y archivos.....	37
2.4.4 API (Application Programming Interface) para desarrolladores.....	38
3 Comunicación.....	39
3.1 TCP/IP	39
3.2 Servicios Web	39
3.2.1 Controlador Terminal (CT).....	39
3.2.2 Mini Servidor.....	52
3.3 Interfaces de Hardware	57
3.3.1 Pulsadores	57
3.3.2 Actuadores	57
3.3.3 Sensores	58
4 Demostración de implementación y pruebas	59
4.1 Análisis de requerimientos	59
4.1.1 Ejemplo	60
4.2 Diseño.....	61
4.2.1 Ejemplo	63
4.3 Implementación.....	64
4.3.1 Configuración de dispositivos	64

4.3.2 Instalación de equipos	64
4.4 Pruebas	64
4.4.1 Ejemplo	64
4.5 Mantenimiento	64
Conclusiones	65
Recomendaciones	66
Bibliografía	67
ANEXOS	68

Índice de ilustraciones

Figura 1 Diagrama electromecánico de un relé (Wikipedia, 2014)	3
Figura 2 Esquema básico de Regulación de Intensidad (Txapuzas, 2014).....	4
Figura 3 Componentes de Arduino UNO (BLAM Electronics, 2014)	6
Figura 4 Componentes de Raspberry Pi (Xataka, 2014).....	7
Figura 5 Distribución de los pines en un CT (Arduino UNO)	9
Figura 6 Capas del Controlador Terminal	11
Figura 7 Simbología elementos domóticos.	23
Figura 8 Colores de cables	23
Figura 9 Diagrama del actuador de tipo Encendido/Apagado	24
Figura 10 Diagrama del circuito de Regulación de Intensidad (Txapuzas, 2014)	25
Figura 11 Esquema del circuito de pulsadores	26
Figura 12 Diagrama Entidad-Relación de la Base de Datos	32
Figura 13 Esquema de comunicación Cliente => CT [=> Actuador] => Cliente	40
Figura 14 Esquema de comunicación Pulsadores => CT => CC => CT	49
Figura 15 Esquema de comunicación Sensores => CT => CC => CT =>Actuador	50
Figura 16 Ventana de configuración de CTs.....	50
Figura 17 Esquema de comunicación Cliente => CC => CT => CC => Cliente	55
Figura 18 Ejemplo una red usando diverso medios físicos.	59
Figura 19 Plano de la vivienda de ejemplo (Pedersen, Bienes Raices, 2014)....	61
Figura 20 Plano Domótico - Eléctrico.....	63

Índice de tablas

Tabla 1 Tipos de sensores	22
Tabla 2 Circuitos soportados	26
Tabla 3 Lista de tablas de la Base de Datos.....	31
Tabla 4 DB. Tabla actions	32
Tabla 5 DB. Tabla element_logs.....	33
Tabla 6 DB. Tabla element_types.....	33
Tabla 7 DB. Tabla elements	33
Tabla 8 DB. Tabla element_sensors.....	34
Tabla 9 DB. Tabla parameters.....	34
Tabla 10 DB. Tabla push_buttons	34
Tabla 11 DB. Tabla sensor_types.....	35
Tabla 12 DB. Tabla subnets	35
Tabla 13 DB. Tabla tc.....	35
Tabla 14 DB. Tabla tc_sensors.....	36
Tabla 15 DB. Tabla user_types	36
Tabla 16 DB. Tabla users	36
Tabla 17 Directorios y archivos del software del CT	38
Tabla 18 Estructura de la respuesta HTTP, recurso 102	42
Tabla 19 Estructura de una petición al recurso 902 y su ubicación en la memoria EEPROM.....	44
Tabla 20 Listado de elementos, sensores y CTs requeridos (captura de la plantilla)	61

Resumen

En este documento se describe el procedimiento realizado para la creación de una arquitectura domótica, basada en dispositivos de hardware y software abiertos (Arduino y Raspberry Pi), que funciona sobre una red TCP/IP. Inicia con una breve descripción de los fundamentos teóricos sobre los componentes de una red domótica. Luego se muestra la definición de la arquitectura domótica, tanto a nivel de hardware como software. Seguido se detallan los métodos de comunicación entre dispositivos y elementos involucrados en una red domótica. Finalmente se realiza una demostración de implementación de la arquitectura desarrollada.

ABSTRACT

This paper describes the process followed for the creation of automation or domotic architecture based on hardware devices and open software (Arduino and Raspberry Pi) which works over a TCP / IP network. We begin with a brief description of the theoretical foundations of home automation network components. The definition of domotic architecture is then shown at both hardware and software levels. Next, the methods of communication between devices and elements involved in an automation network are detailed. Finally, a demonstration on how to implement the architecture developed is performed.




Translated by,

Lic. Lourdes Crespo

Introducción

Las infraestructuras domóticas ofrecen un alto nivel de confort y control de un hogar, sin embargo, los costos elevados de los equipos hacen que su aplicación sea muy reducida. Por esta razón, se pretende crear una arquitectura basada en estándares de comunicación existentes y dispositivos de bajo costo, como Arduino y Raspberry Pi.

La domótica es un área en la que se puede experimentar para crear tecnología propia, como alternativa a las soluciones comerciales de alto costo. Se propone desarrollar una arquitectura domótica a partir dispositivos existentes, modificando, configurando e integrando componentes que permitan desarrollar dispositivos para los que quizá no fueron pensados, como “piezas aisladas”. Desde un punto de vista modesto, pero ambicioso, sería un avance significativo si a largo plazo se logra el desarrollo integral de la tecnología involucrada en esta investigación.

Si bien al finalizar esta investigación la arquitectura debe ser funcional y cumplir con los objetivos establecidos, para que pueda llegar a popularizarse debe madurar en muchos aspectos como funcionalidad, usabilidad y sobre todo fiabilidad. La madurez de la arquitectura no llegará sino con nuevas investigaciones que fortalezcan este pequeño aporte. Además, para el verdadero crecimiento, no solo de este trabajo, sino de cualquier área e índole, es necesario que diversas escuelas (no solo aquellas con relación técnica) se integren para que, unos con su capacidad comunicacional, otros con sus diseños o cualquier aporte pertinente, se logre obtener un solo producto, uno de calidad.

Objetivos

Objetivo Principal

Realizar el análisis para un sistema domótico con la arquitectura Arduino y Raspberry Pi, sobre TCP/IP.

Objetivos Específicos:

- Establecer las especificaciones para la comunicación física y lógica de dispositivos.
- Desarrollar el software para el control y comunicación de dispositivos
- Crear una Interfaz de Programación de Aplicaciones (API – Application Programming Interface) para que otros desarrolladores puedan ampliar las funcionalidades de control del sistema.
- Demostrar, mediante un prototipo, la arquitectura analizada.

Capítulo I

1 Fundamentos teóricos

1.1 Introducción a la domótica

Etimológicamente, domótica proviene del latín domus (casa) y del griego αὐτόματος (automática). Se usa el término para determinar el intrincado lugar de encuentro entre vivienda, tecnología y el ser humano. (Domínguez & Sáez Vacas, 2006, pág. 15)

Algunos autores expresan que el inicio de la domótica se remonta a los años sesenta cuando aparecieron los primeros dispositivos X-10, que funcionaban mediante las líneas eléctricas del hogar (Madrid vive ahorrando energía, 2007, pág. 15); sin embargo, con la popularización y evolución de los computadores personales y posteriormente los dispositivos móviles inteligentes, se ha dado lugar a la aparición y evolución de los sistemas domóticos como KNX, Lonworks y un sinnúmero de otras arquitecturas, tanto abiertas como propietarias.

Una de las principales funciones de la domótica es ofrecer confort al usuario, al tiempo que reduzca el consumo energético. Especialmente debe ofrecer facilidad en el control, monitoreo y gestión de circuitos de iluminación, climatización, control de acceso, control de utilización energética. (Domínguez & Sáez Vacas, 2006)

1.2 Componentes de una red domótica

Para el funcionamiento de una red domótica convergen una gran diversidad de componentes electrónicos, mecánicos, eléctricos, hidráulicos y otros. Para esta investigación se ha clasificado en cinco categorías de acuerdo a su función dentro del sistema domótico, estos son: sensores, actuadores, controladores, aparatos electrónicos inteligentes y sistemas informáticos.

1.2.1 Sensores

Son dispositivos que recopilan información del entorno (presencia, calor, movimiento, humedad, etc.) y la transmiten a otros equipos para su procesamiento (Domínguez & Sáez Vacas, 2006, pág. 51). En base a esta información pueden tomarse decisiones sobre cambios de estado de equipos controlados.

La mayoría de los sensores no producen directamente voltajes de salida, al contrario, actúan como dispositivos pasivos, generalmente resistivos, cuyo valor cambia en función del estímulo físico que es capaz de detectar (Wilson, 2005, pág. 17). En un circuito básico basta con agregar una resistencia de valor fijo (resistencia

de carga) en serie con el sensor, conectar a una fuente de poder y medir el voltaje en la resistencia de carga. Luego, mediante la ley de OHM o las Kirchhoff se puede calcular el valor de la resistencia, voltaje o corriente, y de esta forma es posible conocer el nivel del fenómeno medido. Entonces, si se conoce la relación entre la resistencia y el fenómeno físico, se podrá obtener el valor leído por el sensor.

Para conocer el nivel de voltaje de la salida se puede aplicar la siguiente formula:

$$(V_S = \frac{R_S}{R_1 + R_S} V_{in})$$

Donde, V_S es el nivel de voltaje medido en la resistencia de salida (carga), R_1 es la resistencia variable correspondiente al sensor y V_{in} es el voltaje de entrada. Finalmente, lo que interesa es conocer el voltaje en R_1 , para lo cual se debe aplicar la siguiente formula

$$V_1 = V_{in} - V_S.$$

Dentro de los tipos de sensores, según el tipo de fenómeno susceptible de medición, que se usarán en este proyecto se encuentran:

1.2.1.1 Fotorresistencia (LDR)

Una fotorresistencia o *LDR* (del inglés - *Light Dependant Resistor*) es un sensor basado en una resistencia cuyo valor disminuye cuando existe luz incidente sobre el mismo e incrementa cuando se encuentra en ambientes de baja iluminación (Margolis, 2012, pág. 186).

1.2.1.2 Termistor

Un termistor es un componente electrónico que actúa como resistencia variable y cambia su resistividad en función de la temperatura. Existen dos tipos:

- Coeficiente de temperatura positivo (*NTC- Negative Temperature Coefficient*)
 - Incrementa su resistividad con el aumento de temperatura.
- Coeficiente de temperatura negativo (*PTC- Positive Temperature Coefficient*)
 - Disminuye su resistividad con el aumento de temperatura.

1.2.1.3 Sensores de Humedad Resistivos.

Los sensores de humedad resistivos generalmente tienen una relación exponencial inversa a la humedad, básicamente debido a que el agua es un conductor de baja resistencia y cuanto más se acumule en el sensor, menor será su resistencia al paso de corriente. Estos sensores resistivos no responden linealmente al cambio de humedad, pero puede ser *linealizado* analógicamente o digitalmente.

1.2.2 Actuadores

Los actuadores son dispositivos capaces de recibir órdenes de los sistemas de control y en función de estas realizar cambios en los equipos, como encenderlos o apagarlos, incrementar o disminuir su nivel de intensidad y abrir o cerrar. (Domínguez & Sáez Vacas, 2006, pág. 52)

Los sistemas de control emiten señales a los actuadores para que estos últimos entren en funcionamiento, se detengan, incrementen, disminuyan o realicen otra acción para la cual hayan sido creados.

Entre los dispositivos actuadores más comunes y que se mencionaran en secciones posteriores se encuentran: relés (permiten la apertura o cierre de circuitos), motores, servomotores, electroválvulas, bombas hidráulicas eléctricas y circuitos de regulación de intensidad. Estos últimos están compuestos de TRIACs (del inglés - *Triode for Alternating Current*) DIACs (*Diode for Alternating Current*), resistencias, capacitores y opto-acopladores.

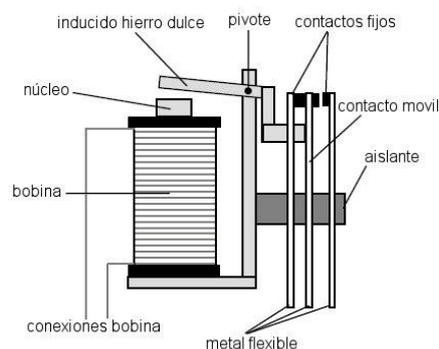


Figura 1 Diagrama electromecánico de un relé (Wikipedia, 2014)

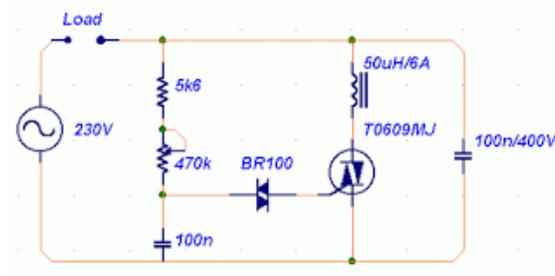


Figura 2 Esquema básico de Regulación de Intensidad (Txapuzas, 2014)

1.2.3 Controladores

Dentro del marco de esta investigación se define como controlador a los dispositivos que mantienen la inteligencia del sistema. En este contexto se diferencian dos tipos:

Controladores Terminales (CT): estos se encargan de la comunicación con los sensores, actuadores, pulsadores y los controladores centrales. Debe tener la capacidad de operar, si bien no con todas sus características funcionales, sin la presencia de un controlador central. Este puede recibir órdenes de los controladores centrales, al igual que enviarle notificaciones.

Controladores Centrales (CC): su función principal es la unificación del sistema domótico. A este dispositivo llegan las órdenes y notificaciones procedentes de los CT y de dispositivos informáticos externos que pueden ser móviles o fijos (PC). Cuando la orden procede de un dispositivo externo (móvil o fijo), el CC la registra y reenvía al CT para que este a su vez transfiera una señal al actuador y se realice el cambio de estado solicitado.

1.2.4 Aparatos electrónicos inteligentes

Son dispositivos que principalmente tienen como finalidad el ocio y entretenimiento. En esta categoría se encuentran los televisores inteligentes, teléfonos inteligentes y sistemas de audio conectados mediante una red multimedia y haciendo uso de la red de Internet. Finalmente, aunque no están catalogados como aparatos de ocio ni entretenimiento, los dispositivos de teleasistencia médica también son considerados dentro de esta categoría. (Domínguez & Sáez Vacas, 2006, pág. 55)

1.2.5 Sistemas informáticos

Los sistemas informáticos sirven de apoyo a la domótica mediante la presentación de interfaces para la comunicación entre diversos tipos de dispositivos y la gestión de la información de los diferentes equipos involucrados en la red.

1.3 Redes Domésticas

Mediante las redes domésticas es posible la intercomunicación de los dispositivos internos de una vivienda, al igual que con los exteriores a través de la pasarela residencial (CC). Dentro del hogar se pueden distinguir tres tipos de redes: red de datos, red de control y red multimedia.

1.3.1 Red de Control

Se encarga de transportar los datos de control de la red domótica, monitorear sensores, actuadores y electrodomésticos de la vivienda. Aunque es independiente de la red de datos, generalmente existe comunicación entre las dos mediante una pasarela residencial (CC), la cual es capaz de interactuar con los dos tipos de redes (Domínguez & Sáez Vacas, 2006, pág. 68). En esta investigación usa una red diseñada para datos (TCP/IP) como red de control.

1.3.2 Red de Datos

Está diseñada para la comunicación entre ordenadores y otros dispositivos informáticos, permitiendo el envío y recepción de mensajes, ficheros y comunicación con la red de Internet. Esta debe ser independiente de la red de control y multimedia, sus velocidades de transferencia de datos son mayores a las de la red control (Plan Avanza, pág. 22).

1.3.3 Red Multimedia

Su objetivo es la comunicación de audio y video, consolas de videojuego y otras plataformas de entretenimiento. Algunas de las aplicaciones soportadas por esta red son los videojuegos en red, la difusión de la señal de televisión de pago desde el decodificador al resto de estancias de la vivienda, el envío de audio y vídeo desde el portero automático (Domínguez & Sáez Vacas, 2006, pág. 67).

1.4 Introducción a la plataforma Arduino

Arduino es una plataforma de hardware abierta, diseñada para la realización de prototipos. Existen diversos modelos, pero, en este caso se usa específicamente el Arduino UNO, el cual consta de un microcontrolador (ATmega328), una entrada USB

que sirve como alimentación y conexión de datos a la PC, un conector de alimentación al que se puede conectar una fuente externa como baterías o pilas, un botón de reinicio, entradas y salidas tanto digitales (14 pines) como analógicas (6 pines).

La funcionalidad de este dispositivo puede extenderse mediante el uso de tarjetas complementarias para Arduino (*shields*) que pueden ser conectadas a la placa base. Por ejemplo: la tarjeta Ethernet Shield, es una placa que encaja sobre la tarjeta Arduino y usa los pines digitales 10,11,12 y 13 para la comunicación, permite agregar la funcionalidad de conexión Ethernet y por lo tanto proporcionar comunicación IP. Entre las funciones que se pueden efectuar con una tarjeta *Ethernet Shield* está la de actuar como servidor o cliente http. (Web Arduino, s.f.).

Si bien los proyectos desarrollados con Arduino pueden interactuar con el computador u otros dispositivos informáticos, no constituyen un requisito para su funcionamiento, es decir, tiene la capacidad de ejecutarse de manera independiente.

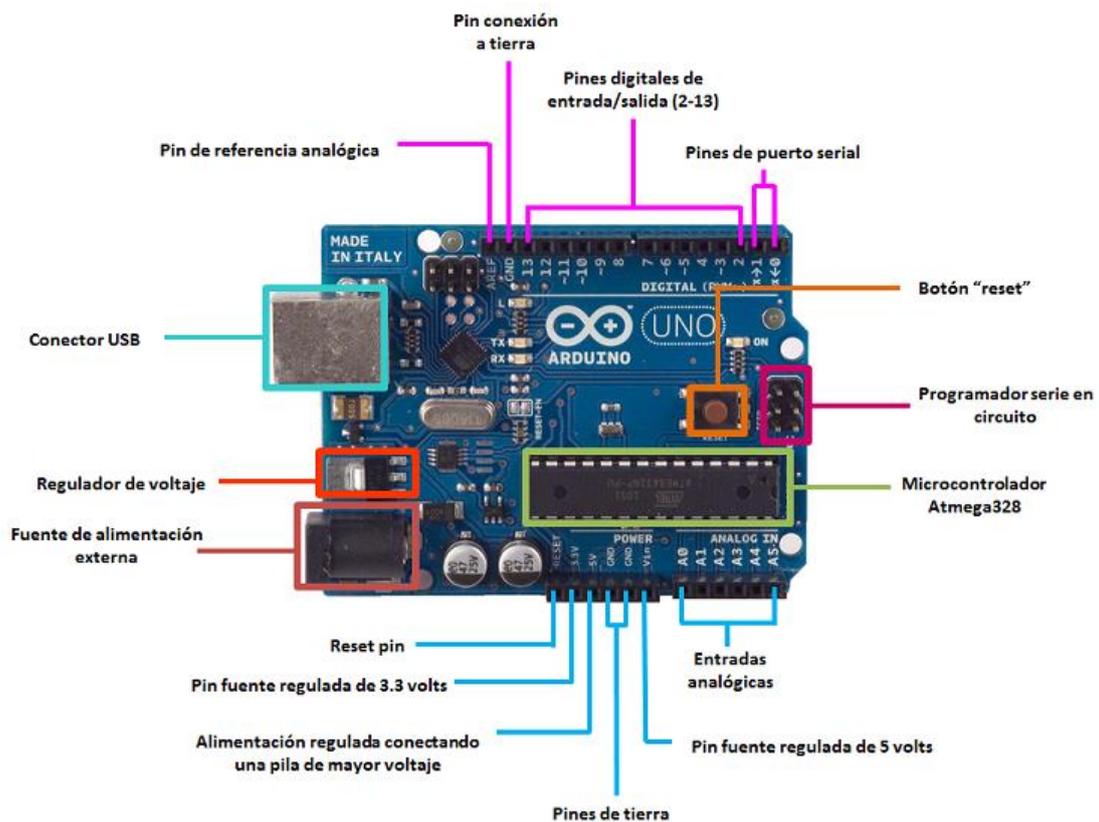


Figura 3 Componentes de Arduino UNO (BLAM Electronics, 2014)

1.5 Introducción a la plataforma Linux (Raspbian OS sobre Raspberry Pi)

Raspberry Pi es un computador de bajo costo y consumo eléctrico además de unas dimensiones muy reducidas. Este equipo tiene una arquitectura de procesador ARM (Broadcom BCM2835) y posee diversas interfaces de hardware para la comunicación y conexión con otros dispositivos.

Es desarrollado y mantenido por *Raspberry Pi Foundation*, diseñado inicialmente por Eben Upton desde 2008, en 2011 los primeros modelos, versión Alfa, fueron producidos y probados. Finalmente, en 2012 sale al mercado de consumo, con dos modelos previstos (A y B). El modelo A, a diferencia del B, no posee conexión Ethernet, además el primero dispone de 256MB de memoria RAM, mientras el segundo tiene 512MB. El consumo eléctrico al igual que el precio del modelo B respecto al A son superiores. (Andrew, 2013, pág. 9)

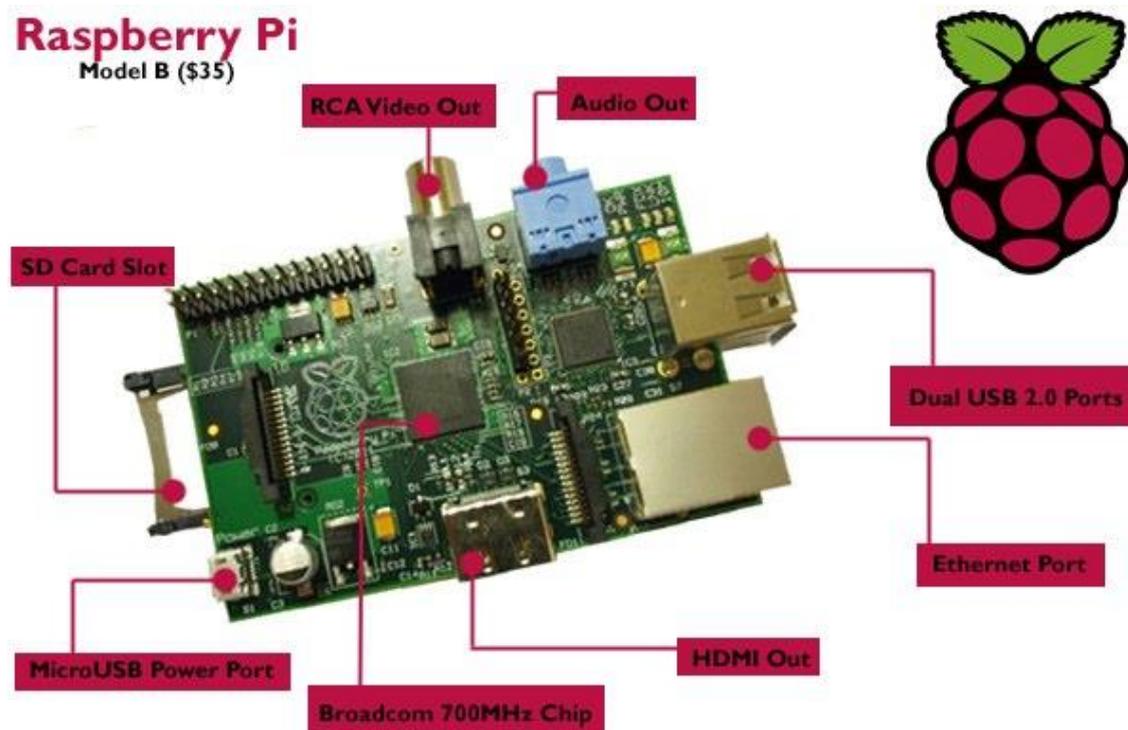


Figura 4 Componentes de Raspberry Pi (Xataka, 2014)

En el computador Raspberry Pi se han incluido varios componentes para soportar diversos modos de comunicación. Básicamente soporta:

- Conexión USB
 - Posee dos puertos USB y un micro USB, en los cuales pueden conectarse prácticamente cualquier dispositivo USB soportado por la plataforma Linux.

- Video RCA
 - Es un conector usado para conexiones de dispositivos audiovisuales, en el caso de Raspberry Pi se usa únicamente un conector para video (Amarillo).
- HDMI
 - El puerto HDMI (del inglés: High Definition Multi-media Interface) permite la conexión con Televisores o monitores de alta definición, adicionalmente soporta audio.
- Audio (Conector de 3.5mm)
 - Este puerto permite la conexión de audífonos o parlantes analógicos.
- Conectividad Ethernet
 - Permite establecer conexiones con dispositivos IP tanto de redes locales o externas.
- Tarjetas SD
 - La tarjeta SD actúa como dispositivo de almacenamiento primario, donde se encuentra instalado el sistema operativo.
- GPIO pins.
 - GPIO pins (del inglés: General Purpose Input/Output pins), es un conjunto de pines digitales de entrada y salida de propósito general, los cuales son programables.

El sistema operativo más común que se encuentra disponible para esta plataforma es *Raspbian OS*. Este SO está basado en Linux-Debian y ha sido optimizado para trabajar con los escasos, pero suficientes, recursos que posee este dispositivo.

La principal función del dispositivo Raspberry Pi dentro de esta investigación es la de actuar como mini servidor. Este proporciona una pasarela que unifica al sistema domótico interno y gestiona la comunicación hacia el exterior mediante la red de datos. Expone servicios web para que otros equipos los consuman o los desarrolladores amplíen la funcionalidad y usabilidad de sistema domótico.

El equipo Raspberry Pi puede ser sustituido por un computador con Linux, pero, se ha preferido este debido a su bajo costo, reducido consumo eléctrico y tamaño físico relativamente pequeño en comparación con la mayoría de computadores.

Capítulo II

2 Definición de la arquitectura

Partiendo del fundamento presentado en el capítulo I, en este se define la arquitectura de la red domótica bajo investigación. En esta sección se especifica el funcionamiento de controladores terminales, dispositivos de control, tipos de circuitos, sensores y la arquitectura de software de controladores centrales.

2.1 Arquitectura de Controladores Terminales (CT)

Los Controladores Terminales (CT) desarrollados sobre la plataforma Arduino UNO poseen pines digitales dedicados para conexiones con actuadores (7 pines), entradas de pulsadores (3 pines), comunicación con la tarjeta Ethernet Shield (4 pines). Los pines analógicos están destinados exclusivamente a entradas de sensores (6 pines). El CT también posee salidas de voltaje para alimentación (GND, 3V, 5V y 9V).

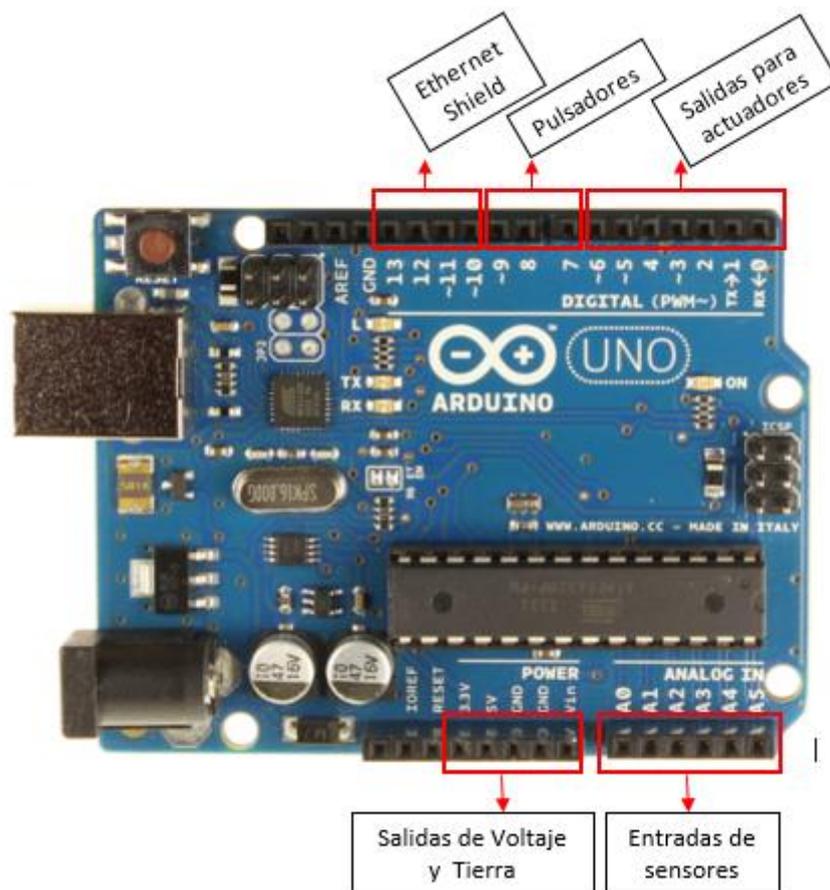


Figura 5 Distribución de los pines en un CT (Arduino UNO)

2.1.1 Configuración y programación de tarjetas Arduino como Controlador Terminal.

Para que el dispositivo Arduino UNO, en conjunto con la tarjeta Ethernet Shield, se convierta en CT se requieren dos pasos:

1. Cargar el *Sketch* “defaultEthernetEEPROM.ino” mediante el software Arduino *IDE*.
 - a. Este archivo contiene instrucciones para restaurar la configuración almacenada en la memoria EEPROM del equipo Arduino UNO a una preestablecida.
2. Cargar el *Sketch* “*RDomoEthernetFirmware*” mediante el software Arduino *IDE*.
 - a. Este posee el firmware para convertir un equipo Arduino UNO con Ethernet Shield en un CT.

Cuando el firmware haya sido cargado, la configuración predeterminada del CT será la siguiente:

- Dirección IP(CT): 192.168.1.177
- MAC Address(CT): DE:AD:C0:A8:01:B1
- Mascara de Subred: 255.255.255.0
- DNS: 192.168.1.1
- Gateway: 192.168.1.1
- IP del Servidor(CC): 192.168.1.11
- PIN de acceso 0-0-0-0

El dispositivo almacena los parámetros de configuración en su memoria EEPROM, la cual es cargada cada vez que el dispositivo se enciende. Un aspecto importante de esta memoria es que posee un límite en el número de escrituras o borrados de 100 000 veces. Por lo tanto, los algoritmos deben tomar en cuenta esta cualidad para realizar la menor cantidad de escrituras posibles y evitar bucles de escritura infinitos. (ATMEL, pág. 1)

Luego de que se haya cargado el firmware y establecido comunicación IP con el CT es posible configurarlo mediante una petición HTTP de tipo GET. La petición tiene una estructura preestablecida, la cual se muestra en la [Tabla 17](#) (Capítulo III).

3. RDomo Ethernet Firmware

2. BootLoader

1. Hardware Arduino UNO

Figura 6 Capas del Controlador Terminal

De las tres capas de la figura anterior, la tercera ha sido desarrollada por el autor de este documento, mientras la primera se refiere al hardware adquirido y la segunda es el software base que posee Arduino para su funcionamiento. De esta forma, la capa que se analiza en este documento es la tercera (3. RDomo Ethernet Firmware).

2.1.1.1 Vista rápida del código fuente del CT.

Para facilitar la comprensión de las secciones posteriores referentes al CT es necesario conocer su funcionamiento. Para entenderlo a nivel técnico se muestra el código fuente, que debido a la extensión del mismo se ha omitido la implementación de las funciones excepto de “*void setup () {}*” y “*void loop () {}*”. El código fuente completo se encuentra en los anexos.

2.1.1.1.1 Librerías usadas:

```
#include <stdio.h>//Standard Input Output
#include <SPI.h>//Interfaz Searial para Perifericos
#include <Ethernet.h>//Conectividad Ethernet(IP)
#include <EEPROM.h>//Usar la memoria EEPROM (Leer o Escribir)
#include <avr/wdt.h>//Reiniciar el dispositivo
```

2.1.1.1.2 Variables globales:

```
byte devType=1;//Identificador del tipo de dispositivo(1 para Ethernet)
byte cfgSize=80;//Tamaño de la configuración en Bytes
double TCVersion=1.0;//Versión del Firmware
unsigned long lastSensorRead=0;//Tiempo de la última lectura de sensores
/*En conjunto, las siguientes dos variables sirven identificar las pulsaciones en el módulo de
pulsadores*/
boolean pushed=0;
boolean pushedButton;
```

```

unsigned long lastPushTime=0;//Tiempo de la última pulsación (pulsadores)
int staticOnPushTime=2000;//Tiempo de pulsación prolongada (Desactiva los sensores del
elemento)
int pushedTime=0;//Cuenta la duración de cada pulsación
IPAddress ip( , , , );//Dirección IP del dispositivo
byte mac[] = { , , , , , };//Dirección MAC
IPAddress DNS( , , , );//Dirección IP del DNS
IPAddress gateway( , , , );//Dirección IP de la puerta de enlace
IPAddress subnet( , , , );//Dirección IP base de la subred
IPAddress serverIP( , , , );//Dirección IP del Servidor(CC)

EthernetServer server(80);//Inicia el servidor http en el Puerto 80

int sensorValue[6];//Almacena los valores actuales de los sensores
int prevSensorValue[6]; Almacena los valores anteriores de los sensores

```

2.1.1.1.3 Función “*void setup()*”: Esta función corre una sola vez en cada reinicio del dispositivo.

```

void setup() {
  //Se establecen los parámetros de red
  Ethernet.begin(mac, ip, DNS, gateway, subnet);

  //Iniciar el servidor TC
  server.begin();

  //Se establecen los pines digitales 0, 1, 2, 3, 4, 5 y 6 como salidas.
  pinMode(0,OUTPUT);
  pinMode(1,OUTPUT);
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);

  /*Se establecen los pines digitales 7, 8 y 9 como entradas para el módulo de pulsadores*/
  pinMode(7,INPUT);
  pinMode(8,INPUT);
  pinMode(9,INPUT);

  lastPushTime=millis();//Se inicializa la variable como pulsación

  /*

```

```

Se construye una petición GET para actualizar el estado de los elementos CC
*/
String url="/api/ws.php?res=updateelementsfromdb&src=tc&idtc=";
url+=ip[0];
url+=".";
url+=ip[1];
url+=".";
url+=ip[2];
url+=".";
url+=ip[3];

//Notifica al servidor que el dispositivo se ha encendido y solicita su último estado.
notifyServer(url);
}

```

2.1.1.1.4 Función “*void loop()*”:

Esta función inicia inmediatamente después de “*void setup()*” y una vez que su ejecución empieza, se repite infinitamente. Con apoyo en las funciones adicionales, se gestiona la comunicación con otros dispositivos (CC), sensores pulsadores y actuadores.

```

void loop() {

//Inicio de análisis de pulsadores
pushedButton=get3PinValue(7); /*Obtiene el valor presente en el bus de
pulsadores*/
if(pushButton>0){// Si existe un botón pulsado
pushedTime=millis()-lastPushTime;
if(pushTime>staticOnPushTime && pushed==101){//Pulsación prolongada de un botón
pushGroupID=EEPROM.read(39+pushedButton);
/*Se construye la petición GET para habilitar o deshabilitar la obediencia a sensores
para el elemento*/
String url="/api/ws.php?res=shiftoseysensor&src=tc&groupid=";
url+=pushGroupID;
url+="&idtc=";
url+=ip[0];
url+=".";
url+=ip[1];
url+=".";
url+=ip[2];
url+=".";
url+=ip[3];

```

```

/*Notifica al servidor para que un elemento obedezca o no a un sensor*/
notifyServer(url);
pushed=102;//para que solo se ejecute una vez cada pulsación prolongada.
}else if(pusheds==0){ //Comprueba que el estado anterior del módulo de pulsadores sea 0
    pushed=1; //Cambia de estado a "presionado"
    lastPushTime=millis();//Tiempo de la última pulsación
}

}else if(pushedsButton==0){ // si el botón ha sido liberado
    pushed=0;//Cambia de estado a "liberado"
}

/*Si su estado es presionado. Distinguir del estado "espera de liberación" (pusheds=101 o
pusheds=102)*/
if(pusheds==1){
    int newVal;//Valor que se establecerá al elemento la pulsación del botón.
    int pushGroupID=EEPROM.read(39+pushedsButton);//Obtiene el id del elemento afectado
    int pushAction=EEPROM.read(46+pushedsButton);//Obtiene la acción del pulsador
    int elemType=EEPROM.read(pushGroupID+25);//Obtiene el tipo de elemento
    if(elemType<128){//los tipos de elemento menores a 18 son de tipo ON/OFF
        newVal=(digitalRead(pushedsButton-1)+1)%2;//cambio de 1 a 0 o viceversa
    }else {//Si es de Incremento o disminución(3 bits o 3 pines)
        newVal=get3PinValue(pushGroupID-1);//Obtiene el valor de los tres pines contiguos
        if(pushAction==3 && newVal<7){//Si la acción es incremento y no se desborda
            newVal++;//Incrementa en uno su valor
        }else if(pushAction==2 && newVal>0){ //Si la acción es disminución y no se desborda
            newVal --;Decremento en uno su valor
        }
    }
}

setValue(pushGroupID, newVal);//Establece el nuevo valor al elemento

//Construcción de la URL para la notificación al servidor (CC).
String url="/api/ws.php?res=setvalue&src=tc&groupid=";
url+=pushGroupID;
url+="&idtc=";
url+=ip[0];
url+=".";
url+=ip[1];
url+=".";
url+=ip[2];
url+=".";
url+=ip[3];

```

```

url+="&value=";
url+=newVal;

notifyServer(url); // Notificación al servidor(CC)

    pushed=101;//Cambio a estado "espera de liberación"
}
//Fin de análisis de pulsadores

//Inicio de notificación de valores de sensores al CC
if((millis()-lastSensorRead)>sensorReadInterval){//Si ha transcurrido el intervalo requerido
    lastSensorRead=millis();//actualización de la última notificación
    String sensorUrl("/api/ws.php/?res=sensorquery&query="+sensorRead());
    notifyServer(sensorUrl); // Notificación de sensores al servidor(CC)
}
//Fin de notificación de valores de sensores al CC

//Inicio de gestión de peticiones http a este CT
EthernetClient client = server.available();
if (client) {//Si existe una petición http
    int resValue=-1;//Variable para obtener el valor recurso solicitado
    int pValue;
    int i=0;
    boolean pinOK=true;//Variable para determinar si el PIN es correcto

    int groupId;
    int groupValue;

    while (client.connected()) {//Mientras exista comunicación con el cliente
        if (client.available()) {//Si existen datos disponibles
            client.findUntil("res", "\n");//Se posiciona luego la palabra "res" de la petición http
            resValue = client.parseInt();//Obtiene el valor del recurso

            //Inicio de envío de cabeceras http al cliente
            client.println("HTTP/1.1 200 OK");
            client.println("Access-Control-Allow-Origin: *");//Se permiten peticiones cruzadas
            client.println("Content-Type: text/html");
            client.println();
            //Fin de envío de cabeceras http al cliente
            switch(resValue){//Analiza el valor del recurso
                case 101://Obtiene la información básica del dispositivo
                    getStateJSON(client, resValue);
                    break;
                case 102://Obtiene la configuración completa del dispositivo en formato JSON

```

```

getFullConfig(client);
break;
case 111://Verificación de PIN (Útil para probar la conexión)
for(i=501;i<=504;i++){
    client.findUntil("-", "\n\r");
    pValue = client.parseInt();
    if(EEPROM.read(i)!=pValue){
        pinOK=false;
    }
}
if(pinOK==true){
    getStateJSON(client, 1001); //Devuelve éxito en formato JSON
}else{
    getErrorJSON(client, 921); //Devuelve error en formato JSON, código 921
}
break;
case 201://Establecer valores a un elemento del CT.
/*Este recurso requiere dos parámetros adicionales luego del recurso
y deben estar separados con un guion medio "-" (Ej: 192.168.0.177/?res=201-1-0)
1) group: El primero identifica al elemento
2) value: El segundo corresponde al valor
*/

client.findUntil("-", "\n\r");//Busca el separador (guion medio "-")
groupId = client.parseInt();//Obtiene el id del elemento
client.findUntil("-", "\n\r");//Busca el separador (guion medio "-")
groupValue = client.parseInt();//Obtiene el valor para el elemento
if(setValue(groupId, groupValue)){//Si se logra establecer el valor
    getStateJSON(client, 1001); //Devuelve éxito en formato JSON
}else{
    getErrorJSON(client, 221); //Devuelve error en formato JSON, código 221
}
break;
case 901://Subir la configuración básica: red, pulsadores, sensores y elementos
{
    for(i=501;i<=504;i++){//Verificación del PIN de acceso
        client.findUntil("-", "\n\r");//Busca el separador(guion medio "-")
        pValue = client.parseInt();//Obtiene el valor
        if(EEPROM.read(i)!=pValue){//comprueba secuencialmente byte por byte el PIN
            pinOK=false;//Si no coincide un byte del PIN
        }
    }
}

client.findUntil("-", "\n\r");//Busca el separador(guion medio "-")

```



```

        break;
    }
case 903://Cambiar solo parámetros de red
for(i=501;i<=504;i++){//Se verifica el PIN de acceso
    client.findUntil("-", "\n\r");//Busca el separador(guion medio "-")
    pValue = client.parseInt();//Obtiene un byte del PIN
    if(EEPROM.read(i)!=pValue){
        pinOK=false;
    }
}

client.findUntil("-", "\n\r");
pValue = client.parseInt();
byte myValue;
if(pinOK && pValue==devType){
    for(i=0;i<26;i++){//Obtiene y escribe los parámetros de red byte por byte
        client.findUntil("-", "\n\r");//Busca el separador (guion medio "-")
        pValue = client.parseInt();//Obtiene un byte del de configuración
        EEPROM.write(i,pValue);//Escribe la configuración la memoria EEPROM
    }
    getStateJSON(client, 1001); //Devuelve éxito en formato JSON
}
break;
case 904://Establece el nombre y descripción del dispositivo
{for(i=501;i<=504;i++){//In this loop, PIN es verified
    client.findUntil("-", "\n\r");
    pValue = client.parseInt();
    if(EEPROM.read(i)!=pValue){
        pinOK=false;
    }
}

client.findUntil("-", "\n\r");
pValue = client.parseInt();

char name[]="?????????";//10 bytes
char description[]="????????????????????????????????????????????????????????????";40 bytes
char myChar;//Variable para obtener los caracteres de la petición GET
if(pinOK && pValue==devType){ //Lee la desde la URL el nombre del CT
    client.findUntil("-", "\n\r");//Devuelve éxito en formato JSON
    for(i=0;i<10;i++){
        myChar = client.read();
        name[i]=myChar;
    }
}

```

```

for(i=0;i<40;i++){//Lee la desde la URL la descripción del CT
    myChar = client.read();
    description[i]=myChar;
}
writeTCDesc(name, description);//Escribe el nombre y descripción del dispositivo
getStateJSON(client, 1001);
}
break;
}

case 905://Establece los nombres y descripciones de los elementos
/*Este recurso es muy similar al anterior (904). Únicamente se agrega un bucle de 7
*repeticiones para escribir cada uno de los nombres de los elementos (foco, AC, etc.)
*Los dos parámetros (nombre, descripción)
*son guardados uno a la vez en cada iteración.
*/
{
    for(i=501;i<=504;i++){//In this loop, PIN es verified
        client.findUntil("-", "\n\r");
        pValue = client.parseInt();
        if(EEPROM.read(i)!=pValue){
            pinOK=false;
        }
    }

    client.findUntil("-", "\n\r");
    pValue = client.parseInt();

    //Name 10 bytes, Description 30 bytes
    char name[]="?????????";
    char description[]="????????????????????????????????????";
    char myChar;
    if(pinOK && pValue==devType){
        int j=0;
        client.findUntil("-", "\n\r");
        for(j=1;j<=7;j++){
            for(i=0;i<10;i++){//Lee el nombre de cada elemento
                myChar = client.read();
                name[i]=myChar;
            }
            for(i=0;i<30;i++){// Lee la descripción de cada elemento
                myChar = client.read();
                description[i]=myChar;
            }
            writeElemDesc(j,name,description);//Graba el nombre y descripción del elemento

```

```

    }
    getStateJSON(client, 1001);
  }
  break;
}
case 999://Reinicia al dispositivo
  getStateJSON(client, 1001);
  delay(200);
  client.stop();
  delay(1000);
  resetMe();//Reinicia al dispositivo
  break;
}
break;
}
}
delay(1);
client.stop();// se detiene el cliente y finaliza la comunicación
//Fin de la gestión de peticiones http a este CT
}
}

```

2.1.1.1.5 Funciones adicionales: se ha omitido la implementación de cada función.

```

/*Verifica el arreglo de configuración*/
boolean configVerifier(byte cfgArray[]){}

```

```

/*Escribe la configuración básica en la memoria EEPROM del CT*/
boolean configWriter(byte cfgArray[]){}

```

```

/*Envía un mensaje en formato JSON con su código en formato JSON*/
void getErrorJSON(EthernetClient &refCli, int errorCode){}

```

```

/*Envía un mensaje en formato JSON como respuesta a una petición http*/
void getStateJSON(EthernetClient &refCli, int action){}

```

```

/*Envía un mensaje en formato JSON con toda la configuración del CT al cliente http*/
void getFullConfig(EthernetClient &refCli){}

```

```

/*Reinicia al CT*/
void resetMe(){}
/*Envía una notificación al servidor (CC)*/

```

```

void notifyServer(String addURL){}

/*Lee los valores de sensores y los devuelve separados por un guion medio "-"*/
String sensorRead(){}

/*Cambia el PIN de acceso al dispositivo y devuelve verdadero si se pudo completar
satisfactoriamente el cambio, caso contrario devuelve falso*/
boolean changePass(byte oldPass[], byte newPass[]){}

/*Escribe el nombre y la descripción del controlador terminal*/
void writeTCDesc(char* name, char* description){}

/*Escribe el nombre y descripción de un elemento especificado en el primer parámetro*/
void writeElemDesc(int idGroup, char* name, char* description){}

/*Establece un valor a un elemento dado y devuelve verdadero en caso de éxito, caso
contrario falso*/
boolean setValue(byte groupId, byte value){}

/*Obtiene la descripción del dispositivo CT*/
String getDevDescription(){}

/*Obtiene el nombre del dispositivo CT*/
String getDevName(){}

/*Obtiene el nombre de un elemento dato*/
String getElemName(int idGroup){}

/*Obtiene la descripción de un elemento dado*/
String getElemDesc(int idGroup){}

/*Obtiene el valor en decimal de tres pines contigua, teniendo como base al parámetro de
entrada*/
int get3PinValue(int pinBase){}

```

2.1.2 Sensores

Cada tipo de sensor debe ser calibrado para obtener una precisión aceptable. Siempre que haya una hoja de datos de un sensor, será posible calibrarlo de acuerdo a sus especificaciones técnicas (variación de resistencia o voltaje en función del fenómeno). En caso de que no exista, una de las formas de calibrarlo es utilizar un instrumento de medición pre calibrado y un medidor de resistencia, medir simultáneamente la resistencia y el valor del fenómeno físico, de esta forma se puede encontrar la relación entre el fenómeno físico y la resistencia. Para mayor información Refiérase a la sección [1.2.1 Sensores](#)

Los sensores son conectados a los pines analógicos de la tarjeta Arduino UNO, la cual posee 6 entradas de este tipo y son capaces de gestionar valores entre 0 y 5 voltios con una precisión digital de 10 bits (valores entre 0 y 1023). Los pines analógicos se han destinado para el uso exclusivo de sensores cuyos valores de voltaje de salida oscilen entre 0 y 5 voltios.

Códigos de tipos de sensores			
Código		Descripción de tipo de circuito	# de pines requeridos
Decimal	Binario		
1	00000001	Sensor de Luminosidad basado en LDR	1
2	00000010	Sensor de Humedad	1
3	00000101	Sensor de Temperatura	1

Tabla 1 Tipos de sensores

Inicialmente, a cada sensor se asocia un elemento, un umbral y una acción (Encender o Apagar) en caso de sobrepaso del umbral. Es recomendable usar el umbral del sensor en un nivel crítico, luego del cual se considere un peligro. En conjunto con el CC es posible configurar acciones avanzadas para cada sensor, pudiendo un solo sensor afectar a múltiples elementos y a cada elemento asociar distintas acciones en función del rango en el que se encuentre el valor del sensor, esta característica funciona únicamente cuando el CC está activo y exista comunicación con los CT.

Los circuitos de *control de acceso* requieren que siempre exista un sensor o dispositivo fin carrera, el cual cuando la puerta está cerrada debe enviar al CC un nivel alto (1), caso contrario, si la puerta se encuentra abierta un nivel bajo (0).

Los circuitos de *climatización* requieren que siempre exista un sensor de temperatura, actualmente el sistema es capaz de gestionar un solo sensor por cada

circuito de este tipo, de esta forma, si existen n circuitos de climatización se necesitarán n sensores de temperatura, cada uno asociado a un y solo un circuito.

2.1.3 Actuadores

Dentro de esta investigación se consideran dos tipos de actuadores *Encendido/Apagado* e *Incremento/Disminución*. Los pines destinados en el CT para la comunicación con los actuadores van desde el pin 0 hasta el 6, que dentro del software de control estarán mapeados de 1 a 7. A estos pines pueden colocarse cualquiera de los dos tipos de circuitos.

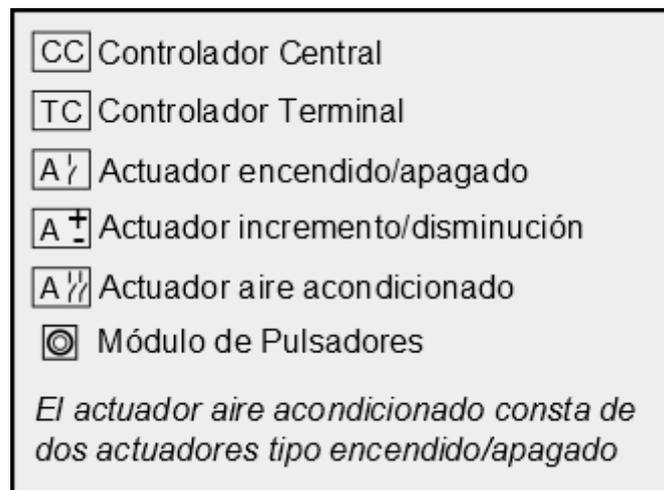


Figura 7 Simbología elementos domóticos.

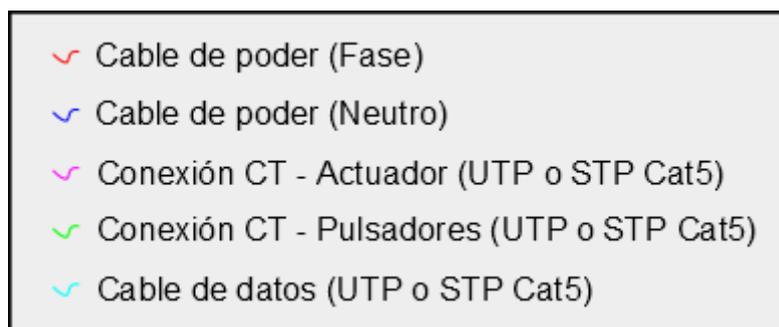


Figura 8 Colores de cables

2.1.3.1 Encendido / Apagado

Los actuadores de este tipo disponen de un relé, que se encarga de integrar los circuitos de Corriente Continua y Corriente Alterna sin que existan daños en los equipos de bajo voltaje. Este dispositivo al ser activado mediante una señal desde el

CT abre o cierra el circuito mediante un electroimán. Para este tipo de actuador se usa un pin digital del CT.

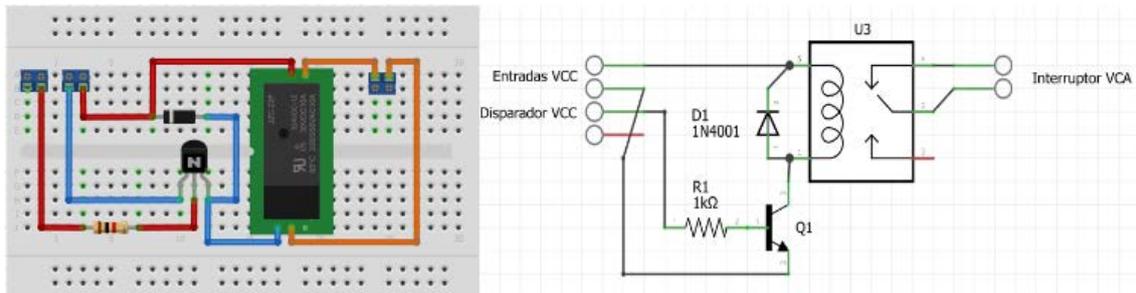


Figura 9 Diagrama del actuador de tipo Encendido/Apagado

En la figura anterior se puede observar que existen dos entradas de voltaje continuo para alimentar al relé y un disparador para activar o desactivar al relé. Cuando el disparador esté en un nivel alto (1) el relé se activará y cerrará el circuito CA, caso contrario si el disparador se encuentra en nivel bajo (0) el relé se desactiva y se desconecta el circuito de CA.

2.1.3.2 Incremento / Disminución

Estos tipos de circuitos permiten regular la intensidad con la que se presenta un determinado fenómeno. Esta investigación se limita al incremento o disminución de luminosidad en 8 niveles (0-7), el efecto se logra mediante la modulación de onda de la corriente alterna, para lo cual se usan triacs, diacs, opto-acopladores y resistencias. Este tipo de circuito requiere de 3 pines digitales contiguos para controlar los 8 ($2^3 = 8$) niveles de iluminación. 0 es completamente apagado y 7 es totalmente encendido.

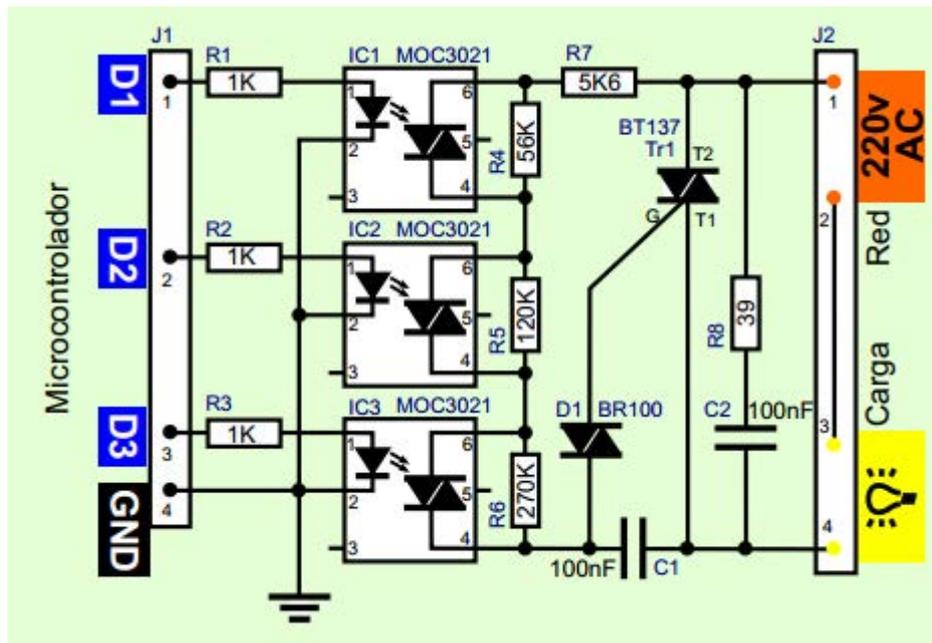


Figura 10 Diagrama del circuito de Regulación de Intensidad (Txapuzas, 2014)

En la figura anterior, el corte de la onda sinusoidal viene dada por la suma de las resistencia R4, R5 y R6, pero, cuando el opto acoplador (IC1, IC2 o IC3) se active (D1, D2 o D3 en nivel alto) permitirá el paso de corriente sin resistencia, anulando de esta forma a la resistencia conectada a IC activado. A menor resistencia total, mayor será la iluminación (No existirá corte de la onda) mientras que a la mayor resistencia posible, la iluminación se apagará completamente.

2.2 Dispositivos de Control o interacción con el sistema

Los dispositivos de control permiten que un sistema domótico sea gestionado de manera sencilla, de tal forma que un usuario pueda comenzar a utilizarlo sin que exista una curva de aprendizaje prolongada.

2.2.1 Control Físico – Pulsadores

Es conveniente que siempre exista una alternativa de control manual del sistema domótico. Manteniendo, de cierta forma, un mecanismo de control tradicional. Por lo expuesto, cada CT dispone de un bus para pulsadores que está compuesto por 3 bits, enlazados a los pines digitales 7, 8 y 9. Siendo el bit más significativo el correspondiente al pin 7 y el menos significativo al 9.

Cada botón se asocia con un grupo de circuito y una acción, las acciones puede ser de tipo: Encendido/Apagado, Incremento y finalmente Disminución.

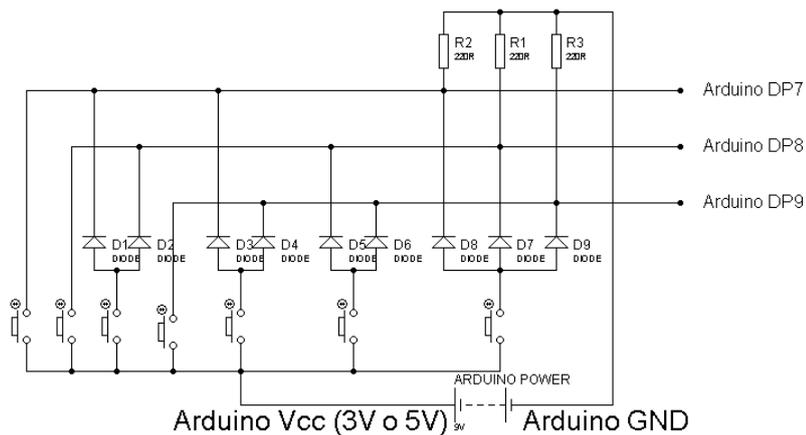


Figura 11 Esquema del circuito de pulsadores

De acuerdo a la figura anterior, cuando el bus esté inactivo, el valor presente en el mismo será 000. Al presionar un pulsador se envía una señal en binario (001 - 111). Para evitar la retroalimentación de corriente hacia otros pines, se utilizan diodos rectificadores, los cuales permiten el paso de corriente en una sola dirección. Se usan tres resistencias como divisores de corriente.

2.2.2 Control por Software

Para el control por software, el CC es el dispositivo encargado de gestionar la comunicación con los CTs y hacia el exterior (con otros dispositivos), actuando a modo de pasarela residencial. Las órdenes se envían a los CTs mediante la URL, y como respuesta se obtiene un mensaje en formato JSON. Una descripción detallada sobre este tipo de control es expuesta en el capítulo III.

2.3 Circuitos Soportados

Básicamente se ha dividido en tres tipos de circuitos primarios: encendido/apagado, Climatización e incremento/disminución. Una lista completa de los circuitos soportados, con su respectivo código y número de pines requeridos para su funcionamiento se encuentran en la siguiente tabla.

Código		Descripción de tipo de circuito	# de pines requeridos
Decimal	Binario		
1	0000001	Circuito Encendido/Apagado de propósito general	1
2	0000010	Circuito Encendido/Apagado de Iluminación.	1
3	0000101	Climatización	2
4	0000100	Riego (Encendido / Apagado)	1
5	0000101	Control de acceso	1
129	1000001	Circuito Incremento/Disminución de Iluminación	3

Tabla 2 Circuitos soportados

Todos los circuitos de tipo Encendido/Apagado funcionan bajo el mismo principio, la única razón por la que se los ha separado en varios tipos es para mantener una clasificación de los elementos conectados. Todos los circuitos basados en este tipo utilizan el mismo tipo de actuador (actuador Encendido/Apagado).

El control de Acceso, si bien es de tipo Encendido/Apagado, es el único que tiene una variación en su funcionamiento. Este, al ser encendido, permanece activo por 1 segundo, luego del cual se apaga, enviando simplemente un pulso al actuador. Se identifica si la puerta está abierta o cerrada mediante un pulsador (fin carrera).

Los circuitos de Incremento/Disminución de Iluminación funcionan con un actuador capaz de gestionar 8 estados (3 bits). El bit más significativo corresponde al pin de mayor valor nominal, esto es, si un circuito de este tipo utiliza los pines 0, 1 y 2, el bit más significativo corresponde al pin 2 y el menos significativo al pin 0.

2.4 Arquitectura de Software

El CC está compuesto por una base de datos en MySQL, un paquete de software desarrollado en PHP y librerías de código abierto para PHP (cURL y nmap).

2.4.1 Configuración y programación de Raspberry Pi como Controlador Central

Para que un dispositivo Raspberry Pi pueda actuar como CC, es necesario instalar un servidor LAMP (Linux, Apache, MySQL y PHP), el software nmap, la librería cURL (Client URL) para PHP, un servidor FTP (vsftpd) y cargar al dispositivo el software desarrollado.

2.4.1.1 Instalación y configuración la plataforma base.

A continuación se muestra una guía detallada sobre la instalación y configuración Raspbian OS con el paquete NOOBS, con todo el software base requerido para la finalidad de este proyecto, de tal forma que se obtenga un entorno funcional con Raspberry Pi, que posea:

- Servidor web con soporte para php (Apache).
- Servidor de base de datos (MySQL).
- Cliente URL desde php (cURL).
- php PEAR (Paquete Mail)
- Herramienta para exploración de redes TCP/IP (nmap).

2.4.1.1.1 Prerrequisitos

1. Equipo Raspberry Pi
2. Tarjeta SD(mínimo 4GB)
3. Descargar NOOBS_v1_3_4.zip desde:
<http://www.raspberrypi.org/downloads/>
4. Nota: RaspbianOS y Raspberry Pi pertenecen a organizaciones independientes la una de la otra.
5. Descargar SD Formatter 4.0 desde:
https://www.sdcard.org/downloads/formatter_4/

2.4.1.1.2 Preparación

1. Formatear la tarjeta SD, usando SD Formatter 4.0.
2. Extraer todos los archivos de NOOBS_v1_3_4.zip dentro de la tarjeta SD.
3. Insertar la tarjeta SD en la ranura de Raspberry Pi.
4. Conectar un teclado y un ratón.
5. Conectar una salida de video(RCA o HDMI)
6. Conectar a la fuente de poder.

2.4.1.1.3 Instalación de SO

1. Seleccionar Raspbian OS de la lista de sistemas operativos disponibles.
2. Seleccionar el lenguaje preferido (Para este caso se ha mantenido el predeterminado - Ingles UK).
3. Seleccionar la distribución de teclado (Se utiliza un teclado de distribución española, por lo tanto - es).
4. Clic en el botón Install.
5. Esperar que la instalación termine.

Nota: Cada vez que inicie el sistema, este se cargará en modo de línea de comandos (CLI-Command Line Interface), para poder usar el modo gráfico (GUI – Graphical User Interface) se debe ejecutar el comando **startx**. Los datos predeterminados para el inicio de sesión son:

Usuario: pi

Contraseña: raspberry

2.4.1.1.4 Configurar los parámetros de Red

Abrir una sesión de terminal

1. Editar el archivo `/etc/network/interfaces` (`sudo nano /etc/network/interfaces`). El fichero debe contener algo similar al siguiente texto:

```
iface eth0 inet static
address 192.168.1.11
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
dns-nameservers 192.168.1.1
```
2. Reiniciar el servicio de red
 - a. `sudo /etc/init.d/networking restart`.

Nota: Una vez que se posea conectividad IP, se pueden realizar conexiones al Raspberry Pi mediante ssh (preinstalado), una herramienta muy útil para esto es Putty. Para acceder por escritorio remoto (*RDP-Remote Desktop Protocol*) se requiere instalar xrdp (`sudo apt-get install xrdp`) o algún software similar, de esta forma es posible tener control total del dispositivo mediante un entorno gráfico, sin la necesidad de un monitor físico.

2.4.1.1.5 Instalación de software base.

Una vez establecida la conexión de red y teniendo acceso a internet se debe instalar el siguiente software:

1. Actualizar la lista de repositorios y el software.
 - a. `sudo apt-get update`
 - b. `sudo apt-get upgrade`
2. Instalar Apache 2
 - a. `sudo apt-get install apache2`
 - b. Se puede comprobar su funcionamiento accediendo mediante un navegador web con la dirección IP como URL.
3. Instalar PHP 5
 - a. `sudo apt-get install php5`
4. Instalar MySQL Server
 - a. `sudo apt-get install mysql-server`

5. Instalar conector php-mysql
 - a. `sudo apt-get install php5-mysql`
6. Instalar cURL para php
 - a. `sudo apt-get install php5-curl`
7. Instalar php PEAR
 - a. `Sudo apt-get install php-pear`
8. Instalar el paquete mail
 - a. `Sudo apt-get install php-mail`
9. Instalar librería GD para php
 - a. `sudo apt-get install php5-gd`
10. Instalar nmap para el mapeo de la red
 - a. `sudo apt-get install nmap`
11. Instalar servidor FTP
 - a. `sudo apt-get install vsftpd`
12. Editar el archivo de configuración `/etc/vsftpd.conf`
 - a. `sudo nano /etc/vsftpd.conf`
 - b. Cambiar `anonymous_enable=YES` por `anonymous_enable=NO`
 - c. Quitar comentario de la línea `#local_enable=YES`, esto permitirá que se pueda iniciar sesión con los usuarios locales del sistema.
 - d. Quitar comentario de la línea `#write_enable=YES`. Al quitar e comentario se permite la escritura en los directorios permitidos al usuario.

2.4.1.1.6 Configuración recomendada durante el desarrollo

Durante la etapa de desarrollo de software, frecuentemente se requiere acceder y gestionar remotamente la base de datos del CC. Para habilitarlo es necesario realizar algunas modificaciones en el archivo de configuración del dispositivo y crear usuarios con privilegios de conexión remota.

- Editar el archivo: `sudo nano /etc/mysql/my.cnf`
 - Buscar una línea similar a:
 - `bind-address = 127.0.0.1`
 - Reemplazar por:
 - `bind-address = 0.0.0.0`(Todas las direcciones estarán permitidas)
- Crear un usuario que pueda conectarse desde cualquier lugar.
 - Acceder a mysql mediante la consola(Proporcionar los datos de usuario)
 - `mysql -u root -p`

- Escribir la sentencia:
 - `grant all privileges on *.* to rdomo@'%' identified by 'rdomo';`

Nota: Esta configuración no es aconsejable para ambientes de producción, debido a que se agrega un puerto abierto a conexiones externas.

2.4.1.1.7 Carga del software desarrollado

El software del CT está escrito en el lenguaje de programación PHP para el motor de la aplicación, mientras la interfaz está desarrollada con JavaScript, HTML y CSS. Cuando se hayan cumplido con todos los pasos anteriores será posible cargar el software desarrollado, el método más sencillo es mediante el protocolo FTP (*del Inglés – File Transfer Protocol*). Una herramienta que resulta útil para esta tarea es FileZilla, en esta se debe especificar la dirección IP, el nombre de usuario y la contraseña de acceso al servidor. Luego de esto se deben subir los archivos al directorio raíz del servidor Apache (generalmente `/var/www`).

La base de datos es posible crearla a partir del modelo Entidad-Relación, adjunto en el disco, mediante la herramienta MySQL Workbench. Para esto se debe hacer clic en el menú “Database” y luego en el submenú “Forward Engineer”. A continuación, mediante un asistente, debe completarse la información solicitada en todas sus etapas. Si la información del servidor, usuario y contraseña proporcionados es correcta, el sistema estará listo para funcionar.

2.4.2 Estructura de Base de Datos (DB - Data Base)

La Base de Datos para el sistema domótico consta de las siguientes tablas:

actions	parameters	tc
element_logs	push_buttons	tc_sensors
element_types	sensor_types	user_types
elements	subnets	users
elements_sensors		

Tabla 3 Lista de tablas de la Base de Datos

2.4.2.1 Diagrama Entidad-Relación

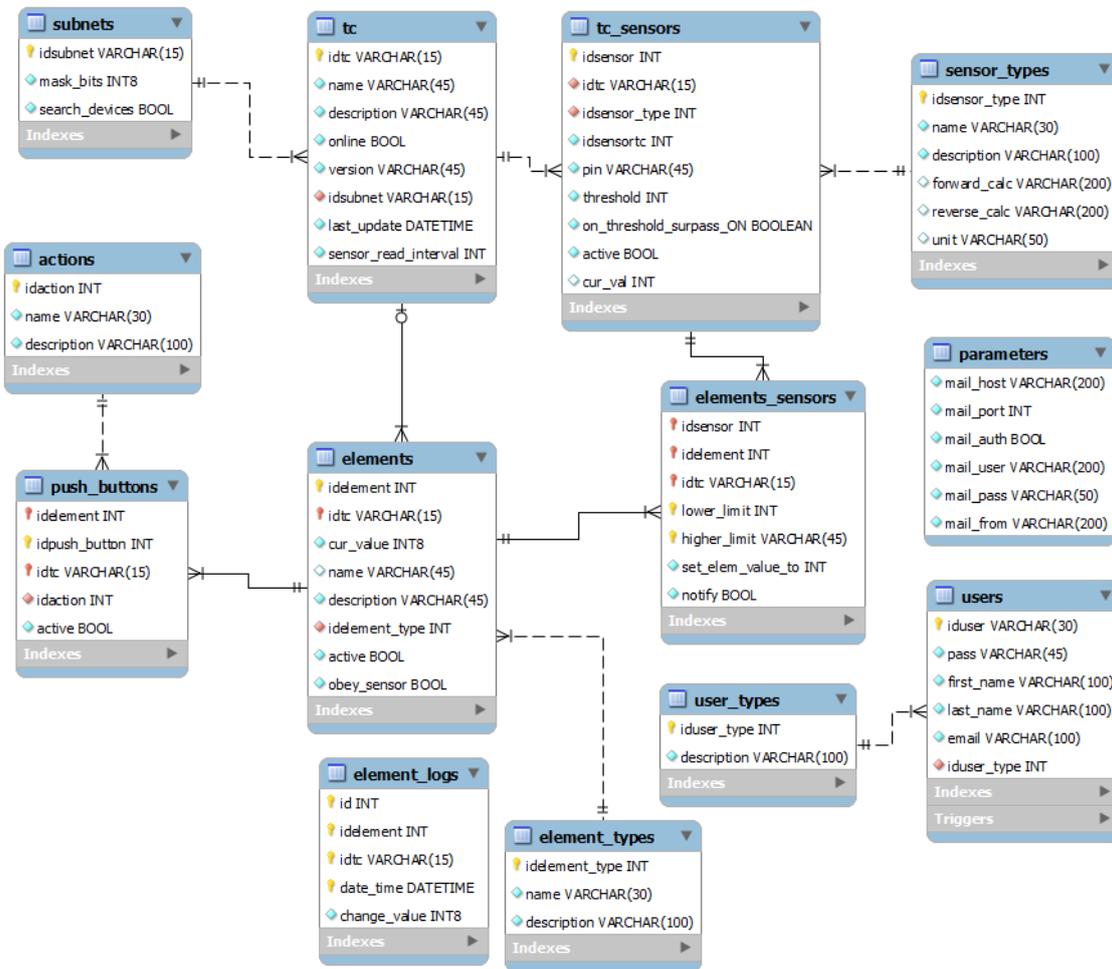


Figura 12 Diagrama Entidad-Relación de la Base de Datos

2.4.2.2 Diccionario de Base de Datos:

actions										
Acciones de los pulsadores										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
idaction	INT	✓	✓							Identificador de la acción
name	VARCHAR(30)		✓							Nombre de la acción (ON, OFF, +/-)
description	VARCHAR(100)		✓							Descripción de la acción

Tabla 4 DB. Tabla actions

element_logs										
Registro de cambios de estado de los elementos. Esta tabla no esta relacionada para evitar inconsistencias en casos de eliminacion de elementos o CTs.										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
id	INT	✓	✓					✓		id secuencial
idelement	INT	✓	✓							Identificador del elemento dentro del CT
idtc	VARCHAR(15)	✓	✓							Identificador del Controlador Terminal
date_time	DATETIME	✓	✓							Fecha y hora de la acción
change_value	BIGINT(20)		✓							Valor del elemento

Tabla 5 DB. Tabla element_logs

element_types										
Tipos de elementos										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
idelement_type	INT	✓	✓							Identificador del tipo de elemento
name	VARCHAR(30)		✓							Nombre del tipo de elemento
description	VARCHAR(100)		✓							Descripción del tipo de elemento

Tabla 6 DB. Tabla element_types

elements										
Elementos conectados a los CT2										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
idelement	INT	✓	✓							Identificador del elemento conectado al CT (en el CT se denomina GroupID)
idtc	VARCHAR(15)	✓								Identificador del CTal cual pertenece el elemento
cur_value	BIGINT(20)		✓							Valor actual del elemento
name	VARCHAR(45)									Nombre del elemento
description	VARCHAR(45)		✓							Descripción del elemento
idelement_type	INT		✓							Identificador del tipo de elemento
active	TINYINT(1)		✓						1	Indica si el elemento está activo
obey_sensor	TINYINT(1)		✓						1	Si el elemento obedece a los sensores asociado será verdadero, caso contrario falso.

Tabla 7 DB. Tabla elements

elements_sensors										
Un sensor de un CT puede asociarse a uno o varios elemento de un CTs distintos y pueden establecerse diferentes rangos y acciones para cada uno.										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
idsensor	INT	✓	✓							Identificador del sensor
idelement	INT	✓	✓							Identificador del elemento
idtc	VARCHAR(15)	✓	✓							Identificador del CT
lower_limit	INT	✓	✓							Límite inferior
higher_limit	VARCHAR(45)	✓	✓							Límite superior
set_elem_value_to	INT		✓							0 Cuando la lectura del sensor este entre el límite inferior y el superior se establece este valor al elemento
notify	TINYINT(1)		✓							0 Indica si se debe notificar cuando el sensor se encuentre en este rango.

Tabla 8 DB. Tabla element_sensors

parameters										
Parámetros										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
mail_host	VARCHAR(200)		✓							Dirección del host de correo electrónico
mail_port	INT		✓							Puerto del servidor de correo electrónico
mail_auth	TINYINT(1)		✓							
mail_user	VARCHAR(200)		✓							Usuario(Correo)
mail_pass	VARCHAR(50)		✓							Contraseña del usuario(Corresponde al correo electrónico)
mail_from	VARCHAR(200)		✓							Dirección de correo electrónico saliente

Tabla 9 DB. Tabla parameters

push_buttons										
Pulsadores asociados a los elementos										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
idelement	INT	✓	✓							Identificador del elemento al cual pertenece el pulsador.
idpush_button	INT	✓	✓							Identificador de pulsador (1-7) para la VI
idtc	VARCHAR(15)	✓	✓							Identificador del Controlador Terminal
idaction	INT		✓							Identificador de la acción
active	TINYINT(1)		✓						1	Si un botón está activo

Tabla 10 DB. Tabla push_buttons

sensor_types										
Tipos de sensores										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
idsensor_type	INT	✓	✓							Identificador del tipo de sensor
name	VARCHAR(30)		✓							Nombre del tipo de sensor
description	VARCHAR(100)		✓							Descripcion del tipo de sensor
forward_calc	VARCHAR(200)									Conversion de unidades reales a niveles de voltaje
reverse_calc	VARCHAR(200)									Conversion de niveles de voltaje a unidades reales
unit	VARCHAR(50)									Unidad de medida

Tabla 11 DB. Tabla sensor_types

subnets										
Subredes en las cuales se encuentran Dispositivos domóticos RDom. Si se implementa únicamente dentro de una red conmutada (Switch o Bridge) solo se requiere una subred.										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
idsubnet	VARCHAR(15)	✓	✓							Se refiere a la base de la red. Ej. Para un host 192.168.0.34/24, la subred es 192.168.0.0
mask_bits	BIGINT(20)		✓							Tamaño de la máscara de la subred (0-32), para limitar las búsquedas, se recomienda usar mascarar mayores a 24 bits.
search_devices	TINYINT(1)		✓						1	Indica al sistema si debe buscar actuadores dentro de esta subred

Tabla 12 DB. Tabla subnets

tc										
Un Controlador Terminal (TC) es un dispositivo RDom desarrollado en la plataforma de Hardware abierto Arduino.										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
idtc	VARCHAR(15)	✓	✓							Identificador del controlador Terminal, en este caso la IP V4 perteneciente al dispositivo.
name	VARCHAR(45)		✓						"	Nombre asignado al actuador.
description	VARCHAR(45)		✓						"	Descripción
online	TINYINT(1)		✓							Determina si el CT está encendido.
version	VARCHAR(45)		✓							Versión del CT, sirve para compatibilidad en futuros desarrollos.
idsubnet	VARCHAR(15)		✓							Subred a la cual pertenece el CT
last_update	DATETIME		✓							Última actualización realizada al CT y notificado al servidor.
sensor_read_interv	INT		✓						255	Intervalo de lectura de sensores

Tabla 13 DB. Tabla tc

tc_sensors										
Sensores que posee un CT										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
idsensor	INT	✓	✓					✓		Identificador del sensor
idtc	VARCHAR(15)		✓							Identificador del controlador terminal al cual esta conectado el sensor
idsensor_type	INT		✓							Identificador del tipo de sensor
idsensortc	INT		✓							CT al cual pertenece el sensor
pin	VARCHAR(45)		✓							Pin al cual debe conectarse el sensor
threshold	INT		✓							Umbral para acción critica.
on_threshold_surp	TINYINT(1)		✓							1 Si es verdadero, en caso de sobrepaso del umbral, se enciende el elemento afectado.
active	TINYINT(1)		✓							1 Si el sensor está activo
cur_val	INT									Valor actual del sensor

Tabla 14 DB. Tabla tc_sensors

user_types										
Tipos de usuarios del sistema										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
iduser_type	INT	✓	✓							Identificador del tipo de usuario
description	VARCHAR(100)		✓							Descripción del Tipo de usuario

Tabla 15 DB. Tabla user_types

users										
Usuarios del sistema domotico										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	efau	Comment
iduser	VARCHAR(30)	✓	✓							Nombre de usuario
pass	VARCHAR(45)		✓							Contraseña de acceso as sistema para el usuario.
first_name	VARCHAR(100)		✓							Nombres del usuario
last_name	VARCHAR(100)		✓							Apellidos del usuario
email	VARCHAR(100)		✓							Correo electrónico del usuario
iduser_type	INT		✓							Id del Tipo de usuario asociado

Tabla 16 DB. Tabla users

2.4.3 Estructura de directorios y archivos.

Nombre	Función
 api	Contenedor de la API
 ws.php	Este archivo se encarga de la gestión de todos los servicios web de la API. Los servicios web difieren unos de otros en el recurso solicitado y sus parámetros. Ejemplo: <a href="http://<IP>/api/ws.php?res=recursoSolicitado&Parametro1=valor1&...&ParametroN=valorN">http://<IP>/api/ws.php?res=recursoSolicitado &Parametro1=valor1&...&ParametroN=valorN
 config	Directorio de la aplicación de configuración de CTs
 css	Contenedor de hojas de estilo
 config.css	Hoja de estilos para la configuración de CTs.
 img	Imágenes usadas en la página de configuración
 js	Archivos JavaScript
 config.js	Aplicación en JavaScript para la configuración.
 jquery-2.1.0.min.js	Librería JQuery.
 jquery-validate.js	Variación de datos con JQuery
 index.php	Archivo principal para la configuración de CTs.
 core	Directorio donde se encuentra la inteligencia del sistema.
 devices.php	Clases para la gestión de CT (autodescubrimiento, establecer valores,
 events.php	Manejo de
 misc.php	Misceláneos
 notify.php	Envío notificaciones por correo electrónico.
 parameters.php	Gestión de parámetros de correo electrónico saliente.
 privileges.php	Manejo de los permisos que tienen los usuarios del sistema.
 sensors.php	Manejo de sensores

 sql.php	Gestión de la comunicación con la Base de Datos.
 subnets.php	Manejo de las subredes establecidas para la red doméstica.
 users.php	Gestión de Usuarios del sistema
 css	
 general.css	Hoja de estilos del sitio web
 img	Imágenes para el sitio principal
 interface	
 home.php	Página principal.
 menu.php	Menú principal.
 js	Contenedor JavaScript
 ui	Librería “JQuery User Interface”
 site.js	Archivo JavaScript para la aplicación web.
 jquery-2.1.0.min.js	Librería JQuery.
 jquery-validate.js	Variación de datos con JQuery
 index.php	Gestiona el sitio web. Todas y cada una de las peticiones son analizadas en este y de acuerdo al recurso solicitado realiza las acciones correspondientes. Ejemplo: <code>http://<IP>/index.php?res=recursoSolicitado &Parametro1=valor1&...&ParametroN=valorN</code>

Tabla 17 Directorios y archivos del software del CT

2.4.4 API (Application Programming Interface) para desarrolladores

Para que otros desarrolladores puedan ampliar la funcionalidad de control del sistema, se presenta una colección de clases desarrolladas en PHP, prestando de esta forma un alto nivel de abstracción del sistema y facilitando el desarrollo de complementos como aplicaciones web, móviles o de escritorio.

El código fuente de la aplicación y el modelo entidad relación se encuentran en un anexo digital.

Capítulo III

3 Comunicación

La comunicación siempre se realiza entre dos dispositivos o elementos, dependiendo de estos últimos pueden existir varios tipos. Entre el CC y los CTs se realiza mediante el intercambio de mensajes HTTP (Hypertext Transfer Protocol), sobre el protocolo de red TCP/IP. Los CTs se comunican directamente con los sensores y actuadores, leyendo o estableciendo niveles de voltaje en sus pines.

3.1 TCP/IP

Cada dispositivo, tanto CC como CT, poseen su propia dirección IP, dirección MAC, máscara de subred, puerta de enlace, dirección IP del servidor DNS y dirección IP del CC al cual notificarán sus acciones. Dado que el protocolo TCP/IP puede funcionar sobre varios medios físicos, es posible utilizar en sus conexiones intermediarias radio enlaces, redes por cable de cobre o fibra óptica, enlaces ópticos inalámbricos o cualquier otro que sea soportado por TCP/IP.

El sistema es capaz de funcionar dentro de redes tanto ruteadas como conmutadas o una combinación de las dos. En caso de redes ruteadas, la puerta de enlace de cada CT debe estar configurada correctamente, además, la dirección IP del CC ni las de los CTs deben ser modificadas al salir del enrutador (*NAT* - Network Address Translation), si esto ocurre, es posible que el equipo sea inalcanzable y por lo tanto no exista comunicación entre ellos.

3.2 Servicios Web

La comunicación entre el CC y los CTs se realiza mediante servicios web, intercambiando mensajes HTTP por el método GET. La mayoría de respuestas del CT son codificadas en formato JSON. Los dos tipos de dispositivos actúan, a su debido tiempo, como servidor o cliente.

3.2.1 Controlador Terminal (CT)

En cada iteración de la función `void loop()` se analiza si existen peticiones HTTP al CT y luego se verifica si se requiere enviar notificaciones al CC. Si es una petición al CT, se analiza el recurso solicitado y sus parámetros, luego de lo cual se responde en función de estos. Siempre que el CT requiera enviar notificaciones al CC actuará como cliente HTTP, en cuyo caso la respuesta del CC no es analizada. Existen notificaciones al CC que, luego de ser analizadas, pueden desencadenar

cambios de estado en uno o varios CTs, en cuyo caso el CC se inicia nuevas peticiones HTTP a los CT afectados.

3.2.1.1 Recepción de órdenes y peticiones.

Las ordenes al CT pueden provenir desde los pulsadores o el CC. Si la orden procede del CC, el CT recibirá una URL que deberá ser analizada para determinar la acción a realizar sobre los actuadores, mientras que, si la orden proviene desde el módulo de pulsadores, el CT debe aplicar la acción respectiva al actuador correspondiente y luego notificar al CC mediante una petición HTTP.

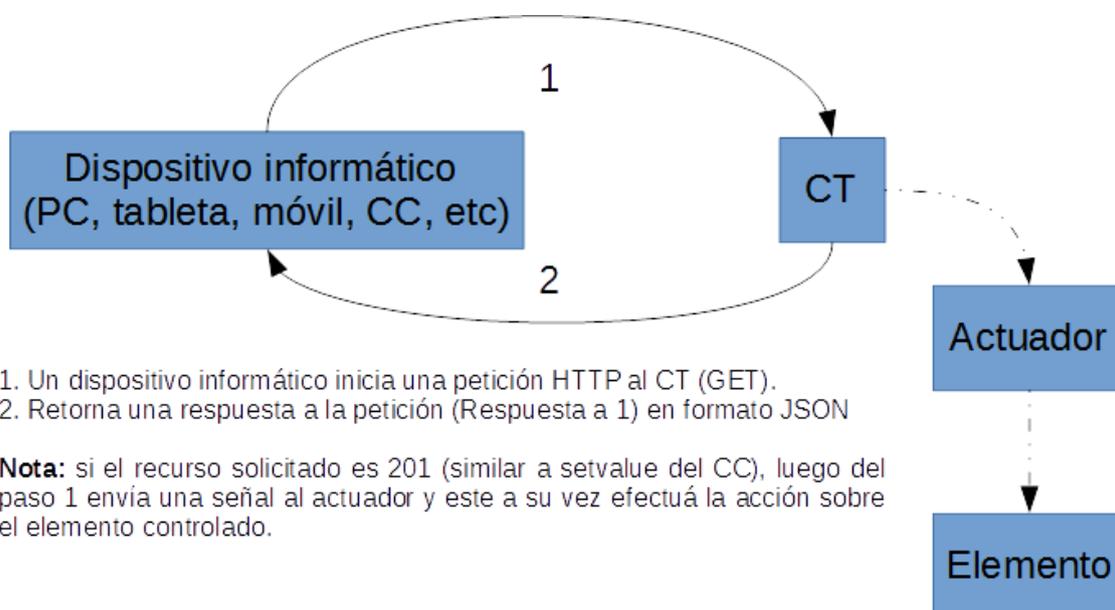


Figura 13 Esquema de comunicación **Cliente => CT [=> Actuador] => Cliente**

3.2.1.1.1 Estructura de peticiones HTTP GET al CT

Dentro de la función `void loop(){}` del código fuente se puede observar que existe una sección abarcada dentro de la instrucción `switch(resValue){...}`, en la cual se analiza el valor de cada caso (recurso o `res`), se realiza una determinada acción según su valor y se devuelve una respuesta.

3.2.1.1.1 case 102:

Obtiene la configuración del CT en formato JSON, con la siguiente estructura:

Parámetro	Valor/es	Tipo	Descripción
{ "device" :			
"success":	true,	Booleano	Identifica que la petición ha sido servida exitosamente
"TCname":	"Nombre del dispositivo",	Texto	Nombre global del CT
"TCdescription" :	"Descripcion del dispositivo",	Texto	Descripción global del CT
"TCtype":	"RDomo",	Texto	Tipo de Dispositivo
"TCversion":	"1.00",	Texto	Versión del CT
"ip":	"192.168.1.177",	Texto	IP del CT
"mac":	"222-173-192-168-1-177",	Texto	Dirección MAC del CT
"dns":	"192.168.1.1",	Texto	Dirección IP del servidor DNS
"gateway":	"192.168.1.1",	Texto	Puerta de enlace del CT
"netmask":	"255.255.255.0",	Texto	Mascara de subred del CT
"serverIp":	"192.168.1.11",	Texto	Dirección IP del Servidor (CC)
"cType":	[#, #, #, #, #, #],	Arreglo Entero	Arreglo en JSON con los valores correspondientes al tipo de circuito, el arreglo esta ordenado de acuerdo al número de pin en el CT.
"groupId":	[#, #, #, #, #, #],	Arreglo Entero	Existen circuitos compuestos, que usan más de un pin digital para su funcionamiento, entonces, en este arreglo se identifica a que grupo pertenece cada pin.
"elemName":	["Nombre1", "Nombre2", "Nombre3", "Nombre4", "Nombre5", "Nombre6", "Nombre7"],	Arreglo Texto	En este arreglo se encuentran los nombres de cada uno de los elementos
"elemDesc":	["Descripcion1", "Descripcion2", "Descripcion3", "Descripción4", "Descripcion5", "Descripcion6", "Descripcion7"],	Arreglo Texto	Descripciones de los elementos

"groupVal":	[#, #, #, #, #, #, #, #],	Arreglo Entero	Valores de cada pin digital, en caso de que sea un circuito compuesto debe calcularse el valor del grupo en decimal.
"pAsoc":	[#, #, #, #, #, #, #, #],	Arreglo Entero	Cada valor de este arreglo identifica la que grupo afectado por una pulsación en un determinado botón.
"pAction":	[#, #, #, #, #, #, #, #],	Arreglo Entero	Se identifica la acción que desencadena la pulsación de un botón(ON/OFF, -, +)
"sType":	[#, #, #, #, #, #, #, #],	Arreglo Entero	Identifica al tipo de sensor
"sTh":	[#, #, #, #, #, #, #, #],	Arreglo Entero	Umbral de seguridad del sensor
"sSurp":	[#, #, #, #, #, #, #, #],	Arreglo Entero	Acción en caso de sobrepaso del umbral
"sAsoc":	[#, #, #, #, #, #, #, #],	Arreglo Entero	Grupo asociado al sensor
"sTimer":	#	Entero	Frecuencia de lectura y actualización de valores de los sensores en segundos
	}		
	}		

Tabla 18 Estructura de la respuesta HTTP, recurso 102

Ejemplo:

Petición:

<http://192.168.1.177/?res=102>

Respuesta:

```
{
  "device": {
    "success": true,
    "TCname": "Sala",
    "TCdescription": "CTSala",
    "TCtype": "RDomo",
    "TCversion": "1.00",
    "ip": "192.168.1.177",
    "mac": "222-173-192-168-1-177",
    "dns": "192.168.1.1",
    "gateway": "192.168.1.1",
    "netmask": "255.255.255.0",
    "serverIp": "192.168.1.11",
    "cType": [129, 129, 129, 2, 3, 3, 5],
    "groupid": [1, 1, 1, 4, 5, 5, 7],
    "elemName": ["DimmerBar", "DimmerBar", "DimmerBar", "LampSala", "ClimaSala", "ClimaSala", "Puerta"],
    "elemDesc": ["DimmerParaElBar", "DimmerParaElBar", "DimmerParaElBar", "LamparaSala", "ClimatizacionSala", "ClimatizacionSala", "PuertaPrincipal"],
    "groupVal": [0, 1, 0, 1, 1, 0, 0],
    "pAsoc": [1, 1, 1, 1, 6, 7, 0],
    "pAction": [2, 3, 2, 3, 1, 1, 0],
    "sType": [1, 1, 1, 3, 0, 0],
    "sTh": [44, 44, 144, 144, 0, 0],
    "sSurp": [1, 1, 2, 2, 0, 0],
    "sAsoc": [1, 4, 5, 7, 0, 0],
    "sTimer": 5
  }
}
```

3.2.1.1.1.2 case 111:

Se comprueba si el PIN de acceso es correcto. Esta petición requiere que se envíen los valores de los cuatro bytes correspondientes al PIN de acceso a continuación del código de recurso, cada uno separado por un guion medio (-).

Ejemplo:

Petición 1(PIN Correcto):

<http://192.168.1.177/?res=111-0-0-0>

Respuesta 1:

```
{"device":{"success":true}}
```

Petición 2 (PIN Erroneo):

<http://192.168.1.177/?res=111-0-3-0-31>

Respuesta 2(PIN erróneo):

```
{"device":{"success":false,"errorCode":921}}
```

3.2.1.1.1.3 case 121:

Obtiene los valores de los sensores. Cada valor del sensor debe ser separado por un guion medio (-) y ordenados ascendentemente de acuerdo al número de pin analógico.

Ejemplo

Petición:

<http://192.168.1.177/?res=121>

Respuesta:

1023-1023-0-0-119-163

3.2.1.1.1.4 case 201:

Establece valores a un elemento o grupo de elementos. Luego del código de recurso deben escribirse las parejas de grupo-valor de la siguiente forma:

“-IDgrupo1-Valor1-...-IDGrupoN-ValorN”

Ejemplo:

Petición:

<http://192.168.1.177/?res=201-1-7-4-1>

Respuesta:

```
{"device":{"success":true}}
```

En este ejemplo, al actuador 1 se establece un valor de 7(completamente iluminado) y al elemento 4 con 1(encendido).

3.2.1.1.1.5 case 901:

Permite subir la información al CT. Debido a la estructura compleja y confusa de este recurso, se ha creado la siguiente tabla con una explicación detallada de su organización.

Estructura de la petición HTTP GET, recurso 902 (Carga de configuración básica)					
http://192.168.1.177/?res=901-0-0-0-0-1-192-168-1-177-222-173-192-168-1-177-192-168-1-1-192-168-1-1-255-255-255-0-192-168-1-11-129-129-129-0-3-3-5-1-1-1-0-5-5-7-1-1-1-1-6-7-0-2-3-2-3-1-1-0-1-1-1-3-0-0-44-44-144-144-0-0-1-1-2-2-0-0-1-4-5-7-0-0-5-234					
Para una fácil interpretación la petición al recurso 901, se ha dividido en sección de colores alternados(-azul-rojo) cada trama de color diferente se explica en la siguiente tabla					
	Nombre de la sección	Valores de Ejemplo	Direcciones en la memoria EEPROM		Bytes usados
			Dir. Inicial	Dir. Final	
Parámetros de conexión	URL raíz	http://192.168.1.177/?	NoExiste	NoExiste	0
	recurso	res=901	NoExiste	NoExiste	0
	PIN de acceso	-0-0-0-0	501	504	4
	Tipo de dispositivo	-1	505	505	1
Parámetros de configuración de red	Dirección IP del CT	-192-168-1-177	0	3	4
	Dirección MAC del CT(decimal)	-222-173-192-168-1-177	4	9	6
	Dirección IP del Servidor DNS	-192-168-1-1	10	13	4
	Dirección IP de la puerta de enlace	-192-168-1-1	14	17	4
	Máscara de subred	-255-255-255-0	18	21	4
	Dirección IP de servidor	-192-168-1-11	22	25	4
Configuración de circuitos	Tipos de circuito para cada pin digital	-129-129-129-0-3-3-5	26	32	7
	Grupo asociado a cada pin digital	-1-1-1-0-5-5-7	33	39	7
Configuración de pulsadores	Asociación Pulsador-Grupo	-1-1-1-1-6-7-0	40	46	7
	Acción del pulsador	-2-3-2-3-1-1-0	47	53	7
Configuración de sensores	Tipos de sensor	-1-1-1-3-0-0	54	59	6
	Umbral crítico del sensor	-44-44-144-144-0-0	60	65	6
	Acción en caso de sobrepaso del umbral(ON/OFF)	-1-1-2-2-0-0	66	71	6
	Pin afectado por el sensor	-1-4-5-7-0-0	72	77	6
	Intervalo de lectura de sensores	-5	78	78	1
Fin	Fin de la trama (-234)	-234	Total Bytes		84

Tabla 19 Estructura de una petición al recurso 902 y su ubicación en la memoria EEPROM.

Respuesta en caso de éxito:

```
{"device":{"success":true}}
```

Respuesta si el PIN es incorrecto:

```
{"device":{"success":false,"errorCode":921}}
```

Respuesta si el tipo de dispositivo establecido es incorrecto:

```
{"device":{"success":false,"errorCode":922}}
```

A continuación se explica brevemente la función de cada fila registrada en la tabla anterior, que también resulta de utilidad para entender los campos de la herramienta de configuración.

PIN (del Inglés - Personal Identification Number) de Acceso.

El PIN consta de cuatro números de un byte cada uno (###-###-###-###) que funcionan como contraseña. Ofrece un nivel de seguridad básico para que únicamente los usuarios autorizados puedan cambiar los parámetros del dispositivo. El PIN predeterminado es: 0-0-0-0. El recurso 111 permite verificar el PIN, mientras el 902 es el único que sirve para cambiarlo.

Dirección IP V4 del dispositivo.

La dirección IP (Internet Protocol) funciona como el número identificador dentro de una red TCP/IP, está compuesto por cuatro bytes y debe ser único de cada dispositivo.

Dirección MAC.

La dirección MAC es el identificador del dispositivo TCP/IP dentro de una subred, sirve para la conmutación de paquetes y está compuesta por 48 bits (6 Bytes)

Dirección IP del Servidor DNS.

El servidor DNS (Domain Name System) o Sistema de Nombres de Dominio se encarga de resolver nombres de dominio, convirtiéndolo a direcciones IP.

Dirección IP de la puerta de Enlace.

Es la dirección del equipo que sirve de ruta predeterminada para todos los paquetes destinados a una dirección ajena a la subred del dispositivo.

Mascara de subred.

La máscara de subred indica, de manera lógica, la cantidad de dispositivos dentro de una subred.

Dirección IP del servidor.

Dirección IP del Controlador Central. A esta dirección se notifican los cambios de estado realizados mediante hardware.

Tipo de circuito.

Identifica la función que desempeña cada pin digital. Puede ser: Encendido/Apagado o Incremento/Disminución.

Grupo (circuito).

Establece la asociación de los pines, para casos en los que se requiera más de uno. Para los circuitos de iluminación variable se usan tres pines, por lo tanto los tres tendrán el mismo código de grupo.

Acciones de pulsadores.

Existe un bus dedicado, de 3 bits, para gestionar las acciones de los pulsadores (Pines digitales: 7, 8 y 9). Existen 3 acciones posibles: Encendido/Apagado, Incremento y finalmente Disminución

Grupo asociado al pulsador.

Cuando un pulsante es accionado se verifica la acción y el grupo, posteriormente envía la señal correspondiente al actuador, que a su vez realiza el cambio de estado en los dispositivos controlados.

Tipos de Sensores.

Se identifica el tipo de sensor instalado a cada pin analógico.

Umbral del sensor.

Cuando se cruce el umbral establecido, tanto en subida como en bajada, se solicita información al servidor sobre un posible cambio de estado. Es preciso establecer este valor en el nivel crítico porque será es el único que funcione cuando el CC se apague o falle.

Acción de sobrepaso del umbral.

Si el sensor detecta un sobrepaso en el umbral, el dispositivo puede apagar o encender un circuito.

Grupo asociado al sensor.

Se refiere al pin, o circuito al cual está asociado un determinado sensor. Un sensor de información, que no afecte a ningún circuito, siempre tendrá un valor de 201.

Intervalo de lectura de sensores.

Indica la frecuencia de lectura y notificación de los valores de los sensores al Controlador Central, para que este a su vez los analice y tome las acciones respectivas a cada valor.

3.2.1.1.1.6 case 902:

Este recurso permite el cambio del PIN de acceso al CT. Para cambiarlo se debe escribir a continuación del código de recurso el PIN anterior y el PIN nuevo, separando cada byte con un guion medio (-).

Petición:

<http://192.168.1.177/?res=902-0-0-0-0-15-0-9-255>

Respuesta:

```
{"device":{"success":true}}
```

3.2.1.1.1.7 case 903:

Permite cambiar únicamente la configuración de red del CT. Su estructura es muy similar al recurso 901, pero simplemente posee la sección “Parámetros de configuración de red”.

Ejemplo:

Petición:

<http://192.168.1.177/?res=901-0-0-0-0-1-192-168-1-177-222-173-192-168-1-177-192-168-1-1-192-168-1-1-255-255-255-0-192-168-1-11>

Respuesta en caso de éxito:

```
{"device":{"success":true}}
```

Respuesta si el PIN es incorrecto:

```
{"device":{"success":false,"errorCode":921}}
```

Respuesta si el tipo de dispositivo establecido es incorrecto:

```
{"device":{"success":false,"errorCode":922}}
```

3.2.1.1.1.8 case 904:

Mediante este recurso es posible cambiar el nombre y la descripción del CT, el nombre puede contener un máximo de 10 caracteres ASCII mientras la descripción puede contener hasta 40, el identificador de NULO es un guion bajo (_) a partir de la aparición de este carácter ningún otro es considerado.

Ejemplo:

Petición:

<http://192.168.1.177/?res=904-0-0-0-0-1->

[Sala_____ControlDeSala_____](#)

Respuesta:

```
{"device":{"success":true}}
```

Las direcciones de memoria EEPROM para guardar el nombre y descripción del dispositivo van desde la 601 hasta la 610.

3.2.1.1.1.9 case 905:

Es muy similar al anterior (904), excepto que la descripción de cada elemento puede contener hasta 30 caracteres ASCII, 10 menos que en el caso 904. Las parejas de valores (nombre-descripción) de los siete pines del CT se escriben uno a continuación de otro sin ningún separador.

Ejemplo:

Petición:

<http://192.168.1.177/?res=905-0-0-0-0-1->

[DimmerBar_DimmerParaElBar_____DimmerBar_DimmerParaElBar_____](#)

[_____DimmerBar_DimmerParaElBar_____LampSala_LamparaSala_____](#)

[_____ClimaSala_ClimatizacionSala_____ClimaSala_ClimatizacionSala_____](#)

[_____Puerta_____PuertaPrincipal_____](#)

Respuesta:

```
{"device":{"success":true}}
```

Las direcciones de memoria EEPROM para guardar el nombre y descripción del dispositivo van desde la 701 hasta la 980.

3.2.1.1.10 case 999:

Este recurso reinicia al CT, resulta útil luego de haber realizado modificaciones en los parámetros de red. La configuración de red se carga al encender el CT, por lo tanto para reflejar sus cambios es necesario reiniciar el equipo.

Ejemplo:

Petición:

<http://192.168.1.177/?res=999-0-0-0-0>

Respuesta:

```
{"device":{"success":true}}
```

3.2.1.2 Envío de notificaciones al servidor.

Existen dos tipos de notificaciones al servidor:

- Notificación de cambios por pulsadores.
- Notificación de valores de sensores, que a su vez se divide en dos tipos.
 - Notificación por cruce de umbral.
 - Notificación periódica: se establece el intervalo en segundos.

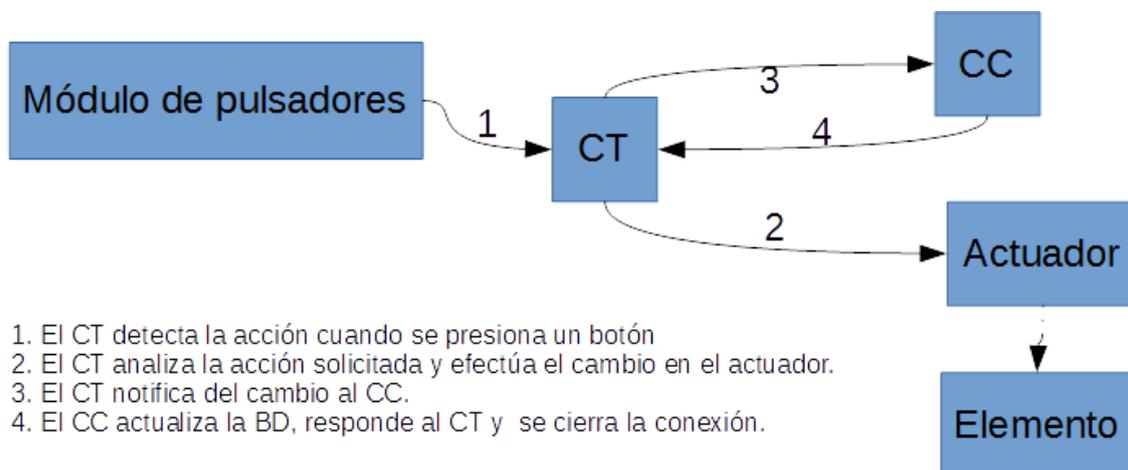


Figura 14 Esquema de comunicación **Pulsadores => CT => CC => CT**

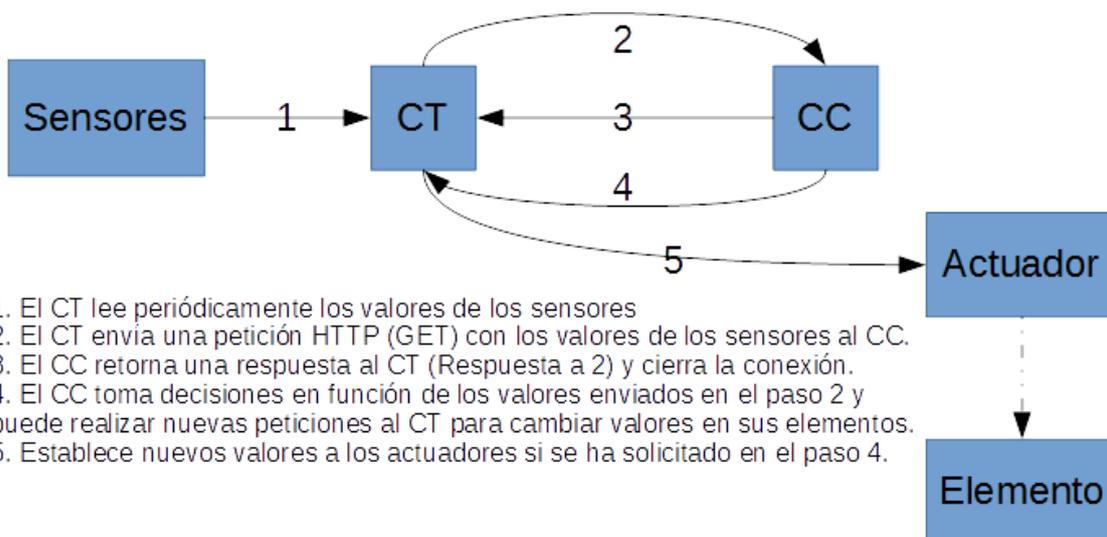


Figura 15 Esquema de comunicación **Sensores => CT => CC => CT => Actuador**

3.2.1.3 Herramienta de configuración

Debido a la gran cantidad de parámetros, las tramas de configuración pueden resultar confusas para ser generadas manualmente por una persona. Para evitar este inconveniente se dispone de una aplicación desarrollada en HTML, CSS y JavaScript (jQuery). Esta permite ingresar los valores, validarlos y enviarlos al dispositivo mediante AJAX (Asynchronous JavaScript And XML).

Digital Pins:	Name:	Description:
1: -/+ Lightning	DimmerB	DimmerParaEIBar
2: -/+ Lightning	DimmerB	DimmerParaEIBar
3: -/+ Lightning	DimmerB	DimmerParaEIBar
4: 0	LampSal	LamparaSala
5: ON/OFF AC	ClimaSal	ClimatizacionSala
6: ON/OFF AC	ClimaSal	ClimatizacionSala
7: Acces Control	Puerta	PuertaPrincipal

Sensor Types:	Sensor Threshold:	Threshold Surpass:	Affected Dpin:
1: Light Lc	44	ON	1
2: Light Lc	44	ON	4
3: Light Lc	144	OFF	5
4: Temper	144	OFF	7
5:	0		
6:	0		

Figura 16 Ventana de configuración de CTs.

En la figura anterior se ha marcado y numerado cada sección. A continuación se explica cada una de estas:

1. Nombre y descripción de CT.
2. Esta sección contiene 3 campos y 3 botones:
 - a. Campo "Current IP": debe escribirse la dirección IP del CT que se desea configurar.
 - b. Botón "Load Config From Device": Al hacer clic sobre este botón, si la Dirección IP establecida en el campo anterior corresponde a un CT conectado a la red, cargará su configuración actual (no requiere PIN)
 - c. Campo "Current PIN": para cargar una configuración nueva es necesario que el PIN de acceso coincida con el guardado en el CT. El PIN de acceso consta de 4 bytes expresados en decimal y separados cada uno por un guion medio (-).
 - d. Botón "Change PIN": al presionar este botón se muestra una nueva ventana para realizar el cambio de PIN, en la cual debe escribirse el nuevo valor, con las mismas consideraciones que se tienen para el campo "Current PIN".
 - e. Campo "Device Type": debe seleccionarse el tipo correcto de dispositivo (Actualmente existe únicamente un tipo). Resultará útil si en el futuro se definen nuevos tipos de CT.
 - f. Botón "Test Connection": Comprueba que la conexión con el dispositivo pueda establecerse de acuerdo a los tres campos anteriores.
3. En esta sección debe establecerse los nuevos parámetros de red que tomará el CT luego de cargar la configuración.
 - a. Campo "New IP": nueva dirección IP del CT.
 - b. Casillero "Auto MAC Address": si está activado el software auto calcula una dirección MAC de administración local para el CT. El cálculo se realiza en función de la dirección IP. Los dos primeros bytes son fijos, mientras los cuatro octetos siguientes son los mismos de la dirección IP V4, este resultado es finalmente convertido a hexadecimal y mostrado en el siguiente campo.
 - c. Campo "New MAC Address": dirección MAC del CT.
 - d. Campo "New DNS Server": dirección IP del servidor DNS, actualmente no se usa debido a que se maneja directamente por dirección IP.

- e. Campo “New Gateway”: Puerta de enlace predeterminada, resulta de utilidad en caso de redes ruteadas.
 - f. Campo “Net Mask”: máscara de subred del CT.
 - g. Campo “New Server IP”: IP de CC al que se le enviarán las notificaciones:
 - h. Botón “Save Network Config”: Guarda únicamente la configuración de red del CT.
4. En esta sección se establece la función de cada pin digital, se debe seleccionar el tipo de circuito conectado a un pin o un grupo de pines. En caso de que el circuito seleccionado requiera más de un pin, automáticamente bloqueará a los siguientes que necesite.
 5. Se establece un nombre para cada actuador.
 6. Se escribe una descripción para cada actuador
 7. En esta sección se asocian actuadores y acciones a cada botón del módulo de pulsadores.
 8. Debe especificarse el tipo de sensor conectado a cada PIN analógico.
 9. Se deben establecer los umbrales para cada sensor.
 10. Establecer una acción en caso de sobrepaso del umbral de cada sensor, encender o apagar un circuito.
 11. Se debe seleccionar el circuito afectado por el sensor.
 12. Intervalo de tiempo entre notificaciones al CC sobre valores de sensores.

Una explicación complementaria se encuentra en la sección [3.2.1.1.1.5 case 901](#)

3.2.2 Mini Servidor

El Mini servidor o Controlador Central (CC) expone servicios web para dos finalidades: primeramente para que otros desarrolladores puedan extender la funcionalidad del sistema domótico mediante el desarrollo de aplicaciones que faciliten su manejo y segundo para que los CT envíen sus notificaciones. Además, actúa como cliente HTTP cuando requiera enviar ordenes al CT. Con estas dos funciones se mantiene actualizada la base de datos sin importar si las ordenes provienen de un módulo de pulsadores o un equipo informático cliente.

3.2.2.1 Servicios Web

Todos los servicios web han sido escritos en el mismo archivo (<http://<IP>/api/ws.php/?res=...>). La diferencia entre unos y otros comienza a partir del parámetro *res* (en la tabla se denomina recurso).

Recurso	Descripción de recursos	Parámetros opcionales	Valores posibles	Método
tcdiscoverer	Busca CTs dentro de las subredes establecidas en la base de datos.			GET
Respuestas	1){"device":{"success":false}} 2) {"device":{"success":true}}			
checkelements	Actualiza el estado de todos los dispositivos que se encuentran registrados en la Base de datos. Si el parámetro idtc es establecido, scope es ignorado.	idtc	Dirección IP del CT	GET
		scope	all, online, offline	
Respuestas	1){"device":{"success":false}} 2) {"device":{"success":true}}			
getelements	Obtiene todos los elementos de la base de datos de acuerdo al parámetro scope y los devuelve en un formato establecido en el parámetro format(scope	all, online, offline	GET
		format	json, ssv	
Respuesta	[{"idtc":"192.168.1.177", "idelement":1, "description":"DimmerParaElBar", "idelementType":129, "value":7, "online":true, "obeySensor":true}]			
settempvalues	Establece los umbrales para el sensor de temperatura	idtc*	Dirección IP del CT	GET
		groupid*	1-7 (identificador del elemento en el CT), debe ser de tipo AC (3), caso contrario no se efectúa la petición.	
		lowerlimit*	Límite inferior de la temperatura en grados C.	
		higherlimit*	Límite superior de la temperatura en grados C.	
Respuestas	1){"device":{"success":false}} 2) {"device":{"success":true}}			
getcurrenttemp	Obtiene la temperatura actual del sensor de temperatura asociado al	idtc*	Dirección IP del CT	GET

	CT	groupid*	1-7 (identificador del elemento en el CT), debe ser de tipo AC (3), caso contrario no se efectúa la petición.	
Respuestas	{"temp":28"unit":"C"}			
setvalue	Establece un valor a un elemento	idtc*	Dirección IP del CT	GET
		groupid*	1-7 (identificador del elemento en el CT)	
		value*	0-7 (Depende del tipo de elemento)	
		src	"tc" (Solo se debe establecer cuando la petición proviene del CT)	
Respuestas	1){"device":{"success":false}} 2) {"device":{"success":true}}			
setvaluemulti	Establece valores a múltiples elementos	idtc*	Dirección IP del CT	GET
		values*	group1-value1-group2-value2-.....group_n-value_n	
Respuestas	1){"device":{"success":false}} 2) {"device":{"success":true}}			
sensorsnotify	Notifica al servidor con los nuevo valores de sensores y en función de estos toma decisiones sobre los elementos de los CTs.	idtc*	Dirección IP del CT donde provienen los valores de los sensores	GET
		sensorvalues*	#-#-#-#-#-# (Valores de los sensores separados por un guion medio)	
Respuestas	1){"device":{"success":false}} 2) {"device":{"success":true}}			
getsensordata	Obtiene los valores actuales de los sensores de un CT	idtc*	Dirección IP del CT	GET
Respuestas	1023-1023-0-0-135-176			
shiftoveysensor	Cambiar de obedecer a no obedecer (o viceversa) a cambios por sensores	idtc*	Dirección IP del CT	GET
		groupid*	1-7(Elemento al cual se activa o desactivan los sensores)	
Respuestas	1){"device":{"success":false}} 2) {"device":{"success":true}}			
obeysensor	Parecido a shiftoveysensor, con la	idtc*	Dirección IP del CT	GET

	excepción de que en este caso el valor se establece explícitamente.	groupid*	1-7(Elemento al cual se activa o desactivan los sensores)	
		obey*	1 o 0 (obedecer o NO obedecer)	
Respuestas	1){"device":{"success":false}} 2){"device":{"success":true}}			
turnoff	Apaga por completo a un elemento al mismo tiempo que deja de obedecer a los sensores	idtc*	Dirección IP del CT	GET
		groupid*	1-7(Elemento afectado)	
Respuestas	1){"device":{"success":false}} 2){"device":{"success":true}}			
login	Verifica las credenciales de un usuario del sistema	user*	Nombre de usuario	POST
		pass*	Contraseña de usuario	
Respuestas	1){"device":{"success":false}} 2){"device":{"success":true}}			

3.2.2.2 Envío de órdenes a los CTs.

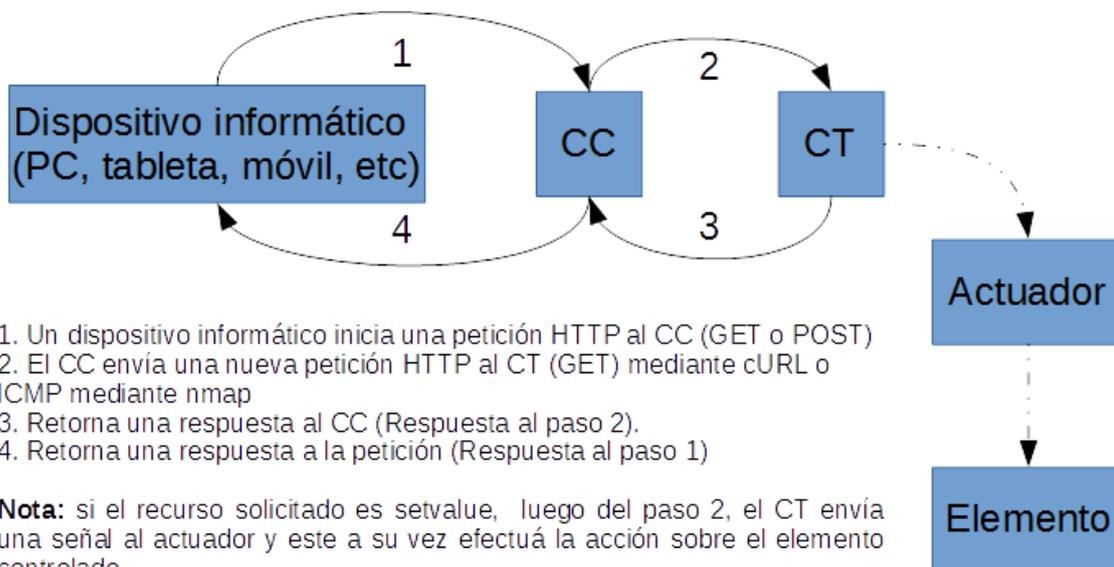


Figura 17 Esquema de comunicación *Cliente => CC => CT => CC => Cliente*

3.2.2.3 Recepción de notificaciones procedentes de un CT.

Actualmente se han definido tres tipos de notificaciones desde los CTs hacia el CC, estos son:

1. Notificación de valores de sensores: esta clase de notificaciones se ejecuta periódicamente de acuerdo a un parámetro ajustable en la configuración de cada TC.
2. Actualización de valores de elementos: cada vez que un CT se encienda, luego de haberse auto configurado con sus parámetros de red, solicita al CC sus últimos estados. Luego, el CC realiza una nueva petición HTTP al CT con los valores.
3. Notificación de cambios por pulsadores: siempre que se haya producido un cambio en los actuadores, mediante el módulo de pulsadores, se envía una notificación al CC el cual a su vez registra el cambio en la base de datos.

3.2.2.4 Autodescubrimiento y reconstrucción del estado actual del sistema.

La función de autodescubrimiento de dispositivos se compone de dos etapas.

1. Descubrimiento de dispositivos IP
 - a. Se realiza mediante el envío pings, sobre el protocolo ICMP (*Internet Control Message Protocol*), usando el software nmap. Las peticiones son enviadas a todos los dispositivos dentro de las subredes establecidas para la búsqueda.
2. Filtrado de dispositivos.
 - a. Cuando se tiene una lista con todos los dispositivos que han respondido a las solicitudes de ping, se envían peticiones http especiales de tipo GET a todos los dispositivos de la lista usando la librería cURL. Las respuestas a estas solicitudes son verificadas y en el caso de determinarse valida, se agrega o actualiza la información en la base de datos.
 - b. Para evitar tiempos de respuesta demasiado altos, las peticiones se realizan en paralelo y se establece un tiempo de respuesta prudentemente bajo (entre 1 y 5 segundos).

3.2.2.5 Programador de acciones de acuerdo a comportamientos anteriores.

Todo cambio de estado que se realice sobre un elemento de un determinado CT se registra en la tabla element_logs. Basado estos registros y con la ayuda de un *script* que se ejecute periódicamente, evalúe los valores anteriores y finalmente los replique al sistema actual se logra la simulación de presencia en el hogar. La

herramienta contab de Linux permite realizar ejecuciones periódicas de aplicaciones o *scripts*.

3.2.2.6 API para desarrolladores

La API está dividida en dos niveles, una a nivel de clases y otra de servicios web. La primera se puede usar para mejorar o crear clases nuevas o servicios web, mientras la segunda resulta muy importante para desarrolladores que requieran crear nuevas aplicaciones para el manejo o configuración del sistema. Para una descripción detallada de los servicios web vea la sección 3.2.2.1 (Servicios Web).

3.3 Interfaces de Hardware

De acuerdo a la interfaz de hardware conectada directamente al CT existen tres tipos de comunicación con el CT. Esta comunicación se caracteriza por no poseer un acuse de recibo, es decir, el dispositivo que emite la señal no recibe respuesta a dicha notificación.

3.3.1 Pulsadores

El CT recibe las órdenes procedentes del módulo de pulsadores. Los valores pueden estar entre 001 y 111, 000 indica que el bus de pulsadores está inactivo.

3.3.2 Actuadores

La comunicación con los actuadores es directa y no existe retroalimentación. Es así que si enviamos una orden de encendido a un actuador y este se encuentra desconectado o no posee un elemento (lámpara, motor, etc...), no existe forma de conocer si la acción ha ocasionado el efecto deseado. En los casos en los que se requiera verificar si la orden enviada al actuador se ha efectuado, es recomendable agregar sensores que retroalimenten al CT sobre el estado actual de un determinado fenómeno físico.

3.3.2.1 Encendido / Apagado

Un único bit es usado para este tipo de circuitos pudiendo cambiar únicamente a dos estados (0-1).

3.3.2.2 Incremento / Disminución

Los circuitos de incremento/disminución utilizan tres pines digitales contiguos para su funcionamiento Ej. (3, 4, 5) siendo el bit más significativo el primer pin usado (3) y el menos significativo el último (5).

3.3.3 Sensores

El CT lee periódicamente los valores de voltaje procedentes de sensores conectados a sus pines analógicos, dichos valores pueden estar entre 0V a 5V. El CT tiene una precisión de 10 bits (0-1023). Los valores recolectados son analizados, y basado en el umbral de nivel crítico y la acción en caso de sobrepaso, enciende o apaga el circuito afectado. Posterior a esto, el CT envía una notificación al CC para que este último tome una decisión de acuerdo a la configuración avanzada de sensores.

técnico encargado de la instalación domótica, con el conocimiento que posee, debe guiarlo para que sus decisiones sean correctas y alcanzables.

1. Obtención de la cantidad requerida de elementos de cada tipo, de esta forma también se estará consiguiendo el número y tipo de actuadores demandados por el sistema.
2. Determinar la cantidad y tipo de sensores necesario. Debe tomarse en cuenta que para el correcto funcionamiento del sistema existen circuitos, como los de climatización y control de acceso, que siempre requieren de un sensor.
3. De acuerdo a la cantidad y tipo de elementos requeridos, se debe determinar cuántos CTs son necesarios para gestionarlos. No siempre es conveniente tratar de optimizar al máximo el uso de las salidas digitales, básicamente por dos razones:
 - a. En el futuro puede existir la necesidad de agregar nuevos circuitos, entonces es conveniente que el sistema posea pines libres para estos nuevos elementos.
 - b. El ahorro en CTs puede estar asociado con otros gastos como cable extra debido a mayores distancias hasta los actuadores o sensores.
4. Determinar el número de módulos de pulsadores: uno por CT resulta suficiente, pero, si se requiere replicar los comandos para distintos lugares, pueden agregarse módulos adicionales. Debe notarse que cada CT puede gestionar únicamente 7 comandos por pulsador.
5. Determinar el costo de los equipos (CTs, actuadores, sensores y pulsadores). Con el conocimiento de los equipos requeridos para el sistema es posible calcular su costo.

Para simplificar el cálculo de costos se creó una plantilla en formato Excel

4.1.1 Ejemplo

El ejemplo que se muestra a continuación se tomará como referencia para el resto de este capítulo y, debido a que es una demostración pequeña, el número de elementos requeridos es bajo en comparación a una vivienda real, pero, el procedimiento es similar para un caso concreto.

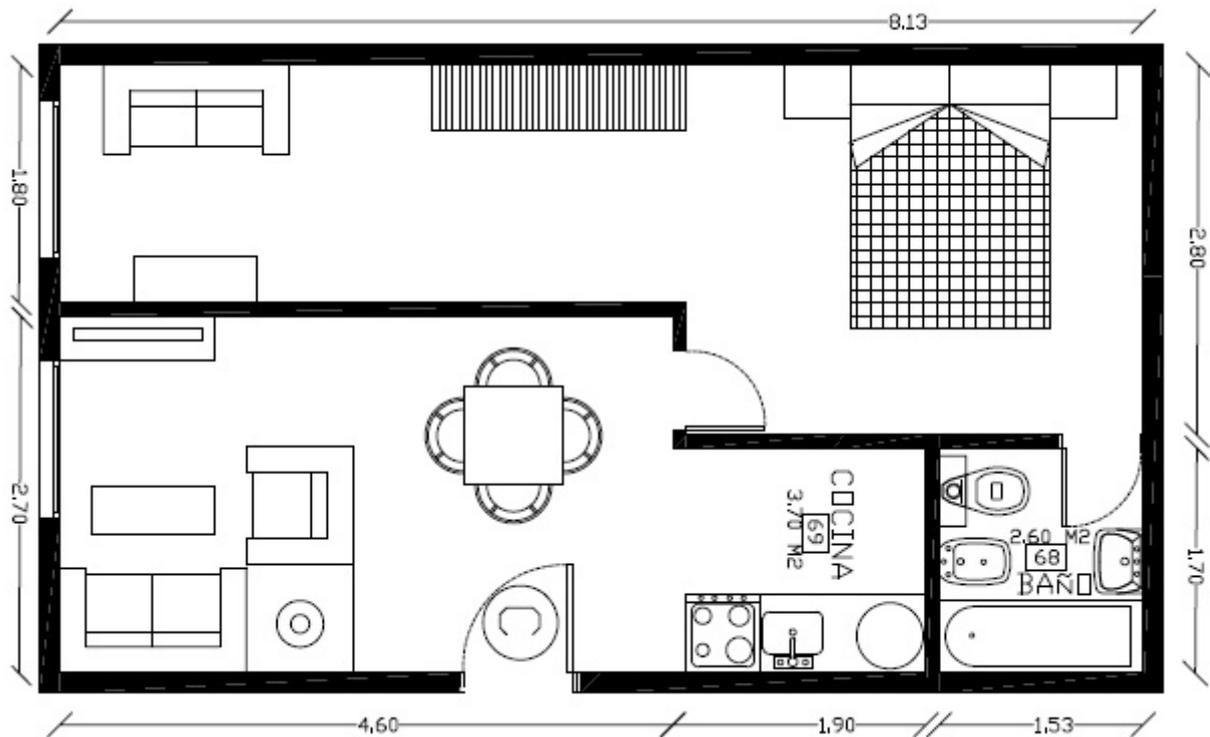


Figura 19 Plano de la vivienda de ejemplo (Pedersen, Bienes Raices, 2014)

Para la vivienda mostrada en la imagen anterior se pueden identificar las siguientes necesidades:

Cálculo de costos de elementos domésticos

Descripción	Cantidad	Costo Total
CC	1	\$ 75,00
Total CTs	2	\$ 120,00
Total pulsadores (módulos)	2	\$ 50,00
Elementos	10	\$ 240,00
Sensores	3	\$ 65,00
P. Total		\$ 550,00

Requerimientos de la vivienda

#	Tipo de actuador	Nombre	Descripción	ID CT	#pinos	Costo Actuador	Sensor	Costo Sensor
1	ON/OFF Iluminación	LampBaño	Lámpara Baño	192.168.1.177	1	\$ 20,00	?	\$ 0,00
2	ON/OFF Iluminación	LampDorm	Lámpara Dormitorio	192.168.1.177	1	\$ 20,00	?	\$ 0,00
3	ON/OFF Iluminación	LampCocina	Lámpara Cocina	192.168.1.177	1	\$ 20,00	?	\$ 0,00
4	ON/OFF Iluminación	LampPasil	Lámpara Pasillo	192.168.1.177	1	\$ 20,00	?	\$ 0,00
5	ON/OFF Iluminación	LampMiniS	Lámpara Mini Sala	192.168.1.177	1	\$ 20,00	?	\$ 0,00
6	ON/OFF Riego	Riego	Riego Jardín	192.168.1.177	1	\$ 20,00	Sensor de Humedad	\$ 30,00
7	INC/DIS Iluminación	DimmerSala	Dimmer Sala	192.168.1.181	3	\$ 45,00	Fin Carrera	\$ 20,00
8	Climatización	ClimaSala	Climatización Sala	192.168.1.181	2	\$ 35,00	Sensor de Temperatura	\$ 15,00
9	ON/OFF Iluminación	LampComed	Lámpara Comedor	192.168.1.181	1	\$ 20,00	?	\$ 0,00
10	Control de Acceso	PuertaPrin	Puerta Principal	192.168.1.181	1	\$ 20,00	?	\$ 0,00

Tabla 20 Listado de elementos, sensores y CTs requeridos (captura de la plantilla)

4.2 Diseño

Esta etapa debe iniciar una vez que se han identificado claramente las necesidades que tienen el cliente en la vivienda. Básicamente pueden notarse dos

escenarios de implementación bien diferenciados: viviendas por construir y viviendas construidas, resulta útil la realización de gráficos y planos eléctricos

1. Distribución de actuadores.
 - a. Para ahorrar cable, siempre que sea posible, es beneficioso ubicar los Actuadores cerca de los elementos eléctricos controlados.
2. Ubicación de sensores
 - a. Debe definirse exactamente la zona de control, entonces, los sensores han de ubicarse en ese lugar o alguno cercano que refleje la realidad de la zona objetivo.
3. Distribución de pulsadores
 - a. Los pulsadores deben colocarse en lugares de fácil acceso, si se requiere comandar desde varios lugares debe agregarse nuevos módulos y necesariamente tienen que ser conectados en paralelo.
4. Ubicación de concentradores y enrutadores.
 - a. Debe tomarse en cuenta los costos de los equipos, distancias entre CTs, concentradores y enrutadores, además de las dimensiones de la red. Por ejemplo: a mayor distancia entre equipos se requiere mayor cantidad de cable, mientras que su reducción conlleva a la necesaria instalación de equipamiento de redes y por lo tanto a una reducción de costos en cable pero a costa de un aumento en dispositivos de red.
 - b. Una ventaja de agregar concentradores y enrutadores es la fácil gestión del sistema, pudiendo, por ejemplo, seccionar cada planta dentro de una subred mediante el uso de concentradores conectados un enrutador central.
5. Cableado
 - a. Para evitar interferencias con las líneas de poder, es conveniente que los cables de comunicación de la red domótica se lleven por canales o tuberías separadas.

4.2.1 Ejemplo

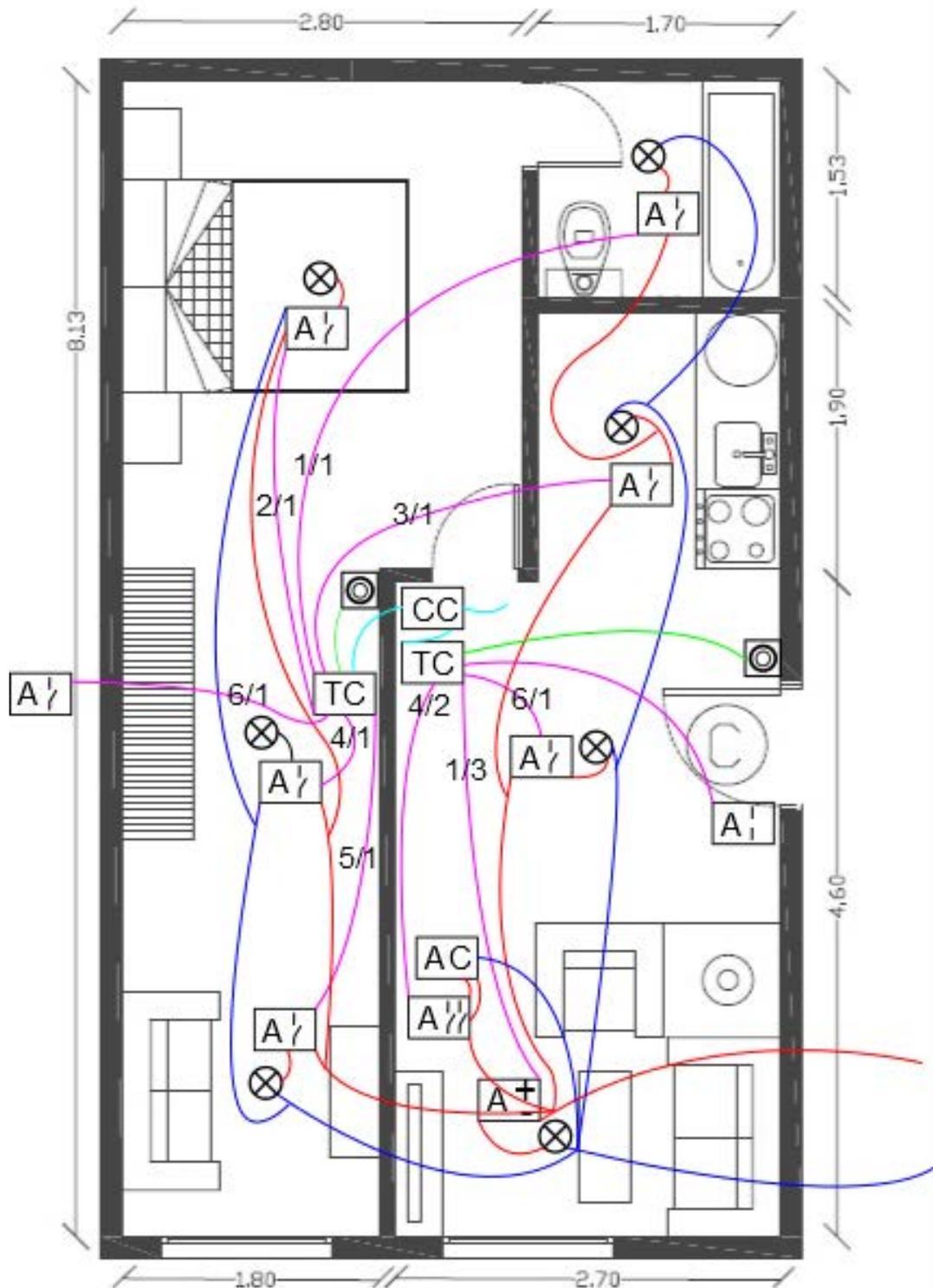


Figura 20 Plano Domótico - Eléctrico

En la figura anterior se muestra el mismo plano expuesto en la etapa de análisis, al cual se le han incluido las conexiones eléctricas y domóticas.

4.3 Implementación

Por lo general, el o los técnicos encargados de la instalación no son los mismos que diseñan red, por lo cual debe existir una estandarización en los procesos de análisis, diseño e implementación.

4.3.1 Configuración de dispositivos

Con la ayuda de la herramienta de configuración de CTs, se debe cargar la información a cada dispositivo. Dicha información debe basarse en la documentación obtenida en las dos etapas de anteriores (Análisis y Diseño).

4.3.2 Instalación de equipos

Para la colocación y cableado de los equipos domóticos, el instalador debe basarse en el plano Eléctrico - Domótico generado en la etapa de diseño. La simbología se encuentra en la sección 2.1.3 (Actuadores).

4.4 Pruebas

Cada circuito instalado debe ser probado antes de dar por terminada la instalación, debe probarse tratando de simular todos los escenarios posibles. Si el instalador detecta fallos que no puede corregir, este debe notificar a la persona encargada del diseño de la red.

4.4.1 Ejemplo

Se supone un ejemplo de fallo: la puerta principal no permanece cerrada, siempre que esta se cierra, inmediatamente vuelve a abrirse. El instalador ha revisado que todas sus conexiones están de acuerdo al plano indicado. El técnico de diseño revisa las configuraciones de equipos y detecta que ha establecido un tipo de circuito erróneo para el portero. Se realizan los cambios en la configuración y el sistema funciona correctamente.

4.5 Mantenimiento

Debido a la etapa prematura en la que se encuentra el proyecto, es difícil estimar con un alto grado de exactitud el tiempo promedio entre fallas y por ende el tiempo de mantenimiento preventivo óptimo. Se recomienda un periodo mínimo de 6 meses entre revisiones preventivas. Luego de acuerdo a la experiencia y estadísticas de fallas, se pueden afinar estos parámetros aun ambiguos.

Conclusiones

Luego de haber terminado con la investigación y desarrollo, tanto de este documento como del software, también se culmina con una arquitectura doméstica funcional, lo cual corrobora que el objetivo general propuesto se ha cumplido. Si bien aún resulta temprano para determinar su robustez, las pruebas de rendimiento ofrecen buenos augurios si la investigación continúa.

Llegado a este punto, necesariamente se ha requerido:

- Establecer las especificaciones para la comunicación física y lógica de dispositivos.
 - Sin estas especificaciones no sería posible la comunicación con el sistema, estas ha sido especificadas en el capítulo II y III de este documento.
- Desarrollar el software para el control y comunicación de dispositivos.
 - Las funciones y clases del núcleo del sistema han sido desarrolladas en el lenguaje de programación PHP y son las que se encargan de gestionar la comunicación y control con los Controladores Terminales.
- Crear una Interfaz de Programación de Aplicaciones (API).
 - Existe una investigación, paralela a esta, denominada: *“Desarrollo de una aplicación en Android para el manejo de una arquitectura doméstica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi”*. El producto final de esta es una aplicación móvil para el sistema Android y usa la API desarrollada para la interacción con el sistema doméstico.
- Demostrar, mediante un prototipo, la arquitectura analizada.
 - Se ha creado un prototipo pequeño para poder demostrar la funcionalidad del sistema.

Una de las desventajas de usar la tecnología propuesta es que no existen pruebas aceleradas del tiempo promedio entre fallas (*Accelerated MTBF - Mean Time Between Failure*), ni del dispositivo Arduino UNO así como tampoco del Raspberry Pi, por lo cual no es apropiado estimar una vida útil del sistema y aún se mantiene como una incógnita en contra.

Recomendaciones

Si la investigación continúa:

- Crear vistas en MySQL para que la obtención de información requerida, producto de varias tablas y cálculos, se simplifique y estandarice.
- Modificar el firmware para otros tipos de Arduino y *Shields* puedan funcionar como CT.
- Mejorar el sistema electrónico de dimerización y buscar la manera de crear un sistema incremento o disminución continua.
- Tomar en cuenta que el dispositivo Raspberry Pi puede ser reemplazado con otro dispositivo en el cual se pueda instalar una distribución de Linux. Para evitar mayores cambios en la configuración del mismo, de preferencia el dispositivo debe soportar una distribución basada en Debian.
- Buscar una solución al problema de que los pines analógicos 0 y 1 no funcionan como se espera, la lectura de sensores en estos pines muestra un valor mucho más alto del real.
- Optimizar el módulo de pulsadores para que varias acciones puedan ser ejecutadas simultáneamente en un mismo CT.
- Mejorar el sistema de notificaciones por correo electrónico.

Durante la instalación de los elementos domóticos:

- La instalación de los actuadores y controladores terminales debe ser realizada por personal calificado, debido a que un error puede conllevar a daños graves en los equipos, personas que instalan o residen en la vivienda.

Para los usuarios del sistema:

- Notificar sobre cualquier anomalía en el sistema a la persona encargada de la instalación. De ninguna manera debe intentar repararlo si desconoce su funcionamiento. Esto puede ocasionar daños mayores.

Bibliografía

- Andrew, D. (2013). *Raspberry Pi Home Automation with Arduino*. Birmingham: Packt Publishing.
- ATMEL. (s.f.). *ATMEGA328P Datasheet*. ATMEL.
- BLAM Electronics. (08 de 06 de 2014). *BLAM Electronics*. Obtenido de <http://mikiblam.blogspot.com/p/blog-page.html>
- Domínguez, H. M., & Sáez Vacas, F. (2006). *Domótica: Un enfoque sociotécnico*. Madrid: Fundación Rogelio Segovia para el Desarrollo de las Telecomunicaciones.
- Madrid vive ahorrando energía. (2007). *La domótica como solución de futuro*. Madrid.
- Margolis, M. (2012). *Arduino Cookbook* (2 ed.). Sebastopol: O'Reilly.
- Pedersen, Bienes Raices. (10 de 07 de 2014). *Pedersen, Bienes Raices*. Obtenido de http://www.pedersenbienesraices.com/?attachment_id=3655
- Plan Avanza. (s.f.). *La casa digital*. Red.es.
- Txapuzas. (29 de 04 de 2014). *Txapuzas*. Obtenido de <http://txapuzas.blogspot.com/2010/06/regulador-de-potencia-por-triac-para.html>
- Web Arduino. (s.f.). *Arduino - Home Page*. Recuperado el 14 de 03 de 2014, de *Arduino - Home Page*: <http://www.arduino.cc/es/>
- Web Raspberry Pi. (s.f.). *FAQs | Raspberry Pi*. Recuperado el 14 de 03 de 2014, de *FAQs | Raspberry Pi*: <http://www.raspberrypi.org/faqs>
- Wikipedia. (30 de 05 de 2014). *Wikipedia*. Obtenido de <http://es.wikipedia.org/wiki/Rel%C3%A9>
- Wilson, J. (2005). *Sensor Technology Handbook*. Burlington: Elsevier.
- Xataka. (15 de 06 de 2014). *Xataka*. Obtenido de <http://www.xataka.com/componentes-de-pc/conoce-a-la-placa-que-quiere-revolucionar-tu-mundo-digital-raspberry-pi-a-fondo>

ANEXOS

Los documentos anexos, producto de esta investigación, servirían de poca ayuda si se presentan embebidos en este documento. Por esta razón se adjuntan en formato digital, en un disco óptico. El disco contiene lo siguiente:

1. Código fuente del Controlador Central.
2. Modelo Entidad-Relación de la base de datos, desarrollado con el software MySQL Workbench 6.1 Community Edition.
3. Código Fuente del Controlador Terminal, desarrollado con el IDE Arduino v1.0.5-r2
4. Hoja de cálculo: Plantilla de Análisis y Diseño. Adjunto en el disco óptico.
5. Manual de usuario de la aplicación de configuración del sistema.

DOCTOR ROMEL MACHADO CLAVIJO,

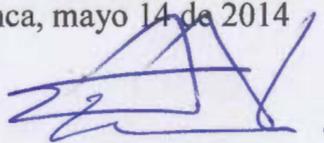
SECRETARIO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACION

DE LA UNIVERSIDAD DEL AZUAY,

CERTIFICA:

Que el H. Consejo de Facultad de Ciencias de la Administración en sesión del 14 de mayo de 2014, conoció la petición de señor **ORLANDO PATRICIO CHACON MOLINA (42433)** que denuncia su trabajo de titulación denominado: **“ANALISIS PARA UN SISTEMA DOMOTICO CON LA ARQUITECTURA ARDUINO Y RASBERRY PI, SOBRE TCP”** presentado como requisito previo a la obtención del Grado de Ingeniero de Sistemas. El Consejo acoge el informe de la Junta Académica y aprueba la denuncia. Designa como Director de dicho trabajo al ingeniero Diego Chacón Troya y como miembro del Tribunal Examinador al ingeniero Kenneth Palacio Baus. De conformidad a las disposiciones reglamentarias el denunciante deberá presentar su trabajo de graduación en un plazo no mayor a **TRES MESES** contados a partir de la fecha de aprobación, esto es hasta el 14 de agosto de 2014.

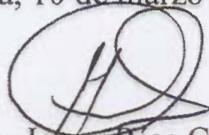
Cuenca, mayo 14 de 2014



CONVOCATORIA

Por disposición de la Junta Académica de Ingeniería de Sistemas, se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación "ANÁLISIS PARA UN SISTEMA DOMOTICO CON LA ARQUITECTURA ARDUINO Y RÀSBERRY PI, SOBRE TCP", presentado por el estudiante Orlando Patricio Chacón Molina con código 42433, previa a la obtención del grado de Ingeniero de Sistemas, para el día **JUEVES 17 DE ABIL DE 2014 A LAS 08H30.**

Cuenca, 10 de marzo de 2014



Dra. Jenny Ríos Coello
Secretaria de la Facultad

Ing. Diego Chacón Troya



.....

Ing. Kenneth Palacio Baus

.....

unical
m.



ACTA

SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

- 1.1 **Nombre del estudiante:** Orlando Patricio Chacón Molina
Código 42433
Director sugerido: Ing. Diego Chacón Troya
- 1.2 **Codirector (opcional):** _____
- 1.3 **Tribunal:** Ing. Kenneth Palacio Baus
- 1.4 **Título propuesto:** "Análisis para un sistema domótico con la arquitectura Arduino y Raspberry Pi, sobre TCP"
- 1.5 **Resolución:**

1.5.1 Aceptado sin modificaciones _____

1.5.2 Aceptado con las siguientes modificaciones:

Corregir los fallos ortográficos
Revisar el cronograma
Revisar el presupuesto y componer los.

- Responsable de dar seguimiento a las modificaciones (designado por la Junta Académica de entre los Miembros del Tribunal): Ing. Diego Chacón Troya

1.5.3 No aceptado

- Justificación:

Tribunal

.....
Ing. Diego Chacón Troya

.....
Ing. Kenneth Palacio Baus

.....
Sr. Orlando Chacón Molina

.....
Dra. Jenny Ríos Coello
Secretario de Facultad



RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN

- 1.1 Nombre del estudiante:** Orlando Patricio Chacón Molina, código 42433
- 1.2 Director sugerido:** Ing. Diego Chacón Troya
- 1.3 Codirector (opcional):** apellido, nombre y título.
- 1.4 Título propuesto:** “Análisis para un sistema domótico con la arquitectura Arduino y Raspberry Pi, sobre TCP”
- 1.5 Revisores (tribunal):** Ing. Kenneth Palacio Baus
- 1.6 Recomendaciones generales de la revisión:**

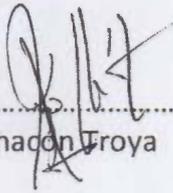
	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
Línea de investigación				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?	/			
Título Propuesto				
2. ¿Es informativo?	/			
3. ¿Es conciso?	/			
Estado del arte				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?	/			
5. ¿Describe la teoría en la que se enmarca el trabajo	/			
6. ¿Describe los trabajos relacionados más relevantes?	/			
7. ¿Utiliza citas bibliográficas?	/			
Problemática y/o pregunta de investigación				
8. ¿Presenta una descripción precisa y clara?	/			
9. ¿Tiene relevancia profesional y social?	/			
Hipótesis (opcional)				
10. ¿Se expresa de forma clara?	/			
11. ¿Es factible de verificación?	/			
Objetivo general				
12. ¿Concuerda con el problema formulado?	/			
13. ¿Se encuentra redactado en tiempo verbal infinitivo?	/			
Objetivos específicos				

- Opcional cuando cumple totalmente,
- Obligatorio cuando cumple parcialmente y NO cumple.

.....*Asegure de acreditar presupuesto. Falta costo materiales.*.....

.....*Revisar el tiempo en cada actividad.*.....

.....



.....
Ing. Diego Chacon Troya



.....
Ing. Kenneth Palacio Baus

Oficio Nro. 034-2014-DIST-UDA

Cuenca, 09 de Abril de 2014

Señor Ingeniero
Xavier Ortega Vázquez
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 9 de Abril del 2014, revisó el proyecto de monografía titulado "Análisis para un sistema domótico con la arquitectura Arduino y Raspberry Pi, sobre TCP/IP", presentada por el estudiante Orlando Chacón, estudiante de la Escuela de Ingeniería de Sistemas, previo a la obtención del título de Ingeniero de Sistemas.

La Junta considera que el diseño de trabajo de titulación cumple con los requisitos normados en la "Guía de Elaboración y Presentación de la Denuncia/Protocolo de Trabajo de Titulación", razón por la cual solicita, por su digno intermedio, notificar al tribunal designado y determinar lugar, fecha y hora de sustentación.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomienda como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Diego Chacón (docente del curso de graduación), y como miembro del Tribunal al Ing. Kenneth Palacio.



Atentamente,

Ing. Marcos Orellana Cordero
Director Escuela de Ingeniería de Sistemas y Telemática
Universidad del Azuay

Sustentación del Diseño de Monografía (JENNY RIOS COELLO)

Fecha: 10-04-2014

ESCUELA DE INGENIERIA DE SISTEMAS

Diseños de Monografía

Escuela de Ingeniería de Sistemas

Estudiante: Orlando Patricio Chacón Molina con código 42433.

Tema: "ANALISIS PARA UN SISTEMA DOMOTICO CON LA ARQUITECTURA ARDUINO Y RASBERRY PI, SOBRE TCP/IP"

Para: La obtención del título de Ingenieros de Sistemas

Director: Ing. Diego Chacón.

Tribunal: Ing. Kenneth Palacio.

DIA: Jueves

FECHA: 17 de Abril / 2014

HORA: 08h30



UNIVERSIDAD DEL
AZUAY

Cuenca, 23 de abril de 2014.

Ing. Javier Ortega

DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACION

De mis consideraciones:

Yo, Prof. Ing. Diego Chacón, en calidad de director de monografía del estudiante Orlando Patricio Chacón Molina (código 42433), me dirijo a usted con la finalidad de solicitar la aprobación del Diseño de monografía perteneciente al estudiante mencionado, quien previamente ha cumplido con las correcciones a las observaciones realizadas durante la defensa de su diseño.

Por la favorable acogida que se dé a la presente, anticipo mi agradecimiento.

Atentamente

Ing. Diego Chacón. Mhd



1 DATOS GENERALES

1.1 Nombre del estudiante:

Chacón Molina Orlando Patricio

1.2 Código:

42433

1.3 Contacto:

07.3.010.917 / 098.4.421.529 orlando@igotel.net

1.4 Director sugerido:

Ing. Chacón Diego Mdhd

1.5 Tribunal Designado

Ing. Kenneth Palacio

1.6 Línea de Investigación de la carrera:

1203 Informática de computadores

1.7 Área de estudio:

Domótica y Hogar Digital.

1.8 Título propuesto:

Análisis para un sistema domótico con la arquitectura Arduino y Raspberry Pi, sobre TCP/IP.

1.9 Estado del proyecto

Nuevo.



2 Contenido

2.1 Motivación de la investigación:

Las infraestructuras domóticas ofrecen un alto nivel de confort y control de un hogar; sin embargo, los costos elevados de los equipos hacen que su aplicación sea muy reducida. Por esta razón, se pretende crear una arquitectura basada en estándares de comunicación existentes y dispositivos de bajo costo, como Arduino y Raspberry Pi.

2.2 Problemática:

Actualmente, los equipos domóticos tienen un costo alto en el mercado, razón por la cual su implementación se ve limitada a un nicho elitista muy reducido. Una alternativa a esta restricción es el desarrollo de una arquitectura domótica abierta, basada en hardware y software libres. Inicialmente, hay que resolver las necesidades más comunes dentro de la automatización de hogares: iluminación, seguridad anti-intrusión y control de persianas.

La domótica es un área en la que se puede experimentar para crear tecnología propia como alternativa a las soluciones comerciales de alto costo. Se propone desarrollarla a partir de dispositivos existentes, modificando, configurando e integrando componentes que permitan desarrollar dispositivos para los que quizá no fueron pensados, como "piezas aisladas". Desde un punto de vista modesto, pero ambicioso, sería un avance significativo y a largo plazo constituirse en verdaderos desarrolladores de tecnología domótica.

2.3 Pregunta de investigación

¿Se puede realizar una implementación domótica con la arquitectura Arduino y Raspberry Pi, bajo el protocolo TCP/IP?

2.4 Resumen

Mediante esta investigación se pretende crear una arquitectura domótica con hardware y software libres, con una red de control que funcione sobre el protocolo de red TCP/IP. Se contemplarán sensores, actuadores, controladores y dispositivos controlados. Existirán dos tipos de controladores: Controladores Terminales (CT) y Controladores Centrales (CC). Los CT serán desarrollados en la plataforma de hardware Arduino, los cuales recolectarán la información de sensores, pulsadores y las órdenes procedentes de los sistemas informáticos. En base a la información recibida de los periféricos o CC, el CT enviará señales a los actuadores para que estos, a su vez, realicen los cambios sobre los dispositivos controlados. El CC estará desarrollado sobre Raspberry Pi, con el sistema operativo Raspbian, este podrá realizar consultas a cualquier CT dentro de las subredes que se establezcan y así determinar el estado de todos sus elementos. El CC, podrá configurarse en modo maestro o esclavo y dependiendo de esta, podrá presentar servicios web para que otros CC puedan obtener su información. La comunicación entre los CT y CC será establecida mediante el intercambio de mensajes en formato JSON.



2.5 Estado del Arte

Etimológicamente, domótica proviene del latín *domus* (casa) y del griego *αὐτόματος* (automática). Se usa el término para determinar el intrincado lugar de encuentro entre vivienda, tecnología y el ser humano. (Domínguez & Sáez Vacas, 2006, pág. 15)

Algunos autores expresan que la adopción domótica se remonta a los años sesenta cuando aparecieron los primeros dispositivos X-10, que funcionaban mediante las líneas eléctricas del hogar (Madrid vive ahorrando energía, 2007, pág. 15); sin embargo, con la popularización y evolución de los computadores personales y posteriormente los dispositivos móviles inteligentes, se ha dado lugar a al avance de los sistemas domóticos como KNX, Lonworks y un sinnúmero de otras arquitecturas, tanto abiertas como propietarias.

En la domotización de un hogar se involucran varios dispositivos eléctricos, electrónicos, mecánicos e informáticos trabajando para un mismo fin, la automatización y sofisticación del hogar. Entre los principales dispositivos, presentes en una implementación domótica, están:

Sensores: son dispositivos que recopilan información del entorno (presencia, calor, movimiento, humedad, etc.) y la transmiten a otros equipos para su procesamiento (Domínguez & Sáez Vacas, 2006, pág. 51). En base a esta información pueden tomarse decisiones sobre cambios de estado de equipos controlados o envíos de notificaciones.

Actuadores: son dispositivos capaces de recibir órdenes de los sistemas de control y en función de estas realizar cambios en los equipos, como encenderlos o apagarlos, incrementar o disminuir su nivel de intensidad y abrir o cerrar.

Transmisores: en esta categoría se encuentran los pulsadores, sistemas informáticos en dispositivos portátiles, pantallas táctiles y otros, los cuales reciben las instrucciones del usuario y las envían a la inteligencia del sistema para la posterior toma de decisiones.

Aparatos electrónicos inteligentes: son dispositivos con finalidades de ocio y entretenimiento.

Sistemas informáticos: sirven de apoyo a la domótica, mediante la presentación de interfaces para la comunicación con los actuadores, sincronización con otros sistemas.

Existen algunas consideraciones respecto a las redes presentes dentro de un hogar para que este pueda considerarse domótico y no solo automatizado, de este modo, las redes que deben estar presentes son:

Red de datos: está diseñada para la comunicación entre ordenadores y otros dispositivos informáticos, permitiendo, entre otros, el envío y recepción de mensajes, ficheros e internet. Esta debe ser independiente de la red de control y multimedia, con velocidades de transferencia de datos mayores a las de la red control (Plan Avanza, pág. 22).



Red Multimedia o de entretenimiento: su objetivo es la comunicación de audio y video, consolas de videojuego y otras plataformas de entretenimiento. Algunas de las aplicaciones soportadas por esta red son los videojuegos en red, la difusión de la señal de televisión de pago desde el descodificador al resto de estancias de la vivienda, el envío de audio y vídeo desde el portero automático (Domínguez & Sáez Vacas, 2006, pág. 67).

Red de Control: esta red se encarga de transportar el tráfico de control y monitoreo de sensores, actuadores y electrodomésticos de la vivienda. Aunque es independiente de la red de datos, generalmente existe comunicación entre las dos redes mediante una pasarela residencial, la cual es capaz de interactuar con los dos tipos de redes (Domínguez & Sáez Vacas, 2006, pág. 68).

Para el desarrollo de esta investigación se usarán tarjetas Arduino y equipos Raspberry Pi, a continuación se explican brevemente cada uno de ellos.

Según el sitio web oficial de las tarjetas Arduino, "Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP)." (Web Arduino)

Por otro lado está Raspberry Pi, el cual es un computador de bajo consumo eléctrico, con una arquitectura de procesador ARM. El sistema operativo de mayor popularidad para este computador es Raspbian OS, el cual es un sistema basado en Linux-Debian.

En el sitio oficial, en la sección de "FAQs", se encuentra la siguiente definición sobre Raspberry Pi "Raspberry Pi es un ordenador del tamaño de una tarjeta de crédito que se conecta a su televisor y un teclado. Se trata de un pequeño ordenador que puede ser utilizado en proyectos de electrónica, y para muchas de las cosas que hace su PC de escritorio, como hojas de cálculo, procesadores de texto y juegos. También reproduce vídeo de alta definición." (Web Raspberry Pi)

2.6 Hipótesis:

Se puede realizar el análisis para un sistema domótico con la arquitectura Arduino y Raspberry Pi, sobre TCP/IP.



2.7 Objetivo general:

Realizar el análisis para un sistema domótico con la arquitectura Arduino y Raspberry Pi, sobre TCP/IP.

2.8 Objetivos Específicos:

- Establecer las especificaciones para la comunicación física y lógica de dispositivos.
- Desarrollar el software para el control y comunicación de dispositivos
- Crear una Interfaz de Programación de Aplicaciones (API – Application Programming Interface) para que otros desarrolladores puedan ampliar las funcionalidades de control del sistema.
- Demostrar, mediante un prototipo, la arquitectura analizada.

2.9 Metodología:

Para el desarrollo de esta investigación se iniciará por establecer un laboratorio personal que constará de los elementos mencionados en la lista de presupuesto. Esta investigación tendrá una tendencia cualitativa, se trata de desarrollar una arquitectura domótica, pero no se contemplan aspectos cuantitativos más que de los equipos utilizados.

2.10 Alcances y resultados esperados

Definir y probar una arquitectura domótica usando herramientas de software y hardware abiertos (Arduino y Raspberry Pi). La comunicación entre los dispositivos se realizará mediante servicios web sobre el protocolo de red TCP/IP. La arquitectura debe permitir la autodetección de dispositivos domóticos dentro de una subred, el encendido, apagado y regulación de elementos eléctricos y/o electrónicos. Como resultado se tendrá un prototipo de automatización bajo la arquitectura desarrollada.

2.11 Supuestos y riesgos

Los riesgos identificados que podrían llevar al retraso o fracaso del proyecto son:

Fallas en el hardware debido a errores de fábrica o conexiones indebidas.

La escasez del hardware necesario en el mercado nacional.

2.12 Presupuesto

Rubro-Denominación	Costo USD (Detalle)	Justificación, para que
3 Tarjetas Arduino UNO	C. Unidad: \$35.00 C. Total: \$105.00	Para realizar las pruebas en un entorno lo más verosímil posible, teniendo varios actuadores, cada uno con múltiples elementos que estarán en comunicación con el mini-server.
3 Tarjetas Ethernet Shield para Arduino	C. Unidad \$16.00 C. Total \$48.00	Estas tarjetas servirán para permitir dotar de comunicación TCP/IP, mediante una conexión Ethernet.



UNIVERSIDAD
DEL
AZUAY

1 Raspberry Pi	C. Unidad	\$42.00	Se usará como mini-server del sistema. Para la finalidad de esta monografía podría funcionar como servidor cualquier computador con un SO/Linux. Sin embargo, un equipo de este tipo tiene un consumo eléctrico muy bajo, por lo que para implementaciones reales sería lo recomendable
Sensores	C. Aprox.	\$300.00	Se usaran para controles automáticos, como encendido o apagado en función de los valores leídos.
Resistencias, Relés, Pulsadores, Cables, Leds, diodos rectificadores, resistencias y otros.	C. Aprox.	200\$	Estos materiales y equipos servirán de complemento a las tarjetas Arduino, por ejemplo, los relés servirán para poder operar con voltajes altos, en corriente alterna.

2.13 Financiamiento

Se espera que la Universidad pueda facilitar algunas placas Arduino con los Shields Ethernet, pero de no ser posible se adquirirán por cuenta del estudiante.



2.14 Esquema tentativo

Abstract

Introducción

Objetivos

1 Capítulo 1. Fundamentos teóricos

1.1 *Introducción a la domótica*

1.2 *Componentes de una red domótica*

1.2.1 Sensores

1.2.2 Actuadores

1.2.3 Controladores

1.2.4 Aparatos electrónicos inteligentes

1.2.5 Sistemas informáticos

1.3 *Redes Domésticas*

1.3.1 Red de Control

1.3.2 Red de Datos

1.3.3 Red Multimedia

1.4 *Introducción a la plataforma Arduino*

1.5 *Introducción a la plataforma Linux (Raspbian OS sobre RaspberryPi)*

2 Capítulo 2 Definición de la arquitectura

2.1 *Arquitectura de Controladores Terminales*

2.1.1 Sensores

2.1.2 Actuadores

2.1.2.1 Encendido / Apagado

2.1.2.2 Incremento / Disminución

2.1.3 Controlador Terminal con Arduino.

2.2 *Transmisores*

2.2.1 Control Físico - Pulsadores

2.2.2 Control por Software

2.3 Circuitos Soportados

2.3.1 Iluminación (Encendido / Apagado)

2.3.2 Iluminación (Incremento / Disminución)

2.3.3 Climatización (Encendido / Apagado)

2.3.4 Riego (Encendido / Apagado)

2.3.5 Control de acceso

2.4 Arquitectura del Controlador Central

2.4.1 Estructura de Base de datos

2.4.2 Configuración y programación de Raspberry Pi como
Controlador Central

2.4.3 API para desarrolladores

3 Capítulo 3. Comunicación

3.1 TCP/IP

3.2 Servicios Web

3.2.1 Unidad de control

3.2.1.1 Recepción de órdenes.

3.2.1.2 Envío de notificaciones al servidor.

3.2.2 Mini Servidor

3.2.2.1 Envío de órdenes a unidades de control.

3.2.2.2 Recepción de notificaciones de la unidad de control.

3.2.2.3 Autodescubrimiento y reconstrucción del estado actual
del sistema.

3.2.2.4 Programador de acciones de acuerdo a comportamientos
anteriores.

3.2.2.5 API para desarrolladores

3.3 Interfaces de Hardware

3.3.1 Pulsadores



3.3.2 Actuadores

UNIVERSIDAD DEL AZUAY

3.3.2.1 Encendido / Apagado

3.3.2.2 Incremento / Disminución

4 Capítulo 4. Demostración de implementación y pruebas

4.1 Análisis de requerimientos

4.2 Diseño

4.3 Implementación

4.4 Pruebas

4.5 Mantenimiento

Conclusiones

Recomendaciones

Anexos

Bibliografía

2.15. Cronograma

Objetivo Especifico	Actividad	Resultado Esperado	Tiempo(Semanas)
<i>Establecer las especificaciones para la comunicación física y lógica de dispositivos.</i>	Definir las necesidades técnicas del sistema domótica	Conocer de manera informal lo que se espera del sistema y sobre estas bases proceder con el desarrollo técnico	3 días (3/7) Semana
	Establecer las especificaciones técnicas para la comunicación Física	Llegar a establecer una comunicación fiable a nivel físico entre dispositivos.	4 días (4/7) Semana
	Establecer las especificaciones técnicas para la comunicación Lógica	Determinar la forma en la que se comunicaran el mini servidor con las Unidades de control y viceversa	4 días (4/7) Semana
<i>Desarrollar el software para la el control y comunicación de dispositivos</i>	Creación de la base de datos en MySQL.	Establecer una base de datos unificada para que los proyectos dependientes de este puedan funcionar en correctamente(Esta parcialmente definida)	4 día (4/7) Semana



Desarrollar el software para las Unidades de Control(Arduino) el funcional para las unidades de control, de tal forma que pueda funcionar de manera independiente al mini servidor

7 días
(1) Semana

Desarrollar el software para el mini servidor e integrar con el Software de las unidades control Poder controlar de manera unificada, desde un servidor a todos las Unidades de control

7 días
(1) Semana

Crear una API para que otros desarrolladores puedan ampliar las funcionalidades de control del sistema.

Análisis y Diseño de servicios web. Conseguir un diseño previo de la manera en la que serán desarrollados los servicios web

3 días
(3/7) Semana

Desarrollo de servicios web para desarrolladores Crear una colección de servicios web que conforme una API para que nuevos desarrollos externos puedan ser incluidos al sistema

7 días
(1) Semana

Armar los circuitos eléctricos y electrónicos para las pruebas Obtener un prototipo funcional del sistema domótico a nivel físico

7 días
(1) Semana

Demostrar, mediante un prototipo, la arquitectura desarrollada.

Cargar el software en los dispositivos Lograr que el sistema funcione correctamente tanto mediante el control físico como por software (mediante URLs)

3 días
(3/7)

Reuniones con los encargados del proyecto de la aplicación móvil para las pruebas de funcionamiento. La aplicación debe recibir notificaciones del sistema domótico y viceversa para lograr una interacción real y dinámica.

7 días
(1) Semana



UNIVERSIDAD DEL AZUAY

3.1 Referencias

Domínguez, H. M., & Sáez Vacas, F. (2006). *Domótica: Un enfoque sociotécnico*. Madrid: Fundación Rogelio Segovia para el Desarrollo de las Telecomunicaciones.

Madrid vive ahorrando energía. (2007). *La domótica como solución de futuro*. Madrid.

Plan Avanza. (s.f.). *La casa digital*. Red.es.

Web Arduino. (s.f.). *Arduino - Home Page*. Recuperado el 14 de 03 de 2014, de Arduino - Home Page: <http://www.arduino.cc/es/>

Web Raspberry Pi. (s.f.). *FAQs | Raspberry Pi*. Recuperado el 14 de 03 de 2014, de FAQs | Raspberry Pi: <http://www.raspberrypi.org/faqs>

3.2 Firma de responsabilidad (estudiante)

Orlando Chacón Molina

3.3 Firma de responsabilidad (director sugerido)

Ing. Diego Chacón Mdh

3.4 Fecha de Entrega

23-04-2014

Oficio Nro. 034-2014-DIST-UDA

Cuenca, 09 de Abril de 2014

Señor Ingeniero
Xavier Ortega Vázquez
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 9 de Abril del 2014, revisó el proyecto de monografía titulado "Análisis para un sistema domótico con la arquitectura Arduino y Raspberry Pi, sobre TCP/IP", presentada por el estudiante Orlando Chacón, estudiante de la Escuela de Ingeniería de Sistemas, previo a la obtención del título de Ingeniero de Sistemas.

La Junta considera que el diseño de trabajo de titulación cumple con los requisitos normados en la "Guía de Elaboración y Presentación de la Denuncia/Protocolo de Trabajo de Titulación", razón por la cual solicita, por su digno intermedio, notificar al tribunal designado y determinar lugar, fecha y hora de sustentación.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomienda como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Diego Chacón (docente del curso de graduación), y como miembro del Tribunal al Ing. Kenneth Palacio.



Atentamente,

Ing. Marcos Orellana Cordero
Director Escuela de Ingeniería de Sistemas y Telemática
Universidad del Azuay

Cuenca, 10 de abril de 2014

Ing. Javier Ortega

DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

De mis consideraciones:

Yo, Orlando Patricio Chacón Molina, con CI. 0104817358 y código 42433, egresado de la Escuela de Ingeniería de Sistemas me dirijo a usted con la finalidad de solicitar la aprobación de mi diseño de monografía titulado "*Análisis para un sistema domótico con la arquitectura Arduino y Raspberry Pi, sobre TCP/IP.*", correspondiente al curso de grado tomado en el periodo 2013-2014.

Por la favorable acogida que se dé a la presente, anticipo mi agradecimiento.

Atentamente:



Orlando P. Chacón Molina

CI: 01048173580

Cod: 42433



UNIVERSIDAD DEL
AZUAY

DOCTORA JENNY RIOS COELLO SECRETARIA, DE LA FACULTAD DE CIENCIAS
DE LA ADMINISTRACIÓN DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, el Señor **Orlando Patricio Chacón Molina**, registrado con el código **42433**
perteneciente a la Escuela de Sistemas, luego de cumplir con todas las asignaturas de su
Pensum de estudios, egresó de la Facultad en el día 08 de Agosto de 2012.

Cuenca, Abril 10 del 2014

UNIVERSIDAD DEL
AZUAY
FACULTAD DE
ADMINISTRACION
SECRETARIA

Derecho 54599

vcf.-

