

Sistema Domótico Distribuido para Controlar el Riego y el Aire Acondicionado en el Hogar

Iván Manuel Laclaustra⁺, Jesús Martín Alonso⁺, Alberto A. del Barrio⁺, Guillermo Botella⁺

Departamento de Arquitectura de Computadores y Automática, Facultad de Informática de la Universidad Complutense de Madrid
Madrid, España
+ {ilaclus, jesusm01, abarriog, gbotella} @ ucm.es

Resumen. Este trabajo presenta el proyecto *SEDomotics*, realizado en la asignatura Sistemas Empotrados Distribuidos, perteneciente a la titulación del Máster en Ingeniería Informática de la Universidad Complutense de Madrid. En este trabajo se describe e implementa una plataforma de control domótico de los sistemas de riego y aire acondicionado en el hogar, utilizando para ello dos placas Arduino y una Raspberry Pi como servidor. Además de capturar los datos en tiempo real, el sistema es capaz de almacenar un histórico con dichos datos.

Palabras Clave: Sistemas Empotrados Distribuidos, Internet of Things, Arduino, Raspberry Pi, Domótica, Sistema de riego.

Abstract. This work presents the *SEDomotics* project, developed in the Distributed Embedded Systems subject, allocated within the Computer Science Master, which is taught in the Universidad Complutense de Madrid. This work describes and implements a distributed domotic system for the irrigation and air conditioner at home, using for this two Arduino boards and a Raspberry Pi as a server. In addition to capturing real time data, the system is capable of storing a record with that data.

Keywords: Distributed Embedded Systems, Internet of Things, Arduino, Raspberry Pi, Domotic, Irrigation System.

1 Introducción

Desde la llegada de Internet a nuestras vidas, vivimos en un mundo cada vez más interconectado. La tecnología nos ha proporcionado la capacidad de comunicarnos salvando grandes distancias y a una gran velocidad. El rápido desarrollo de las tecnologías móviles durante los últimos años ha hecho que las posibilidades aumenten

exponencialmente, y ha cambiado por completo nuestra forma de ver el mundo. Además, el hecho de que los costes de producción se estén reduciendo paulatinamente ha traído al frente un nuevo paradigma, el llamado Internet de las Cosas (IoT) [12-13], que está llamado a hacerse realidad en los años venideros.

En el paradigma del IoT, todos los objetos de uso común están conectados a Internet de forma distribuida [15,17], de manera que pueden proporcionar o recibir información relevante para su tarea. Podemos encontrar un ejemplo reciente en los SmartWatch que, además de ofrecer lo que un reloj convencional, pueden mostrarnos mensajes de nuestro móvil, ritmo cardíaco [14,16], llamadas entrantes, etc. Otra ventaja de este enfoque, es que los objetos podrían comunicarse y coordinarse entre sí. Aunque aún estamos muy lejos de un uso regularizado, desde hace algunos años se está trabajando en la *domótica* [20-21]. La domótica se encarga del control del hogar a distancia. Mediante una conexión a Internet, podríamos encender o apagar luces, televisores, altavoces, encender un hornillo en la cocina, etc. Estas ideas sobre IoT, domótica, smart devices, entre otras, son las que se estudian en la asignatura *Sistemas Empotrados Distribuidos (SED)*, materia ubicada en el Máster de Ingeniería Informática de la Universidad Complutense de Madrid (UCM) [22], y en la cual se realizó este trabajo.

Una de las funcionalidades domóticas más extendidas es la referente al control del riego del jardín. Dado que el riego supone uno de los mayores gastos de agua en un hogar que lo posea, es importante mantener un uso óptimo del mismo, y para ello la domótica es ideal. Con ella, podríamos hacer cosas como programar un riego automático, abrir y cerrar válvulas o comprobar el estado del césped, con la ventaja de poder hacerlo a distancia. Sin embargo, el principal problema es que se hace necesaria la instalación de un equipamiento especial, que servirá de interfaz para que los objetos correspondientes se comuniquen satisfactoriamente. Esto hace que el precio de implantación se dispare. En el caso de los sistemas de riego, suelen rondar los 200€ [1] y el precio aumenta en el caso de querer un sistema domótico completo [2]. En dichos sistemas se ofrecen otras funcionalidades, como es el caso del control automatizado del aire acondicionado.

Debido al alto coste de los sistemas comerciales tradicionales, una buena solución es el uso de placas de bajo coste para la implementación. Por ejemplo, las placas Arduino [3] tienen un microcontrolador integrado con varios pines y otros componentes, todos programables mediante un lenguaje de alto nivel. Además, es posible conectar fácilmente multitud de sensores y otros sistemas. El lenguaje de programación utilizado es de uso común (C). Tanto las placas (a partir de 20€ las oficiales) como los sensores (unos 3€) tienen un precio asequible.

Por tanto, en este artículo se presenta un sistema domótico distribuido para controlar tanto el sistema de riego de una casa como automatizar el uso inteligente del aire acondicionado. En las siguientes secciones, mostraremos una implementación basada en Arduino [3] y Raspberry Pi [4].

El resto del artículo se organiza de la siguiente forma: la Sección 2 describe tanto el diseño del sistema como los detalles de la implementación del mismo. En la Sección 3 se muestran los resultados obtenidos y se describen los experimentos realizados para validar el sistema y finalmente en la Sección 4 se presentan nuestras conclusiones.

2 Propuesta

El sistema descrito en este artículo representa el control domótico de una casa, como muestra la Figura 1. La plataforma propuesta será capaz de controlar el sistema de riego y el del aire acondicionado, de forma que puedan encenderse/apagarse a distancia. Además incluirá un modo automático, de manera que los sistemas conectados se enciendan o apaguen en función del estado exterior (humedad/temperatura).

Por tanto, en el sistema propuesto se diferencian dos componentes principales:

- **Subsistema exterior.** Está conectado al sistema de riego de la vivienda. Se encarga de monitorizar los datos de temperatura exterior y de la humedad del suelo y en función de ellos activa/desactiva el sistema de riego según convenga.
- **Subsistema interior.** Está conectado al sistema de aire acondicionado de la vivienda. Se encarga de monitorizar los datos de la temperatura y humedad interior y en función de ellos activa/desactiva el sistema de aire acondicionado según convenga.

Además se plantea la creación de un centro de control, donde el usuario pueda ver en tiempo real los datos registrados por ambos subsistemas, a la vez que controlar los elementos que integran los subsistemas manualmente y consultar un histórico de los datos registrados por los sensores.

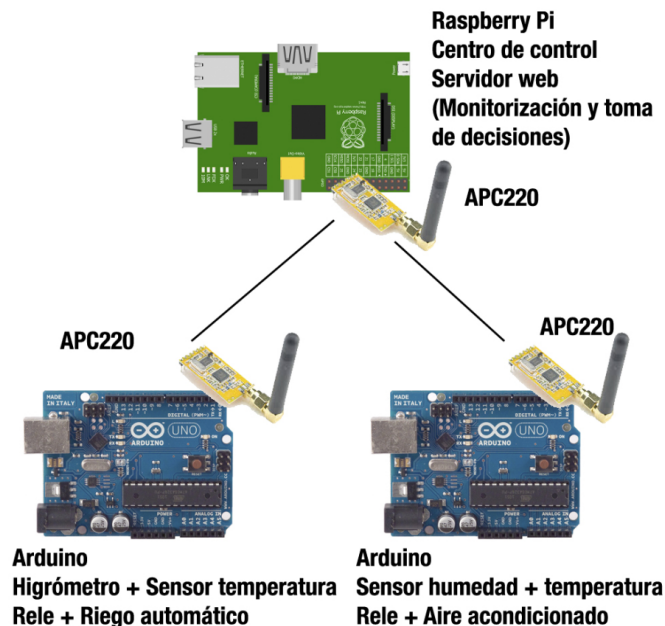


Figura 1. Esquema general del sistema domótico de control del riego y el aire acondicionado

2.1 Hardware empleado

Como muestra la Figura 1, ambos *subsistemas* se implementaron con placas Arduino Uno [3], ya que no requieren una alta potencia computacional. Por el contrario, para el '*centro de control*' se escogió una Raspberry Pi 1 Model B [4], que tiene un microprocesador lo suficientemente potente como para alojar el servidor web y la base de datos. Para completar el proyecto se escogieron los siguientes módulos '*plug & play*' para Arduino:

- Para la comunicación se escogieron módulos de radio APC220, por el bajo precio y su alcance (hasta 1000 metros).
- Un sensor de humedad y temperatura *DHT-11 Keyes* [24]. Este sensor se ha utilizado en el *subsistema interior* para monitorizar la temperatura y el grado de humedad, de tal forma que los datos sensados ayuden a tomar decisiones sobre el sistema de aire acondicionado de la casa.
- Un sensor de temperatura *Keyes DS18B20* [25]. Dicho sensor se ha utilizado en el *subsistema exterior* para monitorizar la temperatura.
- Un sensor de humedad del suelo *YL-69* [26]. Este sensor se ha utilizado en el *subsistema exterior* para monitorizar la humedad del suelo. A partir de los datos obtenidos se tomarán decisiones sobre el sistema de riego.
- Dos módulos de relés *Keyes* [27]. Los relés se han utilizado en ambos *subsistemas* para activar/desactivar los elementos asociados a ellos (aire acondicionado y riego).

2.2 Comunicación entre los componentes del Sistema

La comunicación por cable parece poco adecuada para un sistema domótico, ya que las distancias pueden ser elevadas y la instalación bastante costosa. De esta manera, nos decantamos por elegir comunicación inalámbrica. En concreto utilizamos los módulos de radiofrecuencia APC220 [5], ya indicados anteriormente. Decidimos utilizar radiofrecuencia debido a la simplicidad de uso (utilizando el puerto serie), a su bajo coste y al alto alcance que tienen los módulos APC220.

Estos módulos permiten la comunicación hasta una distancia de 1000 m (sin obstáculos) y pueden configurar su frecuencia (418-455 MHz), así como otras opciones.

La configuración de los módulos se realizó mediante el programa *RFMagic*, de acuerdo a los parámetros mostrados por la Tabla 1.

Tabla 1. Parámetros de configuración de los módulos RF APC220, realizada mediante el programa 'RFMagic'

Frecuencia	433 MHz
Tasa de transmisión de datos	9600 bps
Bit de paridad	Desactivado

Estos módulos se conectan a los puertos serie de las placas Arduino, así como de la Raspberry Pi.

La Raspberry ejerce el control sobre los subsistemas, enviando las órdenes necesarias. De esta manera, cuando se necesitan los datos de los sensores o de los sistemas, la Raspberry envía una orden y las placas Arduino Uno la reciben y contestan. Por motivos de complejidad, se trata de un protocolo sin comprobación de errores. Es decir, no se comprueba que los datos lleguen correctamente o si las placas están apagadas. Por último, al ser un medio de difusión, en cada placa se comprueba si la trama recibida va destinada a ella. A continuación se detalla el formato de las tramas intercambiadas:

Tabla 2. Resumen del formato de las tramas intercambiadas

Trama	Campos
Trama de envío de información del sistema	<i>Identificador de placa</i>
	<i>Temperatura leída por el sensor</i>
	<i>Humedad leída por el sensor</i>
	<i>Estado del relé conectado a la placa:</i> <i>'on' / 'off'</i>
Trama de encendido/apagado de sistemas	<i>"relay"</i>
	<i>Identificador de placa</i>
	<i>Estado del relé conectado a la placa:</i>

	<i>'on' / 'off'</i>
Trama de petición de datos	<i>getData</i>
Trama de puesta en modo manual/automático	<i>manual</i>
	<i>Modo al que se quiere cambiar</i> <i>'ON' para activar modo manual</i> <i>'OFF' para activar modo automático</i>

- **Trama de envío de información del sistema.** Contiene la información del sistema. En el caso del subsistema interior enviará la temperatura y la humedad ambiente y en del subsistema exterior enviará la temperatura y la humedad del suelo. Consiste en una línea (terminada en \n) con campos separados por el carácter ';', como muestra la Tabla 2. El formato es el siguiente:

1 ; temperatura ; humedadAmbiente/Suelo ; estadoSistema

Ejemplo de trama:

1;25;30;off

- **Trama de encendido/apagado de sistemas (Centro de control y subsistema interior/exterior).** El centro de control (Raspberry) envía esta trama cuando el usuario quiere encender/apagar los sistemas de forma manual. A su vez, la trama es enviada por parte de los subsistemas cuando se produce un cambio en el relé, bien debido a una orden del centro de control, o bien por decisión de la placa, en caso de que se encuentre en modo automático. Consiste en una línea (terminada en \n) con campos separados por el carácter ';', como puede observarse en la Tabla 2. El formato es el siguiente:

relay ; identPlaca ; estadoSistema

Ejemplo de trama:

relay;2;on

- **Trama de petición de datos (Centro de control).** El centro de control (Raspberry) envía esta trama cuando necesita obtener los datos de los sensores y sistemas conectados a los Arduinos. Consiste en una línea (terminada en \n):

getData

La respuesta a esta trama es una trama de envío de información del sistema, descrita anteriormente.

- **Trama de puesta en modo manual/automático (Subsistema interior/exterior y Centro de control).** El centro de control (Raspberry) envía esta trama cuando el usuario cambia de modo automático a manual o viceversa. Por su parte, los subsistemas, al encenderse, envían esta petición (manual;ON), para que el resto (Raspberry y Arduinos restantes) se configuren en modo manual. De esta manera se evitan inconsistencias en el sistema. Consiste en una línea (terminada en \n) con campos separados por el carácter ‘;’, tal y como muestra la Tabla 2. El formato es el siguiente:

manual;orden

Ejemplo de trama:

manual;ON

(Activa el modo manual)

2.3 Desarrollo de los subsistemas

Como se comenta en la Sección 2.2, la implementación de los subsistemas se realiza mediante el uso de placas Arduino Uno. A continuación se detalla su funcionamiento.

Al encenderse un subsistema, envía una petición para que el sistema entre en modo manual. De esta forma, conseguimos que no haya incoherencias entre las distintas placas. Por defecto, los relés están apagados. Para finalizar, tras configurar los sensores propios de cada subsistema, se envía una trama de información al controlador, para que éste actualice los datos en la página de control.

Una vez realizadas las configuraciones iniciales, ambas placas esperan la llegada de mensajes a través del puerto serie. Cuando se recibe un mensaje, cada placa comprueba si va dirigido a ella, y responde en consecuencia. Por otro lado, en caso de que el subsistema se encuentre en modo automático, llevará a cabo comprobaciones sobre la información de sus sensores, actuando en consecuencia. Por ejemplo, en el caso del subsistema exterior, el riego se activará automáticamente siempre que

estemos en modo automático y la humedad baje del 35%, y se desactivará si sube de 70%.

Para facilitar la implementación de los subsistemas, se usaron librerías externas en algunos de los sensores:

- **OneWire [6]**. Esta librería sirve para comunicarse con dispositivos que utilicen el protocolo *1-wire*. En nuestro caso, el sensor de temperatura del subsistema exterior.
- **DallasTemperature [7]**. Esta librería hace uso de la OneWire indicada anteriormente. Se utiliza para simplificar el uso de dicha librería, ya que es relativamente complejo.
- **DHT [8]**. Esta es la librería proporcionada por el fabricante para utilizar el sensor de humedad+temperatura del subsistema interior.

2.4 Desarrollo del centro de control

Mediante el centro de control el usuario podrá ver toda la información de los subsistemas (sensores y estado de los sistemas) en tiempo real mediante una página web. Además podrá controlar los subsistemas manualmente desde la misma página.

Se decidió utilizar como servidor *Node.js* [9] debido a su gran variedad de módulos y su facilidad de programación (javascript). Para construir la página web se utilizó Express [10], un framework que facilita la construcción de webs en Node.js. Para la comunicación serie se instaló el módulo *serialport*, que permite utilizar el puerto serie. Para resolver el problema de la visualización en tiempo real se utilizó *Socket.io* [11], que permite enviar datos desde el servidor (Raspberry) al cliente (ordenador de usuario), de manera que los cambios se reflejen de manera inmediata. Por último para guardar los datos, y poder mostrar un histórico se desarrolló una base de datos MySQL [23].

Para mostrar un aspecto visual atractivo se utilizaron los siguientes librerías/componentes:

- Bootstrap para el aspecto visual de la web.
- Highcharts para las gráficas.
- Bootstrap switch para el modo manual.
- JustGage para los marcadores de temperatura/humedad.

El funcionamiento general del servidor es el siguiente:

- Cada 10 minutos se solicitan datos a los subsistemas para almacenarlos en la base de datos y construir el histórico.
- Cuando el usuario está en la página de monitorización se solicitan datos con un intervalo de 1 segundo, para que se refleje cualquier cambio de manera inmediata.
- Cuando el usuario está en modo manual y apaga/enciende un subsistema, se envía una trama al Arduino correspondiente para que actúe en consecuencia.
- Si recibimos una trama "manual;ON", activamos el modo manual.
- Cuando el usuario está en la página histórico, se realizan consultas a la base de datos para construir gráficas.

3 Resultados y experimentos

En esta sección se proporciona el resultado de la implementación del sistema.

3.1 Implementación hardware

En la Figura 2 se puede apreciar el resultado del montaje del prototipo del subsistema exterior. En la Figura 3 se observa el resultado del montaje del prototipo del subsistema interior y finalmente, en la Figura 4, se observa el prototipo hardware del centro de control. Dichas figuras presentan los elementos anteriormente descritos.

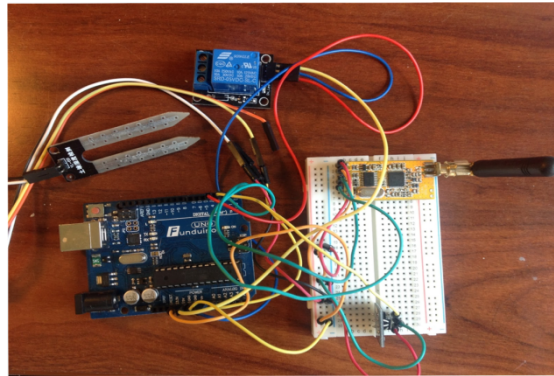


Figura 2. Prototipo del subsistema exterior

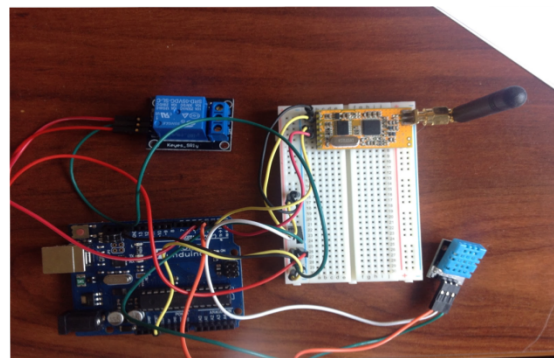


Figura 3. Prototipo del subsistema interior



Figura 4. Prototipo del centro de control

3.2 Implementación Software

En la Figura 5 se puede apreciar la página web que muestra la información de los sensores y el estado de los subsistemas (interior y exterior). En la Figura 6 se observa la página web que proporciona el histórico de los datos recogidos por los sensores. En dicha página se muestran cuatro gráficas: temperatura exterior, humedad del suelo, temperatura interior y humedad Interior. Por defecto, se muestran los datos recogidos en el último día. Sin embargo, el usuario puede elegir para cada gráfica mostrar los datos de la última semana, último mes, o todos los datos registrados.



Figura 5. Vista de la página web que muestra los datos correspondientes a los subsistemas en tiempo real



Figura 6. Vista de la página web que muestra el histórico de los datos recogidos por los sensores

3.3 Experimentos

Para validar el sistema se realizaron los siguientes tests:

- Comprobación del funcionamiento modo automático y de monitorización en tiempo real.

Como se ha indicado anteriormente, el sistema posee un *modo automático* en el que los subsistemas activan/desactivan los elementos que controlan en función de los datos recogidos por los sensores. Para comprobar el correcto funcionamiento se realizaron los siguientes pasos:

1. Cambiar sistema a modo automático.
2. Entrar en el apartado monitorización del servidor.
3. Comprobar subsistema interior. Se incrementó tanto la temperatura como la humedad y se comprobó que los valores recogidos cambiaban automáticamente en la página web, y de manera consistente. Además, al llegar al umbral de temperatura establecido se vio cómo se activaba automáticamente el relé que controla el sistema de aire acondicionado.
4. Comprobar subsistema exterior. Con el sensor de humedad de suelo introducido en tierra húmeda, se procedió a extraerlo poco a poco. Se pudo comprobar cómo los valores relativos a la humedad del suelo van

disminuyendo en la página web. De la misma manera, se observó cómo al sacar por completo el sensor de humedad del suelo, se activaba automáticamente el relé que controla el sistema de riego.

- Comprobación del modo manual.

El modo manual, permite controlar (activar/desactivar) el sistema de riego y el sistema de aire acondicionado manualmente, a través de la opción *control manual*, presente en la página web. Para comprobar su correcto funcionamiento se tomaron los siguientes pasos:

1. Entrar en la sección del servidor *control manual*, a través de la página web.
2. Cambiar el sistema a *modo manual*.
3. Activar y desactivar varias veces el sistema de riego, comprobando que se activaba/desactivaba el relé correspondiente de forma correcta.
4. Activar y desactivar varias veces el sistema de aire acondicionado, comprobando que se activaba/desactivaba el relé correspondiente de forma correcta.
5. Volver a poner el sistema en *modo automático*.

- Comprobación del histórico.

El sistema ofrece la posibilidad de observar un histórico de los datos recogidos por los sensores de los subsistemas mediante gráficas. Para comprobar el correcto funcionamiento de esta característica se realizaron los siguientes pasos:

1. Se dejaron activados los sistemas durante varias horas.
2. Se accedió al apartado *histórico* de la página web
3. Se comprobó en las gráficas que se habían registrado datos durante el periodo de tiempo en el que el sistema había estado activo.

4 Conclusiones

En este artículo se ha presentado un sistema domótico basado en una plataforma distribuida. Dicho sistema se realizó como proyecto para la asignatura de Sistemas Empotrados Distribuidos del Máster de Ingeniería Informática de la Universidad Complutense de Madrid.

Mediante el uso de plataformas de hardware como Arduino y Raspberry Pi, se implementó un sistema domótico inalámbrico y distribuido que permite el control de los sistemas de riego y de aire acondicionado del hogar.

De cara al futuro, se plantean varios puntos, como la elaboración de un segundo prototipo más avanzado que incluya alimentación por batería, la posibilidad de configurar los parámetros del sistema o la elaboración de una aplicación móvil que complementa a la página web elaborada.

Como conclusión final se ha de decir que este proyecto nos ha dado la oportunidad de aplicar en un caso real los conceptos vistos en clase, lo que incrementado nuestra motivación en el ámbito de los Sistemas Empotrados Distribuidos.

5 Trabajo futuro

Debido al limitado tiempo del que disponíamos, sólo pudimos realizar un prototipo limitado del sistema, que nos sirvió como prueba de concepto para sistemas mayores. Por ello, a continuación citamos algunas mejoras que nos hubiera gustado incluir de haber tenido más tiempo:

5.1 Ampliación de dispositivos

Además de los dispositivos de control tradicionales para riego y aire acondicionado, sería interesante añadir otros dispositivos, como por ejemplo:

- Estaciones meteorológicas: podrían monitorizar el tiempo para actuar automáticamente sobre los controladores de los demás dispositivos
- Cerrojos para puertas y ventanas: control a distancia de los cerrojos de la casa.
- Detectores de presencia: combinado con alarmas, podría cerrar todos los cerrojos de la casa en caso de saltar la alarma.

5.2 Reestructuración en módulos

Aunque nuestra implementación destaca por su sencillez, es necesario modificar el código en diferentes puntos a la hora de añadir más elementos. Esto hace que su ampliación o modificación sea un trabajo arduo para usuarios ajenos al sistema. Por ello, sería interesante reestructurar el sistema para hacerlo más modular, de forma que la adición de nuevos módulos sea más directa.

5.3 Configurabilidad

Actualmente, los dispositivos incluidos en el sistema tienen un funcionamiento muy concreto, que vino dado por nuestras especificaciones. Esto no tiene por qué satisfacer a todos los usuarios, por lo que sería interesante poder cambiar los parámetros del sistema (como porcentaje de humedad a partir del cual el riego para).

Otra mejora sería tener un mapa con los módulos que tenemos instalados en nuestro sistema desde la página web, pudiendo acceder a la configuración de cada uno

de ellos fácilmente, o desconectarlos por completo (junto con todos sus componentes).

5.4 Aplicaciones móviles

Dado que no se nos exigía portabilidad para dispositivos móviles, nuestro sistema no está preparado para ello. Debido a que hoy en día las tecnologías móviles están en pleno crecimiento, sería provechoso implementar una versión para móviles de nuestra aplicación, con todas las funcionalidades disponibles actualmente.

Referencias

1. Control de riego GreenIQ Smart Garden Hub, <https://domboo.es/producto/control-de-riego-greeniq-smart-garden-hub/>. Accedido el 14/03/2016.
2. Sistema domótico Control4, <http://www.control4.com/o/new-entertainment-and-automation> . Accedido el 14/03/2016.
3. "Arduino". <https://www.arduino.cc/> . Accedido el 14/03/2016.
4. "Raspberry Pi". <https://www.raspberrypi.org/> . Accedido el 14/03/2016.
5. "APC220 Radio Data Module wiki". http://www.dfrobot.com/wiki/index.php?title=APC220_Radio_Data_Module%28SKU:TEL0005%29 . Accedido el 14/03/2016.
6. "Documentación librería OneWire". <http://playground.arduino.cc/Learning/OneWire> . Accedido el 14/03/2016.
7. "Documentación librería Dallas Temperature". https://milesburton.com/Dallas_Temperature_Control_Library . Accedido el 14/04/2016
8. "Midiendo temperatura y humedad con Arduino y el sensor DHT11", <http://www.internetdelascosas.cl/2014/07/08/midiendo-temperatura-y-humedad-con-arduino-y-el-sensor-dht11/> . Accedido el 14/03/2016.
9. "Node.js" <https://nodejs.org/en/> . Accedido el 14/03/2016.
10. "Express" <http://expressjs.com/es/> . Accedido el 14/03/2016.
11. "Socket.io" <http://socket.io/> . Accedido el 14/03/2016.
12. O. Vermesan and P. Friess "Internet of Things - Global Technological and Societal Trends From Smart Environments and Spaces to Green ICT", 2011, River Publishers.
13. H. Chaouchi, "The Internet of Things: Connecting Objects", 2013, John Wiley & Sons.
14. R. Braojos *et al.*, "Ultra-low power design of wearable cardiac monitoring systems," *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, San Francisco, CA, 2014, pp. 1-6.
15. A. A. Del Barrio, J. Cong and R. Hermida, "A Distributed Clustered Architecture to Tackle Delay Variations in Datapath Synthesis," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 3, pp. 419-432, March 2016.
16. A. Dias Junior, S. Murali, F. Rincon and D. Atienza, "Estimation of Blood Pressure and Pulse Transit Time Using Your Smartphone," *Digital System Design (DSD), 2015 Euromicro Conference on*, Funchal, 2015, pp. 173-180.
17. A. A. Del Barrio, S. O. Memik, M. C. Molina, J. M. Mendias and R. Hermida, "A Distributed Controller for Managing Speculative Functional Units in High Level Synthesis," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 3, pp. 350-363, March 2011.

18. D. Lora et al., "Sistema de Seguridad Basado en una Plataforma Heterogénea Distribuida", Enseñanza y Aprendizaje de Ingeniería de Computadores, 5: 29-38 (2015).
19. F. Parrales et al. "Una Orquesta Sinfónica como Ejemplo de Aplicación de un Sistema Empotrado Distribuido", Enseñanza y Aprendizaje de Ingeniería de Computadores, 5: 115-124 (2015).
20. A. Varma, "Domotics: Smart Technology, Smarter Homes", 2009, Icfai University Press.
21. R. Carbou et al., "Digital Home Networking", 2013, John Wiley & Sons.
22. Máster en Ingeniería Informática de la Universidad Complutense de Madrid.
<http://informatica.ucm.es/estudios/master-ingenieriainformatica>. Accedido el 10/05/2016
23. "MySQL". <https://www.mysql.com/>. Accedido el 14/03/2016.
24. "DHT11 Humidity & Temperature Sensor" <http://www.micropik.com/PDF/dht11.pdf>.
Accedido el 11/04/2016
25. "Keyes DS18B20 Sensor Module", http://misclab.umeoce.maine.edu/boss/Arduino/bensguides/KeyesDs18b20_Digital_Temperature.pdf . Accedido el 11/04/2016
26. "Módulo YL-69 y YL38". <https://es.scribd.com/doc/233114065/Modulo-YL69-y-YL38>.
Accedido el 11/04/2016
27. "Keyes 5V Relay Module" <http://tinkbox.ph/sites/tinkbox.ph/files/downloads/KEYES%205V%20Relay%20Module%20KY-019.pdf>. Accedido el 11/04/2016