

Diseño de la Maqueta Domótica para el Aprendizaje de Sistemas de Automatización Domótica

Juan A. Holgado-Terriza

Universidad de Granada
C/ Periodista Daniel Saucedo Aranda s/n 18071,
Granada, España,
jholgado@ugr.es

Resumen. La utilización de recursos didácticos que permitan a los alumnos disponer de herramientas que fomenten la innovación y la aplicación de los conocimientos adquiridos durante su formación, es una forma muy interesante de potenciar tanto su interés como su participación proactiva en el proceso de aprendizaje. En este trabajo se presenta la experiencia que ha adquirido el grupo de investigación de Sistemas Concurrentes de la Universidad de Granada en la construcción de maquetas domóticas como herramientas docentes que pueden favorecer el proceso de aprendizaje a los alumnos en carreras de carácter técnico, como son la ingeniería informática o la electrónica.

Palabras Clave: Docencia, domótica, maqueta, enseñanza, informática, electrónica.

Abstract. The use of teaching tools that allow students to have available tools which encourage innovation and the application of the knowledge acquired during their training, is a very interesting way to promote both their interests and their proactive participations in the learning process. In this work, we present the experience acquired by the research group of “Concurrent Systems” at the University of Granada in the construction of home automation scale models as teaching tools that can facilitate the active learning process for students in technical careers such as computer engineering or electronics.

Keywords: Teaching, home automation, scale model, computer engineering, electronics

1 Introducción

El aprendizaje en campos como la automática, control de procesos o la programación de sistemas requiere contar con herramientas docentes que faciliten la elaboración de diversos tipos de actividades que motiven y acerquen a los alumnos a problemas reales, haciéndoles protagonista del proceso de aprendizaje para que puedan encontrar sus posibles soluciones. Si además las soluciones pueden probarse sobre plataformas reales, esto permite a los alumnos comprender mejor la utilidad de los conocimientos adquiridos, y a su vez, adquirir competencias para su futuro profesional.

La domótica ofrece en la actualidad un campo de aplicación particularmente interesante desde el punto de vista tecnológico e industrial, ya que se están produciendo avances significativos desde los tradicionales sistemas de automatización del hogar en el que conviven dispositivos domóticos con servicios automáticos como los climatizadores, los detectores o los sistemas de vigilancia, a los más avanzados sistemas de inteligencia ambiental capaces de adaptarse y anticiparse a las necesidades del usuario de forma autónoma sin necesidad de ser programados; sistemas que, además son cada vez más accesibles desde cualquier tipo de terminal fijo, portátil o móvil.

En este contexto resulta particularmente propicio para la formación de los alumnos poder conjugar de forma óptima aspectos como el control industrial, las arquitecturas de telecomunicaciones, el diseño de nuevos dispositivos domóticos, la programación de sistemas o la aplicación de técnicas de inteligencia artificial. Por este motivo, es importante encontrar nuevas herramientas docentes donde pueda reproducirse en una escala manejable para los alumnos el entorno de una vivienda, y posibilitar el desarrollo de aplicaciones tal y como se podrían realizar sobre una instalación real.

Habitualmente se suelen emplear dos tipos de herramientas docentes como son los simuladores y las maquetas didácticas. Los simuladores proporcionan un entorno de trabajo controlado de la realidad en el que se pueden establecer distintos tipos de escenarios como casas adaptadas con el conjunto de dispositivos, sensores, actuadores o controladores específicos. Son sistemas fácilmente configurables y completamente accesibles al ejecutarse en el contexto de una aplicación que puede ejecutarse sobre diferentes tipos de plataformas software. Sin embargo, sólo son capaces de proporcionar modelos parciales de la realidad, no ofreciendo el grado de realismo suficiente para que la interacción con el alumno pueda ser rica y completa.

En cambio, las maquetas físicas didácticas a diferencia de los simuladores permiten desarrollar experiencias docentes más amplias y ricas para todos los actores del proceso de enseñanza/aprendizaje, fomentando el aprendizaje activo del alumno. En este sentido la maqueta domótica da la oportunidad de enfrentarse a los problemas prácticos de manejo de dispositivos reales, los problemas derivados del diseño, implementación y depuración de un sistema completo integrando los aspectos hardware y software, lo que proporciona una experiencia docente mucho más completa que aumenta la competencia de los alumnos para afrontar proyectos completos tal y como se encontrarán en su futuro laboral.

En este trabajo se presenta la experiencia que ha adquirido el grupo de investigación de Sistemas Concurrentes (TIC-157) del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada en la construcción de maquetas domóticas con fines docentes desde el año 2004 hasta hoy en día. Durante estos años se han construido hasta tres maquetas completas y se han realizado distintos tipos de adaptaciones tanto a nivel electrónico, físico, arquitectónico y lógico para adaptarse a distintos tipos de experiencias docentes. Hemos propuesto lo que denominamos la *Maqueta Domótica Didáctica* (MDD) como un modelo de diseño de construcción de maquetas domóticas a escala, que determina la estructura de la maqueta tanto a nivel físico como a nivel lógico, y los pasos a realizar en el proceso de construcción.

Dado que el propósito de la maqueta es proporcionar un entorno donde el alumno pueda probar el desarrollo de aplicaciones a distintos niveles de abstracción proponemos la utilización de un lenguaje de alto nivel común como Java. Se ha

seleccionado Java como lenguaje de programación principal por las características específicas que tiene el lenguaje como la portabilidad, la capacidad de abstracción en objetos, la modularidad, el encapsulamiento y las facilidades para la programación como la ligadura dinámica, la carga dinámica de clases o el recolector de basura. Además permite utilizar el mismo lenguaje en cualquier ámbito de desarrollo, tanto a bajo nivel en el diseño de un driver como a alto nivel en el diseño de un servicio web. Por otra parte, contamos con herramientas de desarrollo muy estandarizadas que facilitan la implementación de las aplicaciones, su depuración y despliegue sobre diferentes tipos de entornos (microcontroladores, sistemas web, middleware, etc.) en el que se cargará un entorno de ejecución basado en una JVM (Java Virtual Machine) para la ejecución directa de las aplicaciones Java implementadas.

2 Diseño de la maqueta domótica didáctica (MDD)

En este trabajo proponemos un modelo de maqueta domótica aplicable al ámbito docente a partir de la experiencia que se ha ido adquiriendo en la construcción de maquetas. Dicha maqueta la denominaremos *maqueta domótica didáctica* (MDD), y consiste en un modelo a escala de una casa física real sobre la que se puede realizar diferentes tipos de actividades prácticas relacionadas con el área de la domótica. La maqueta proporciona una solución hardware-software que facilita al alumno la implementación, despliegue y depuración de aplicaciones que hacen uso de forma extensiva de los dispositivos domóticos presentes en una vivienda.

La MDD tiene características similares a una casa real para poder reproducir a una escala menor las condiciones del entorno real. Por tanto, al igual que en una instalación real es necesario distribuir una gran cantidad de componentes electrónicos por toda la casa para su correcto funcionamiento, así como la interconexión de dichos componentes electrónicos y el control de automatismos dentro del entorno de la casa. En nuestro caso, hemos incluido diferentes tipos de sensores, actuadores y dispositivos a escala para situarlos dentro de la MDD [1]. En la Tabla I se presenta algunos de los dispositivos que se incluyen en la MDD.

Desde un punto de vista topológico podemos distinguir tres niveles en la construcción de las maquetas, tal y como se muestra en la figura 1. En el nivel inferior se encuentra la capa de dispositivos que contiene todos los sensores, actuadores, reguladores y dispositivos que permiten controlar todos los automatismos de la casa. Se encuentran dispersos a lo largo de toda la maqueta, pero organizados en torno a un bus que facilita la identificación física de cada dispositivo. En este nivel, los alumnos pueden diseñar nuevas adaptaciones de dispositivos domóticos como, por ejemplo, nuevos controladores de persianas o un subsistema completo como un sistema de riego automático.

En el segundo nivel, se sitúa el controlador principal de la maqueta encargado del control hardware-software de todos los dispositivos instalados en la maqueta. Todos los elementos físicos se encuentran cableados y conectados al controlador principal a través de un bus. El controlador principal también dispone de los drivers y el código necesario para poder trabajar con los distintos dispositivos conectados. En este nivel los alumnos pueden desarrollar pequeños controladores o reguladores lógicos combinando la información procedente de uno o varios sensores con actuadores; por

ejemplo, se puede construir un sistema de seguridad pasiva activando todos los sensores de presencia de la maqueta, o un sistema de regulación día-noche para mantener el nivel de luz de una habitación.

Tabla 1. Dispositivos utilizados en las MDD construidas.

Funcionalidad	Tipo	Dispositivos
Iluminación	Actuador	Conjunto de bombillas de nivel de luz variable dispersas en cada habitación de la maqueta, así como en la parte exterior de la maqueta. Todas controladas independientemente.
Detector de iluminación	Sensor	Conjunto de sensores de iluminación en cada habitación, así como en el exterior para la detección día/noche.
Control de iluminación	Regulador	Se puede controlar la iluminación combinando los sensores de iluminación y la activación de las bombillas.
Detector ambientales	Sensor	Dispositivos que facilitan la detección de la temperatura interior/exterior, humedad, presión hidrostática, entre otros.
Control de temperatura	Regulador	Se combina un detector de temperatura con un climatizador para controlar la temperatura de una habitación. Se utiliza un sistema basado en calefactor o refrigerador, o una célula peltier y ventiladores cambiando la cara fría/caliente.
Detector de gas y humo	Sensor	Se incluyen varios sensores capaces de detectar la presencia de gases en la cocina para evitar posibles escapes de gas.
Control de persianas	Controlador	Las persianas pueden subirse o bajarse mediante un motor o servomotor, y uno o más sensores para controlar el estado de apertura de la persiana, o para saber si está completamente abierta o cerrada.
Alarma	Actuador	Se disponen un conjunto de zumbadores y sirenas que pueden activarse cuando sea necesario.
Control de presencia	Sensor	Se incluye sensores de presencia por infrarrojo distribuidos por las habitaciones para controlar si se pasa por una habitación u otra.
Apertura de Garaje	Controlador	Se incluye un dispositivo para controlar la apertura y cierre del garaje, y del estado de apertura del mismo.
Control energético	Controlador	Se incluye un dispositivo capaz de determinar la cantidad de corriente manejada. También se ha construido un sistema eficiente energéticamente mediante un sistema de batería para la acumulación de energía y un sistema de captación de energía solar.

Por último, en el tercer nivel se encuentran las aplicaciones interactivas que hacen uso de la maqueta, bien para interactuar con los dispositivos de la maqueta, o bien para recibir las notificaciones de los sucesos o eventos que están sucediendo en el interior o exterior de la maqueta. Dichas aplicaciones se ejecutan con un esquema cliente-servidor utilizando algún protocolo de comunicación, generalmente basados en TCP/IP como Ethernet y Wifi, aunque también es posible hacerlo sobre Bluetooth o Zigbee (o IEEE 802.15.4). Las aplicaciones interactivas desarrolladas pueden ser nativas basadas en las características de los clientes (Windows, Android, IOS, ...) o bien basadas en web utilizando navegadores web.

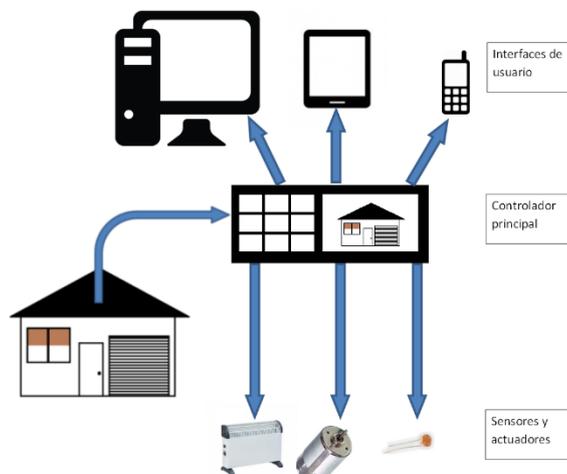


Figura 1. Representación topológica de la MDD

Para el diseño de la MDD se han seguido varios enfoques que dependen de la necesidad de construcción de toda la MDD, así como el conexionado de los distintos elementos. Concretamente, podemos distinguir tres alternativas principalmente:

- Diseño de la MDD sobre un demostrador o prototipo.
- Diseño de la MDD sobre una infraestructura parcialmente construida
- Diseño de la MDD completa.

En el primer caso el diseño de la MDD se basa en la utilización de un demostrador o prototipo que incluye toda la infraestructura necesaria para desplegar los dispositivos domóticos así como el cableado necesario para interconectarlos entre sí. En este caso, el diseño de la MDD consistiría en construir una adaptación del entorno domótico si es posible, establecer el conexionado de los dispositivos con un microcontrolador o pc donde ejecutar las aplicaciones, y el desarrollo del software necesario. El demostrador o prototipo puede consistir en un panel con los controladores y los dispositivos empotrados en el panel, o bien un sistema de construcción en base a piezas individuales tipo LEGO que facilite la construcción del entorno y su sistema de conexionado.

En el segundo enfoque el diseño de la MDD se realiza en base a una infraestructura parcialmente construida. En este caso, podríamos utilizar, por ejemplo, una casa de muñecas o una maqueta de una casa, y adaptarla a las necesidades de la MDD añadiendo los dispositivos necesarios, controladores, sensores, actuadores y el conexionado necesario [2]. En la Figura 2(b) se puede observar el montaje que hemos realizado sobre la infraestructura de una casa de muñecas denominado CasaDomo.

Con el tercer enfoque se puede construir una MDD orientada a un propósito específico, por ejemplo, para cubrir la seguridad perimetral de una vivienda, inundando la maqueta de sensores de detección de presencia y cámaras, o trabajar más el aspecto de automática mediante la inclusión de numerosos dispositivos con un conjunto de reguladores [3]. En este caso, es necesario planificar el proceso de construcción de la infraestructura de la casa, el conjunto de dispositivos que se va a considerar, el sistema de conexionado, y los elementos que conforman la topología.



Figura 2. Diversas maquetas diseñadas siguiendo el enfoque MDD.

Siguiendo los enfoques mencionados anteriormente hemos construido y desarrollado cuatro tipos de maquetas:

- **Maqueta HAL9013** (2004-2005). Siguiendo el enfoque 3 se construyeron varias habitaciones equipadas con sistemas de control de iluminación, control de presencia, detectores de gas y humo, control de persianas y control de temperatura, entre otros [3]. El diseño primaba la facilidad para interconectar los distintos dispositivos, y por este motivo se situó el microcontrolador y el sistema de cableado en la parte superior de la maqueta. Sin embargo, era un sistema pesado y difícil de transportar para las prácticas.
- **Maqueta CasaDomo** (2008-2010). En este caso se buscó un diseño más flexible, fácil de transportar, y sobre el que utilizando un mismo tipo de conector se pudieran conectar diferentes tipos de dispositivos. Se utilizó un enfoque 2 para su construcción utilizando una casa de muñecas con paredes más flexibles y se centralizó toda la electrónica en la parte lateral de la maqueta [2].
- **Maqueta DomoLab** (2010-2012). Basándose en CasaDomo, la maqueta DomoLab plantea la posibilidad de acceder y controlar de forma remota a los dispositivos de la maqueta desde cualquier ordenador conectado a Internet mediante el desarrollo de un laboratorio remoto. Para ello, se tuvo que idear un sistema de carga de aplicaciones Java a distancia para poder ejecutar las aplicaciones que implementaban los alumnos en sus ordenadores, así como la sincronización de un conjunto de cámaras de video para mejorar la experiencia de interacción del alumno sobre la maqueta [4,5].

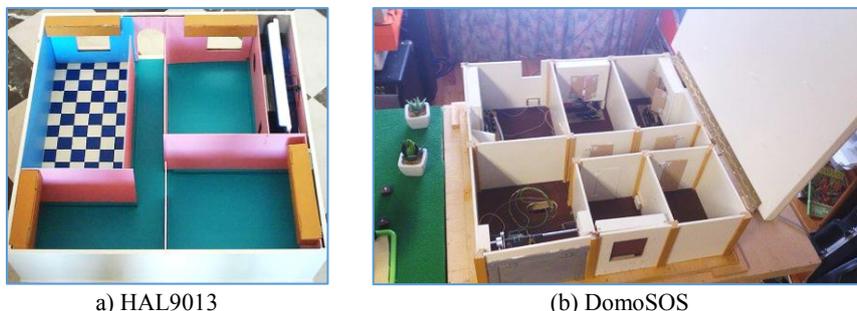


Figura 3. Estructuración de habitaciones en (a) HAL9013 y (b) DomoSOS.

- **Maqueta DomoSOS** (Sistema domótico sostenible) (2012-2013). En este caso, se propuso la construcción de una maqueta sostenible en dos sentidos. Por una parte, la maqueta incluye habitaciones reciclables que pueden cambiar en forma y extensión y, por otra parte, incluye distintos tipos de dispositivos domóticos intercambiables “plug and play” según las necesidades de la práctica. También se diseñó un sistema de gestión energética de todo el sistema, que además puede funcionar con energía solar [6].

3.1. Construcción de la maqueta y preparación de la electrónica.

La construcción de las distintas maquetas se ha realizado teniendo en cuenta los elementos del sistema de control domótico que queríamos automatizar en la casa, con la intención de obtener un modelo físico final ampliable y/o adaptable a nuevos usos futuros de modo que pueda ser utilizado como herramienta didáctica para fines docentes. Para ello se ha elaborado una maqueta fácilmente desmontable para facilitar las labores de mantenimiento y reparación, ocultando en la mayor medida de lo posible el cableado que pasa por las habitaciones y proporcionando la suficiente movilidad para que pueda ser desplazada sin excesivos problemas. En la Figura 3 se pueden ver algunas imágenes de los prototipos de maqueta HAL9013 y DomoSOS. En ambos casos se pueden extraer las paredes y adaptarlas según las necesidades, aunque en el caso de DomoSOS incluso las paredes de la parte exterior de la casa son reutilizables.

Del mismo modo, la electrónica y el cableado se han diseñado con cuidado realizando una planificación de todo el sistema eléctrico y electrónico de la maqueta (ver Figura 4), probando cada dispositivo electrónico (chip, sensor, actuador) antes de ser instalado en la maqueta. Para ello, se ha realizado una selección de los dispositivos electrónicos teniendo en cuenta el tamaño de los mismos, el consumo, el tipo de alimentación, o el tipo de control (si es analógico o digital, etc.), y la construcción de módulos genéricos de adaptación de niveles de corriente e intensidad o circuitos que aislen de posibles interferencias generadas por actuadores. También se ha tenido en cuenta un esquema de colores para distinguir los cables, tipo de conectores específicos, serigrafía de circuitos desarrollados, leds para indicar el estado de las placas adaptadoras, y circuitos para separar la alimentación de los dispositivos de la lógica.

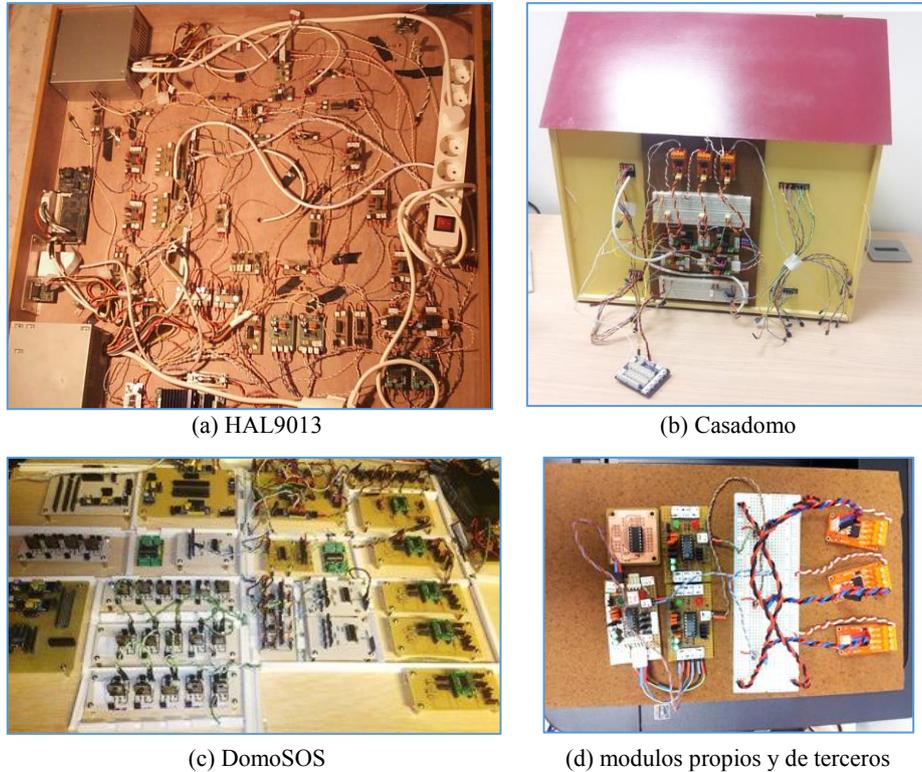


Figura 4. El cableado se ha organizado de forma diferente en cada maqueta: (a) arriba en HAL9013, (b) lateral en CasaDomo/DomoLab y (c) abajo en DomoSOS. En (d) se muestra algunos módulos diseñados por el equipo y de tercero.

3.2. Selección del controlador principal.

El controlador principal de la MDD se encarga de organizar todos los dispositivos, sensores y actuadores disponibles en la MDD para que estén accesibles tanto para las aplicaciones clientes que tienen que interactuar con los dispositivos domóticos como para cargar las aplicaciones que desarrollen los alumnos. Se implementa sobre un microcontrolador o un sistema empotrado conectado a todos los dispositivos domóticos de la MDD, que se encarga de generar las señales eléctricas precisas para leer el estado de los sensores, lo que nos permite monitorizar el entorno físico del hogar, y por otra parte genera las señales de control adecuadas para accionar los actuadores que interactúan sobre el entorno físico del hogar. Dado que los sistemas empotrados cuentan con un número escaso o inexistente de entradas/salidas analógicas y digitales, la incorporación del bus basado en I²C (ver Figura 5) facilita la extensión de las capacidades hardware del sistema agregando, por ejemplo, expansores digitales de E/S o conversores ADC.

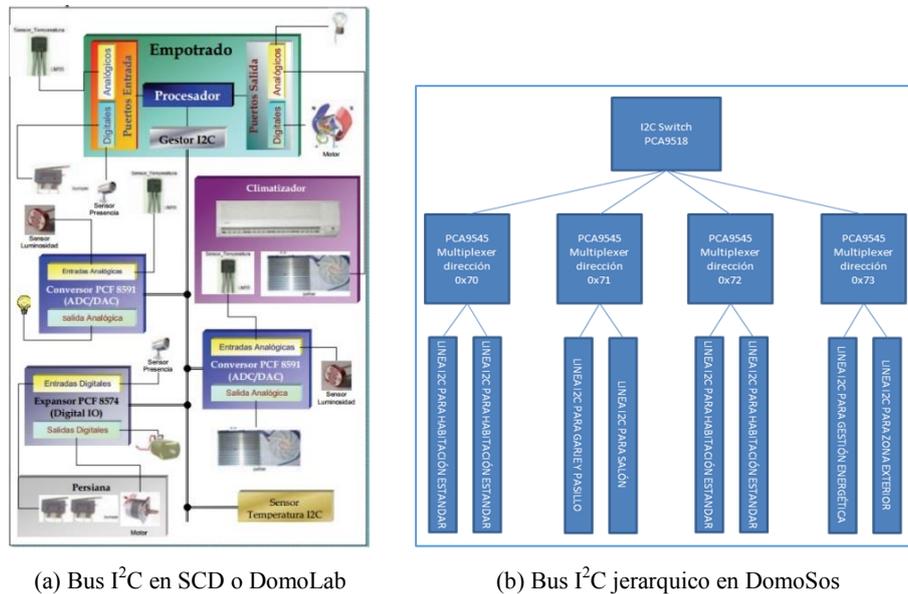


Figura 5. Esquema de conexionado

El bus de comunicaciones I²C [7] permite controlar los dispositivos y expansores de E/S con un paradigma maestro-esclavo utilizando distintas velocidades (100 KHz a 5 MHz). Sin embargo, la velocidad está limitada por el dispositivo más lento y la estabilidad de la señal depende de que la capacitancia total del bus no supere los 400 pF. En la maqueta DomoSOS se ha ido un paso más allá, y se ha diseñado un bus del sistema plug-and-play jerárquico basado en el bus de comunicaciones I²C que facilita la incorporación de nuevos circuitos de adaptación, periféricos de E/S y expansores de E/S sin necesidad de resetear el sistema [8].

Para la selección del microcontrolador o empotrado se han valorado varios aspectos como la capacidad de conectividad, la capacidad de acceder a los dispositivos domóticos esclavos del bus I²C, la presencia de sistema operativo, y la posibilidad de ejecutar aplicaciones en el lenguaje de programación Java. En la tabla 2 se indica las características de los empotrados utilizados para cada maqueta.

Tabla 2. Lista de Empotrados utilizados en la MDD

Empotrado	SmartControl SC210	IM3910	Arduino Uno	Raspberry Pi B
Fabricante	Snijder	Imsys	Arduino	Raspberry
Características	Cirrus Logic EP7312, 64 MB RAM, 16 MB Flash	IM3910 (167 Mhz), 32 Mb RAM, 8 Mb Flash	ATmega328P (16 Mhz), 2 Kb SRAM, 32 KB Flash	ARM1176JZF-S (700 MHz), 512 MB RAM, SD Flash
Conectividad	50 GPIO, 3 ADC, I ² C, LCD 5,7" color	8 I/O digital, I ² C, SPI	13 E/S digital, I ² C, 4 Ent. analó, Shield Ethernet	8 GPIO, SPI, I ² C, UART
Maqueta	SCD	CasaDomo	DomoLab	DomoSOS
Sist. Operativo	S.O. Tiempo Real	S.O. Tiempo Real	-	Linux
Programación	Java	Java	Processing	Java

4. Programación con la MDD

La programación con la MDD se puede realizar en varios niveles de abstracción basados en el lenguaje de programación Java, según las necesidades de la práctica o del sistema que se quiera probar sobre la maqueta. La MDD contiene la infraestructura básica que facilita la ejecución de aplicaciones Java sobre un entorno de ejecución JVM basado en JavaSE o JavaME, según el tipo de empotrado utilizado. Esto es, contiene tanto el sistema de carga de aplicaciones con el que se pueden desplegar aplicaciones sobre el empotrado como un agente de depuración que facilita la depuración remota de las aplicaciones sobre dicho empotrado. De este modo, la MDD proporciona un entorno completo de programación, compilación, despliegue y depuración de aplicaciones.

Si bien es factible desarrollar aplicaciones Java individuales que trabajen sobre un conjunto de dispositivos domóticos teniendo en cuenta las características específicas del entorno JVM instalado en el empotrado, se han desarrollado distintos tipos de tecnologías encuadradas en proyectos de innovación docente y en trabajos fin de carrera o de máster que han permitido extraer todo el potencial que pueden tener estas herramientas didácticas. Una lista de trabajos se muestra en la Tabla 3.

Tabla 3. Trabajos realizados sobre la MDD

Trabajos Fin de Carrera (TFC) / Trabajo Fin de Máster (TFM)
Viúdez J., Montes J.M. Sistema de Control Domótico HAL9013. TFC (2005)
Francisco J. González, Pablo Polo y Francisco J. Viceira. Desarrollo de un sistema domótico "inteligente" utilizando Jini. TFC (2006)
M ^a Dolores Serrano y Sandra Rodríguez. Control domótico del hogar utilizando JXTA. TFC (2007).
Jaime Viúdez Aivar. Desarrollo de una plataforma para el diseño de sistemas de control con requerimientos en tiempo real en entornos empotrados basado en Java. TFM (2007)
Jesús Luis Muros Cobos. Framework con soporte de video para laboratorios remotos y virtual: aplicación al laboratorio remoto Domolab. TFM (2011)
Rafael Hidalgo. DomoWeb: Automatización domótica basado en servicios web. TFC (2012)
Francisco J. Novo y Juan M. Leal. Sistema de Teleasistencia DOMOSMART. TFC (2012)
Jonatan Sánchez Agulló. Diseño y Control de una Centralita Domótica accesible desde Dispositivo Android: Aplicación a la Maqueta Domótica. TFM (2012)
Manuel José Baena Toquero. DomoSOS: Sistema domótico sostenible basado en Raspberry Pi y Android. TFC (2013)

Como no es posible incluir todas las tecnologías desarrolladas, se comentan algunos ejemplos a continuación:

- **JavaES** (Java for Embedded Systems) [9] es un framework basado en Java que ofrece una plataforma adaptable, flexible y robusta para el desarrollo de aplicaciones sobre sistemas empotrados de 8, 16 y 32 bits. Facilita el diseño de aplicaciones de control (incluso con características de tiempo real) sobre dispositivos que se conecten a los puertos de E/S del entorno empotrado, o a través de buses digitales I²C, SPI y 1-Wire. Para ello, proporciona mecanismos y facilidades de alto nivel para el acceso y manejo del hardware subyacente independientemente de la arquitectura hardware del sistema empotrado.

```

package example.buses;
import JavaES.digitalBus.*;
import JavaES.sys.*;

public class pruebaLM92{
    public static void main(String args [ ]) {
        JSYS.initialize(); // To Initiate any of supported embedded system.
        DigitalBus bus = JSYS.getDigitalBus(0,DigitalBus.I2C,"Bus_I2C");
        TemperatureSlave sensor = new LM92_I2CSlave(0,"LM92_sensor", bus);
        while(true){
            try{
                System.out.println("-Temperatura: "+sensor.getTemperature());
                Thread.sleep(1000);
            }catch(Exception e){ e.printStackTrace(); }
        }
    }
}

```

Figura 6. Ejemplo de un programa en Java que accede a un sensor de Temperatura a través de I²C utilizando la tecnología JavaES sobre la maqueta Domolab.

- **DOHA** (Dynamic Open Home Automation) es una plataforma de servicios con una arquitectura orientada a servicios, diseñada para proporcionar soporte a sistemas distribuidos, ubicuos y dinámicos, así como para facilitar el acceso, control y gestión de espacios de interacción ubicuos como el hogar desde cualquier dispositivo de cómputo como ordenadores portátiles, teléfonos móviles o plataformas empotradas. Proporciona un modelo de programación basado en servicios que facilita la integración y colaboración de servicios que pueden ejecutarse en el mismo dispositivo o en diferentes dispositivos, ocultando la complejidad en las comunicaciones o en el acceso al hardware.

5. Conclusiones

La experiencia recogida en la construcción de maquetas nos ha permitido crear diferentes tipos de maquetas domóticas que nos permite ofrecer diferentes tipos de opciones a la hora de plantear actividades prácticas con fines docentes. Como consecuencia de los proyectos de innovación docente se han desarrollado guiones de prácticas para su aplicación en asignaturas de Ingeniería Informática e Ingeniería de Telecomunicación sobre diferentes niveles de abstracción. Así, por ejemplo, se han elaborado guiones de prácticas que enseñan a crear un driver de tiempo real de un dispositivo domótico, y otros guiones que enseñan a construir aplicaciones distribuidas basadas en la composición de servicios web como el diseño de un sistema de seguridad perimetral o de teleasistencia para personas mayores.

Sin embargo, la aplicación de la maqueta domótica al Aula ha tenido un éxito relativo por varios motivos. La utilización de la maqueta plantea algunas dificultades relacionadas con el transporte y puesta en marcha del sistema en el aula de prácticas, dificultades con el uso compartido de la maqueta cuando varios estudiantes tienen que trabajar con el mismo conjunto de dispositivos domótica, y una limitación de tiempo en el uso de la herramienta en ocasiones impuesta por el propio tipo de prácticas que

hacen complicada la finalización y prueba del sistema final. Aunque una solución fue la implantación del laboratorio remoto DomoLab, esto planteó otros problemas de fiabilidad cuando el estudiante realizaba una carga de aplicación no válida o incorrecta.

Por otra parte, debemos tener en cuenta que el abaratamiento del hardware y la aparición de los sistemas de open hardware como Arduino o Raspberry Pi han permitido plantear diseños parciales de una MDD basados en enfoques de primer tipo, que proporcionan acceso a un mercado de dispositivos (entre ellos dispositivos domóticos) que puede ser programados y controlados a través de microcontroladores o sistemas empujados con recursos limitados. No sólo se incluyen facilidades para la programación, sino que se proporcionan conjuntos de dispositivos sobre la base de kit de desarrollo o “starter kit”. Esto permite plantear la realización de prácticas directamente sobre el microcontrolador de una forma mucho más directa en lugar de plantearlo sobre toda la casa. Sin embargo, en este caso la aplicación a prácticas de mayor nivel de abstracción requiere el desarrollo de soluciones ad-hoc específicas.

En líneas generales la experiencia ha sido positiva, especialmente para la puesta en marcha y prueba de nuevas tecnologías como las que se han desarrollado en algunos de los trabajos fin de carrera o trabajos fin de máster.

Agradecimientos. La construcción de las maquetas no habría sido posible sin la colaboración de muchas personas que han contribuido con su aportación como Jaime Viudez Aivar, José Miguel Gutiérrez Guerrero, Jesús Muros Cobos, Manuel Baena Toquero, Francisco Novo Sánchez, Antonio Gutiérrez López, Sandra Sheila Rodríguez Valenzuela, Jaime Viudez Aivar, José Antonio Molina López, Jonathan Sánchez Agulló, entre otros.

Referencias

1. Viúdez J., Holgado Juan A. Diseño y construcción de una maqueta domótica controlable a través de microcontroladores Java. V Jornadas de Enseñanza a través de Internet/Web de la Ingeniería de Sistemas y Automática (EIWISA'07), pp. 47-52. Thomson. ISBN: 978-84-9732-603-2 (2007)
2. Holgado, Juan A. Proyecto de Innovación Docente: Maqueta Domótica para la Programación de Sistemas (número 08-181). Universidad de Granada (2008)
3. Holgado-Terriza, Juan A., Viúdez-Aivar Jaime, Capel-Tuñon Manuel I., Montes J.M. Diseño de un Sistema de Control Domótico basado en Java. Actas de las XXVII Jornadas de Automática (CEA), páginas 1401-1408. ISBN: 84-689-9417-0. Almería. (2006)
4. Holgado, Juan A. Proyecto de Innovación Docente: Maqueta Domótica para la Programación de Sistemas: Laboratorio Remoto DomoLAB (número 10-192). Universidad de Granada (2010)
5. Muros-Cobos, Jesus and Holgado-Terriza Juan A. A Componentizable Server-Side Framework for Building Remote and Virtual Laboratories.. International Journal of Online Engineering (iJOE), Vol (8), Num 4, pag. 43-51 (2012)
6. Manuel José Baena Toquero. DomoSOS: Sistema domótico sostenible basado en Raspberry Pi y Android. Trabajo Fin de Carrera. Universidad de Granada (2013)
7. Bus digital I²C (Inter-Integrated Circuit). Circuitos Inter-integrados. <http://www.nxp.com>

8. Gutierrez-Guerrero, José M.; Holgado-Terriza, Juan A.; Muros-Cobos, Jesús; Pomboza-Junes, Gonzalo. Redes de Sensores I2C Inteligentes. Actas de las XVI Jornadas de Concurrencia y Sistemas Distribuidos, pag. 77-91. ISBN: 978-84-16478-90-3 (2016)
9. Holgado-Terriza Juan A., Viúdez-Aivar Jaime. JavaES, a flexible Java Framework for Embedded Systems. Distributed, Embedded and Real-time Java Systems, pp. 323-355. Springer (2012)