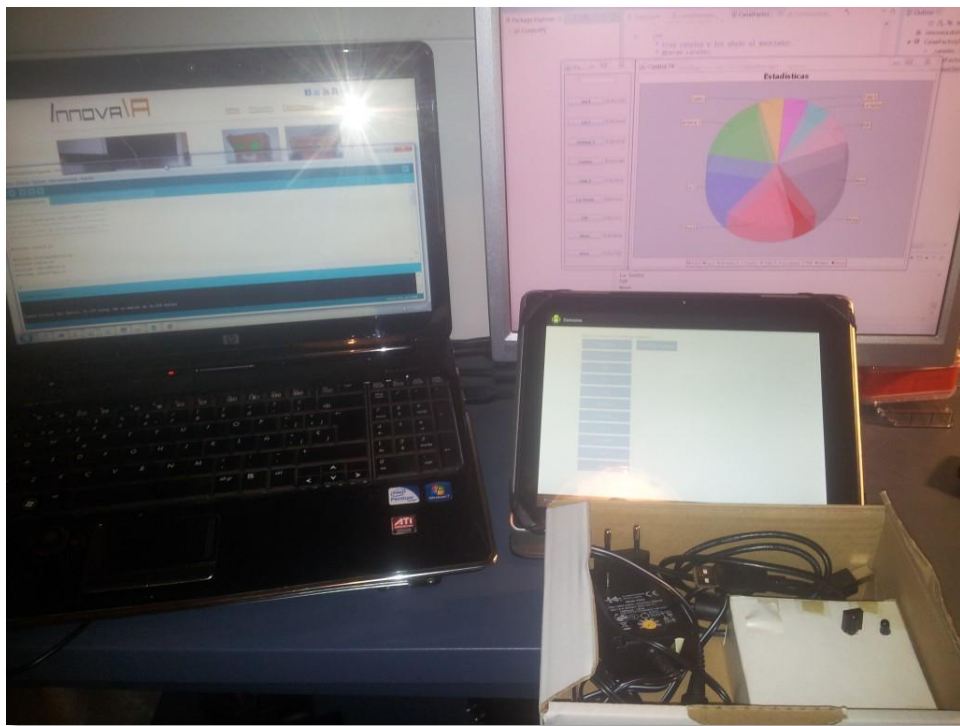

DOMOINO

TAREA 7 - PROYECTO FINAL



Diego Reyes Rodríguez
16/12/2013

CONTENIDO

1. Análisis..... 2

 1) Enunciado 2

 2) Descripción del proyecto..... 2

 3) Componentes 3

 4) Presupuesto 3

2. Desarrollo 4

 1) Diseño de circuitería..... 4

 2) Montaje electrónico 5

 3) Arquitectura 7

 4) Programación 8

 5) Protocolos de comunicación 9

3. Manual de Usuario 10

 1) Instalación 10

 2) Instrucciones de uso..... 10

4. Conclusiones..... 11

1. ANÁLISIS

En este apartado se describirá qué se quiere hacer, por qué y qué necesitamos para hacerlo.

1) ENUNCIADO

La presente tarea consiste en la creación de una aplicación completa con Arduino.

Se trata de crear tanto software como hardware y la documentación para reproducirlo.

Se valorará la originalidad.

Es necesario adjuntar esquema, código y documentación con fotografías en el caso de haberlo realizado.

2) DESCRIPCIÓN DEL PROYECTO

Proyecto

Domoino es un aparato que comunica con otros aparatos que estén en una misma habitación. Este aparato (a partir de ahora, nodo maestro) es capaz de recibir información del resto de aparatos (a partir de ahora, nodos esclavos) para enviarles órdenes y recibir datos.

Este nodo consta de Bluetooth, que permitirá su control mediante una App en Android, un receptor IR para recibir información de otros nodos y un emisor IR, para enviar órdenes al resto de nodos.

El proyecto completo consta de una aplicación Java (desarrollada en la Tarea 5), una aplicación Android, hardware Arduino, con un software que le permite enviar y recibir información de diferentes fuentes de comunicación, y que sirve de unificación del nodo maestro comenzado en las tareas 3 y 4, del monitor IR desarrollado en la tarea 5 y del ejercicio Eco IR puesto en los foros.

Objetivos

Al final de la realización de este proyecto se pretende haber alcanzado los siguientes objetivos:

- **Demostrar conocimientos:** Se pretende demostrar conocimientos avanzados en la plataforma Arduino.
- **Desarrollar un sistema genérico de comunicaciones:** En esta práctica final, se busca centrarse en sistemas de comunicación, desarrollando un sistema base que permita añadir diferentes medios de comunicación.
- **Simplicidad :** Se quiere demostrar que, con pocos ingredientes, se puede realizar un aparato sencillo, dejando la complejidad en las múltiples utilidades de domotización que puede tener.
- **Crear un nodo maestro:** Que servirá para sincronizar todos los nodos esclavos.
- **Crear aplicaciones externas:** Capaces de comunicarse con este nodo central, para enviarle y recibir información de él.

- **Escalabilidad:** Tanto a nivel hardware como a nivel software, se pretende que el sistema pueda crecer fácilmente, tanto incorporándole nuevos componentes como programas.
- **Modularidad:** Se busca desarrollar pequeños módulos, tanto a nivel hardware como software, cada uno con una funcionalidad completa y definida, con la menor dependencia posible entre ellos.
- **Abstracción entre hardware y software:** Para minimizar la dependencia entre hardware y software, esto es, que los programas ejecutados puedan funcionar sin conocer detalles específicos del hardware, como los pines donde están conectados los componentes.
- **Divertirse y aprender:** Por supuesto, el objetivo final tanto de este proyecto como de todo el curso, es aprender Arduino y las posibilidades que ofrece esta plataforma.

3) COMPONENTES

Electrónica

Los elementos necesarios para la fabricación del climatizador son:

- **Módulo Bluetooth:** Un módulo de comunicación Bluetooth, para enviar órdenes del móvil al nodo maestro.
- **Emisor IR:** Un led de emisión infrarroja, que enviará datos a los nodos esclavos.
- **Receptor IR:** Un sensor infrarrojo, que recibirá datos de los nodos esclavos.
- **Placa Arduino:** Para orquestar los elementos anteriores, se empleará una placa Arduino UNO, sencilla, barata y con pines más que suficientes para este proyecto.

4) PRESUPUESTO

El cálculo de costes (con IVA) se ha realizado consultando las webs de [BricoGeek](#) y [Electrónica Embajadores](#) y [Conectrol](#), donde han sido comprados los componentes.

Electrónica	Tipo	Unidades	Precio unidad	Total
Modem Bluetooth Bluesmirk Gold	Comunicaciones	1	64,31 €	64,31 €
Emisor IR	Comunicaciones	1	0,18€	0,18€
Receptor IR	Entrada	1	1,16€	1,16€
Placa Arduino UNO	Placa Arduino	1	26,62 €	26,62 €
Resistencia 220Ω	Resistencia	2	0,37 €	0,74€
Cables, clemas, tornillos, caja.	Varios	-	-	-
TOTAL				93,01€

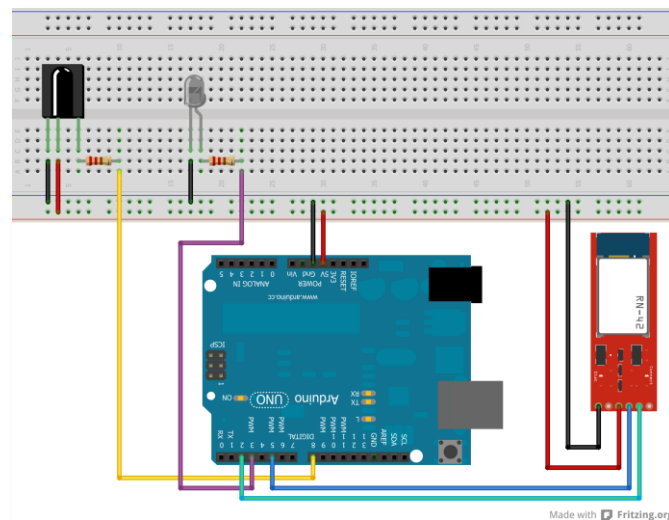
2. DESARROLLO

Explicación detallada de la construcción y desarrollo del proyecto.

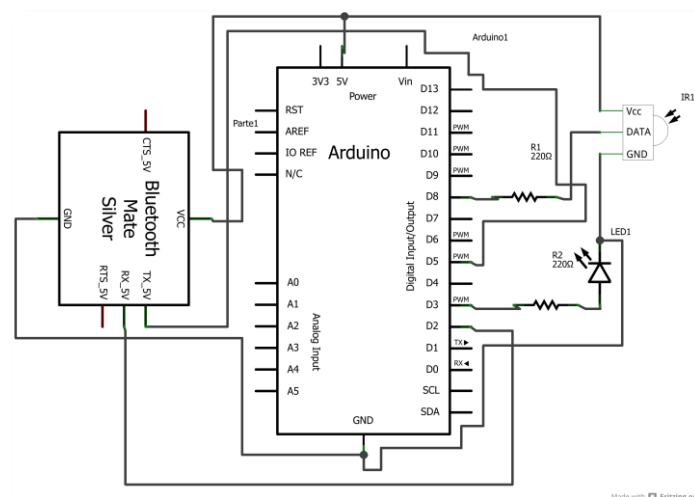
1) DISEÑO DE CIRCUITERÍA

Antes de empezar a montar los componentes electrónicos, debemos diseñar la circuitería, para evitar posteriores errores de ejecución. Los esquemas en alta resolución y el archivo fritzing se encuentran en la carpeta **/Esquemas/**.

Protoboard



Esquema



2) MONTAJE ELECTRÓNICO

Siguiendo el esquema realizado, montaremos todos los componentes en la placa. Lo haremos por partes. Las fotos en mayor resolución se en la carpeta **/Fotos/**.

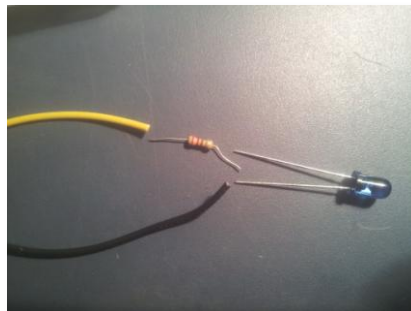
Módem Bluetooth

Conectamos el módem Bluetooth al pin +5v, a GND y a los pines 2 y 5, que harán de Tx y Rx, gracias a la librería Software Serial.



Emisor IR

Conectamos el emisor IR, con su resistencia de 220Ω, a la protoboard al pin 3.



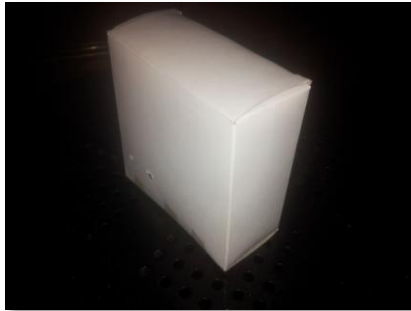
Receptor IR

Conectamos el receptor IR, con su resistencia de 220Ω, al pin 8.

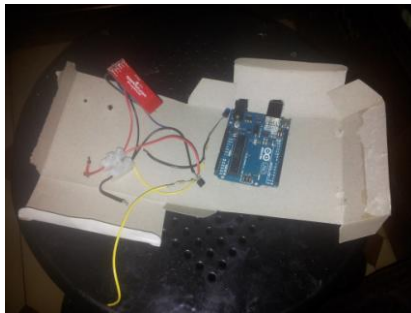


Carcasa

Buscamos una caja de cartón pequeña (en este proyecto hemos usado la de una Raspberry Pi).



La abrimos y colocamos y atornillamos dentro los componentes, utilizando clemas para las conexiones.



Hacemos una pequeña abertura para poder ver las luces de dentro, además de las aberturas para el puerto USB, la toma de corriente y el receptor y emisor IR.

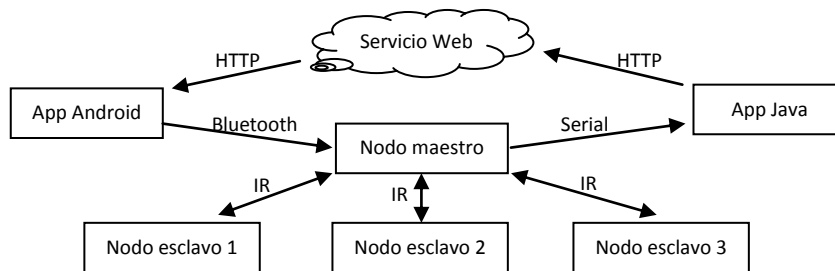


Y éste es el resultado final:



3) ARQUITECTURA

El sistema Domoino consta de los siguientes elementos:



App Java

Sirve para crear, editar y monitorizar la audiencia de los canales de los diferentes nodos a través del puerto serial. El XML que genera se sube a la nube por FTP.

App Android

Esta aplicación sirve para descargar desde un servicio web la lista de nodos con sus canales ya creados con la aplicación Java, y listarlos en pantalla para que el usuario pueda, por Bluetooth, comunicarle al nodo maestro el código y protocolo a emitir.

Servicio web

Este servicio ofrece a la aplicación de Android el XML generado por la aplicación Java.

Arduino

El nodo maestro recibe las órdenes de la App Android y las emite por IR a los nodos esclavos, que a su vez, pueden comunicar al nodo maestro, también por IR, su estado.

También monitoriza la información recibida por IR y se la transmite a la APP Java por serial, para que pueda generar el XML de nodos.

4) PROGRAMACIÓN

Se utiliza una programación sencilla y auto-explicativa gracias a sus comentarios, utilizando técnicas de programación avanzada, ya vistos en la tarea 2.

Java

Sobre la aplicación realizada en la Tarea 5 se han hecho dos ligeras correcciones; el cambio de nombre de **frecuencia** a **código** en el formato del fichero XML y mostrar en las estadísticas sólo aquellos canales con al menos una emisión. Se puede encontrar la última versión en la carpeta **/Código/Java/**.

Android

Esta aplicación, ubicada en **/Código/Android/**, reutiliza parte del código ya hecho para la aplicación de Java.

Servicio web

Actualmente un fichero XML, que se encuentra en:

<http://innovaia.es/arduino/Nodos.xml>

Arduino

Se adjunta el código dentro de la carpeta **Código**. Dentro de esta carpeta podemos encontrar:

- **libraries:** contiene las librerías de comunicación desarrolladas, las cuales heredan de **ComunicacionBase**, que ofrece métodos de envío y recepción de datos. Éstas son **ComunicacionIR**, **ComunicacionBluetooth** y **ComunicacionSerial**.
Se utiliza **ComunicacionBluetooth** para recibir órdenes de Android.
ComunicacionIR, se utilizar para enviar por el emisor IR el código adecuado, según la orden recibida por Bluetooth.
ComunicacionSerial enviará por el puerto serie el código a emitir, para alimentar a la aplicación Java.
- **NodoMaestro:** contiene el sketch principal, **NodoMaestro.ino**, con el controlador, que orquesta las comunicaciones, y el fichero de configuración **Configuracion.h**.
- **Calefactor:** contiene una simplificación del proyecto Climatizador, con la característica añadida de un receptor IR para recibir órdenes (se especifica en el apartado de "Protocolos de comunicación").

La descripción detallada del código está en los comentarios del programa.

5) PROTOCOLOS DE COMUNICACIÓN

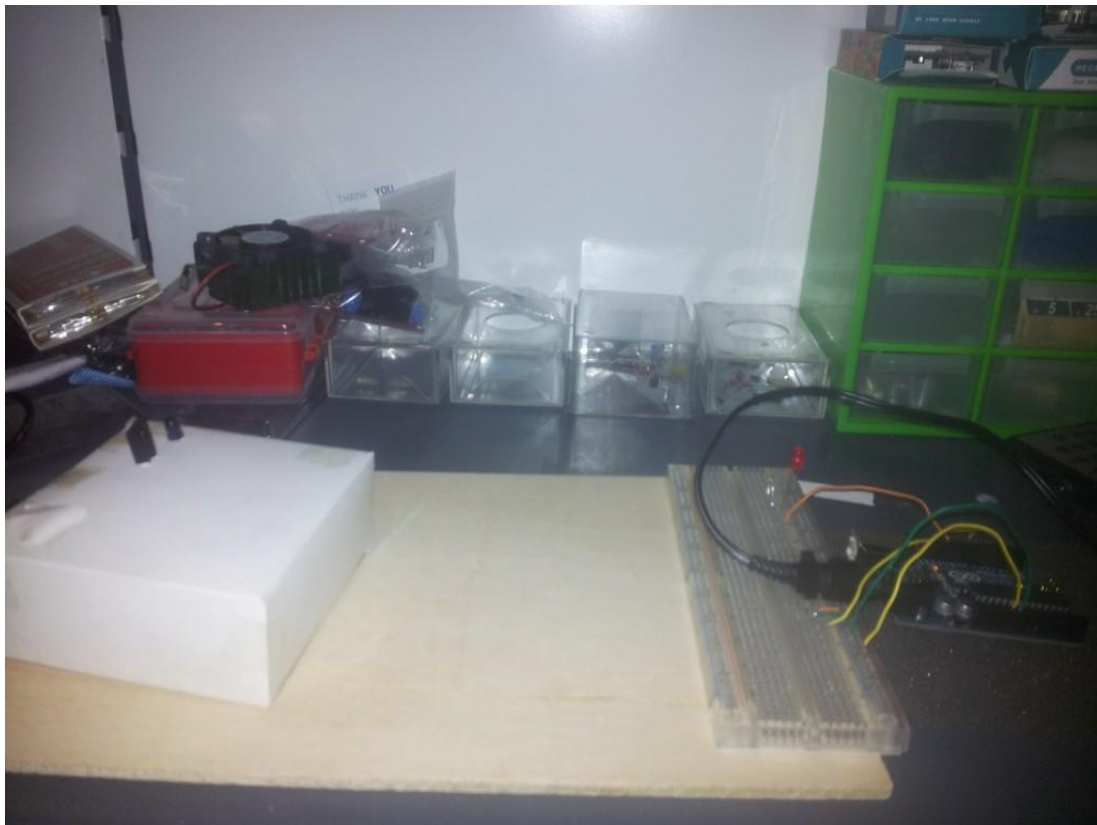
La aplicación Arduino utiliza los protocolos de comunicación definidos en la Tarea 3. Se añade a este protocolo el siguiente comando:

ADA0XY.

Según el protocolo ya definido, ADA es el indicador de que es un nodo Domoino, 0 indica que el emisor es el nodo maestro, mientras que X sería el tipo de mensaje e Y los parámetros.

En el caso de que el nodo maestro sea el emisor, el tipo de mensaje (X) será la ID del nodo esclavo a quien va destinado el mensaje, mientras que Y será el tipo de mensaje. La Y con valor 0 indica que el comando a ejecutar es encender o apagar el nodo esclavo.

Se añade un nuevo nodo esclavo, el nodo **Calefactor**, con ID3, que es una simplificación del Climatizador creado en las tareas 1 y 2.



Este protocolo no se utilizará para la comunicación con otros nodos no correspondientes a la red de Domoino, como una televisión, que maneja sus propios códigos y protocolos.

3. MANUAL DE USUARIO

1) INSTALACIÓN

1. Conectar los componentes siguiendo las instrucciones de montaje, definidas en el apartado "Montaje electrónico".
2. Añadir las librerías ubicadas en **Código/libraries** a las librerías de Arduino.
3. Añadir el programa ubicado en **Código/** a la carpeta de trabajo de Arduino.
4. Conectar la placa a por usb con el ordenador para su comunicación por el puerto serial..
5. Compilar y cargar el sketchbook **NodoMaestro.ino** en la placa **Arduino UNO**.
6. Ejecutar la aplicación Java (ver instrucciones en Tarea 5).
7. Instalar la apk Android en un móvil o tableta con el SO Android.

2) INSTRUCCIONES DE USO

Gestión de nodos

Para configurar los diferentes nodos, conectamos el nodo maestro por el puerto serie al PC, con la aplicación Java en ejecución. Seguiremos las instrucciones descritas en la tarea 5.

Subir el archivo de nodos a la nube

En el sistema actual, el fichero XML generado por la aplicación Java, se subirá manualmente por FTP al servidor, y se descargará de <http://innovaia.es/arduino/Nodos.xml>.

Visualización de nodos

En la tarea 3 se explica cómo se pueden ver la información que los nodos esclavos transmiten por IR al nodo maestro.

Uso de la aplicación móvil

La aplicación para Android requiere permisos de Bluetooth y acceso a Internet. Esta aplicación pedirá activar Bluetooth en el móvil. Una vez hecho, descargará el fichero XML de la nube y mostrará los mandos configurados al usuario, con un botón por cada canal.

Para enviar la orden, simplemente se pulsa sobre el canal que se desee. Este canal será enviado al nodo maestro, el cual replicará por IR al nodo esclavo correspondiente.

4. CONCLUSIONES

Si repasamos los objetivos, podemos ver que:

- **Demostrar conocimientos:** Utilizando de base las tareas realizadas a lo largo del curso, este proyecto finaliza los trabajos realizados, aunque dejando muchos campos abiertos para futuras ampliaciones. El desarrollo de todas estas prácticas me han permitido adquirir conocimientos avanzados de Arduino, así que considero que, pese a que siempre se puede aprender más, el objetivo está cumplido.
- **Desarrollar un sistema genérico de comunicaciones:** Se ha desarrollado una librería de ComunicacionBase, de la que actualmente heredan varios canales de comunicación, y es fácil añadir nuevos canales, como WiFi, e intercambiar estos canales en nuestra aplicación según nos convenga, con lo cual también se cumple el objetivo.
- **Simplicidad :** Sólo tres ingredientes unidos a la placa: bluetooth y emisor y receptor IR. La sinergia del nodo maestro con el servicio web, la app Android y la app Java hace que elementos muy simples consiguen crear las bases para la domotización de una habitación entera.
- **Crear un nodo maestro:** El nodo maestro desarrollado se comunica tanto con nodos esclavos de la red Domoino como con otros nodos IR fuera de la red, como televisores, y es completamente funcional.
- **Crear aplicaciones externas:** Se han desarrollado dos aplicaciones, una en Java y otra en Android, capaces e interactuar con el nodo maestro. La imaginación es el único límite para desarrollar otras Apps, que permitan, por ejemplo, programar nuestro nodo maestro para que encienda la calefacción a una hora o nos apague la televisión cuando vayamos a dormir.
- **Escalabilidad:** El sistema Domoino es capaz de crecer añadiendo nuevos nodos esclavos. Como ya hemos visto, cambiar de sistema de comunicación es muy sencillo. Un cambio del emisor y receptor IR por otro sistema, sólo implicaría el desarrollo de la librería correspondiente, que heredase de ComunicacionBase, y el atributo de comunicación IR en la aplicación por el que se haya desarrollado.
- **Modularidad:** La creación de librerías de comunicación nos da la modularidad buscada.
- **Abstracción entre hardware y software:** El archivo de configuración del nodo maestro contiene toda la información de los pines, de modo que es el único que debemos modificar si, por ejemplo, cambiamos de placa. Nótese que este objetivo no ha sido cumplido para el emisor IR, ya que la librería empleada (IRLib) nos obliga a colocar este led en un pin concreto.
- **Divertirse y aprender:** Montar un nodo maestro domótico, enchufarlo, descargar la aplicación de móvil realizada y ver, al pulsar un botón en el móvil cambia el canal de televisión utilizando algo que tú mismo has desarrollado no se puede describir con palabras. Además de divertirme entre resistencias, leds y demás, he aprendido algo más de electrónica (mi asignatura pendiente), detalles nuevos de programación (C/C++ complica mucho las cosas y el IDE de Arduino no es que las facilite) y temas de comunicación, gracias a pegarme y chocarme con el funcionamiento de los diferentes sistemas empleados.

Mirando más al futuro, este sistema es sólo una base de lo que se puede alcanzar. Posibles mejoras que me gustaría realizar, y que realizaré son:

- **Aplicación Java:** Más completa, con soporte multinodo, que permita seleccionar el puerto al que está conectado la placa, menús más atractivos.
- **Aplicación Android:** Con menús que permitan enviar otro tipo de órdenes, como programar nodos.
- **Nodo maestro:** Con nuevas funcionalidades, como la programación de nodos, una placa WiFi para poder sincronizarse directamente con los nodos, almacenamiento SD donde guardar la información de los nodos o la programación de los mismos. Probar nuevos sistemas de comunicación, con más alcance que IR, como NRF24L01. También me gustaría reducir el tamaño y el precio, utilizando, por ejemplo, ATtyni y otro módulo Bluetooth más sencillo y más barato.
- **Nodos enchufe:** Como nodo esclavo, se podría desarrollar un nodo enchufe inteligente, como el analizado en la tarea 6, un nodo simple, que, embebido en esta red, cobraría gran importancia, ya que permitiría activar o desactivar aparatos eléctricos en función del estado que otros nodos enviaran al nodo maestro.
- **En la nube:** Una web completa, donde los usuarios pudieran gestionar sus nodos desde allí, e incluso pudiera aprender de cada usuario para realizar acciones usando inteligencia artificial, como por ejemplo, aprender que un usuario siempre enciende la calefacción a la misma hora y encenderla sin que él intervenga (preguntando primero, por cuestiones de seguridad).
- **Seguridad:** Con registro de usuario y sistemas de autenticación para las apps.

Por último, en el siguiente enlace se puede encontrar un vídeo donde se puede ver el funcionamiento de Domoino:

<http://youtu.be/kxpjNpY6duQ>