



CREACIÓN DE UN SISTEMA DOMÓTICO INALÁMBRICO LIBRE



12 DE ENERO DE 2017
AUTOR: GUSTAVO EZEQUIEL MARTÍNEZ PINTOS
TUTORA: GUILLER MELLADO

INSTITUT FORT PIUS

Agradecimientos

En primer lugar, quiero agradecer a mi familia y sobre todo a mi madre por soportar cada una de mis ideas y cada una de mis frustraciones con la comprensión y el afecto necesario. También a Eric, un gran compañero de clase y amigo, por su gran predisposición a ayudarme en todo momento. Por último, también agradecer a todas aquellas personas que, de alguna manera u otra, inconscientemente o indirectamente, han aportado alguna idea con un simple comentario o me han dado su opinión sobre el proyecto.



“Gratitude opens the door to the power, the
creativity and the wisdom of the universe.”

- Deepak Chopra



Contenido

Introducción	3
Funcionamiento general del sistema.....	5
El servidor.....	5
Interfaz de usuario gráfica.....	6
Módulo emisor de órdenes	6
Módulos receptores.....	7
Comunicación.....	8
Software	9
Introducción a la programación web.....	9
Introducción a Node.js.....	10
PostgreSQL (base de datos).....	12
Hardware	13
Introducción a Raspberry Pi.....	13
Introducción a Arduino	14
Introducción a la radiofrecuencia	14
Al detalle	16
El servidor, al detalle.....	16
Interfaz de usuario gráfica, en detalle	16
Módulo emisor de órdenes, al detalle.....	18
Módulo interruptor, al detalle.....	19
Conclusión	21
Tabla de ilustraciones	22
Bibliografía	23

Introducción

Creación de un sistema domótico inalámbrico libre. Estas siete palabras, que no casualmente coinciden con el título del proyecto, a su vez concuerdan con el objetivo del mismo, pero, ¿qué es un sistema domótico inalámbrico?

Para empezar, lo principal es entender a qué hacen referencia los conceptos: **domótico**, **inalámbrico** y **libre**. El primero, proviene de domótica, que a su vez proviene etimológicamente del latín *domus* que significa casa, y del griego *tica*, cuyo significado es: “eso que funciona por sí solo”. Por consiguiente, es un concepto que hace referencia al sistema, o conjunto de sistemas, capaces de controlar y automatizar una vivienda. El segundo, como su propio nombre indica, refiere a sin cables, que en el ámbito particular del proyecto se debe interpretar como la intención de utilizar la mínima cantidad de cables posibles. El tercero, hace referencia al concepto de *software libre* que a su vez refiere a la libertad del usuario o usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Estos términos en conjunto, representan los pilares fundamentales de la idea y el **objetivo** del proyecto: crear un **sistema** que me permita **automatizar** y **controlar** una **vivienda**, con la **mínima cantidad de cables** y una gran **libertad de software**.

El sistema que he ideado, simplificándolo al máximo, es un sistema basado en módulos. El mismo, consiste en un **servidor** que recibe **órdenes** del **usuario** a través de una **interfaz gráfica** y que comunica estas órdenes mediante **radiofrecuencia** a los **módulos receptores** que estarían colocados a lo largo de la vivienda.

En términos prácticos, las órdenes que recibe el servidor son enviadas desde una página web que cumple la función de interfaz gráfica y que son gestionadas e interpretadas por un mini-ordenador que tiene el nombre de Raspberry Pi. Este mini-ordenador, al que he vuelto capaz de emitir ondas de radiofrecuencia, traduce las órdenes en ondas y estas son recibidas por los

módulos receptores que mediante un código de identidad único saben a quién está dirigida la orden.

En términos personales, el sistema es un conjunto de ideas que he tenido desde antes de decidir proponer el trabajo al departamento. Esto se debe a que la idea de crear un sistema de automatización me persigue desde hace varios años, precisamente, cuando descubrí la domótica y con ella el mundo de la automatización. Es por ello que, llegado al bachillerato, y gracias a que mis profesores decidieron aceptar mi trabajo, pude compaginar aquella meta personal con un objetivo académico y así, como quien dice, matar dos pájaros de un tiro.

Sin embargo, no fue un tiro fácil de llevar a cabo, ya que me encontré con una gran cantidad de problemas durante el desarrollo del trabajo que investigando pude resolver. Aun así, hay dos cuya solución práctica con el marco de tiempo en el que me encontraba no pude llevar a cabo: la **seguridad** de las ondas de radiofrecuencia y la **eficiencia** energética de los módulos receptores. La solución teórica del primero sería desarrollar un algoritmo de cifrado *rolling code*, que genera un código diferente cada vez que se envía una orden y se auto sincroniza con el receptor. La solución del segundo sería reducir los circuitos a la mínima cantidad de componentes posible y utilizar una programación que permita “poner a dormir” o en estado de *standby* a los receptores.

Finalizando con la introducción, también quiero destacar que este trabajo escrito está pensado para explicar de forma bastante general todo lo que he hecho a lo largo de estos largos cinco meses. Eso quiere decir que primero explicaré el funcionamiento general del sistema de forma básica, luego los conocimientos teóricos necesarios para acabar de entenderlo perfectamente, seguido de los detalles y la conclusión.

Por último, y no por ello menos importante, comentar que toda la información utilizada ha sido extraída de Internet y todos los sitios consultados se pueden encontrar en la bibliografía.

Funcionamiento general del sistema

En este primer bloque se encuentra, como bien dice el título, la explicación del funcionamiento general del sistema, es decir, los elementos principales del proyecto y su comunicación. Todo eso, intentando utilizar la mínima terminología posible, dejando así algunos conceptos sin una explicación detallada para evitar que se pierda el hilo. Sin embargo, es importante destacar que, en los bloques de **software**, **hardware** y **en detalle** expongo los conocimientos teóricos para acabar de entender el proyecto y realizo todas las explicaciones detalladas de los elementos principales del mismo uno a uno.

El servidor

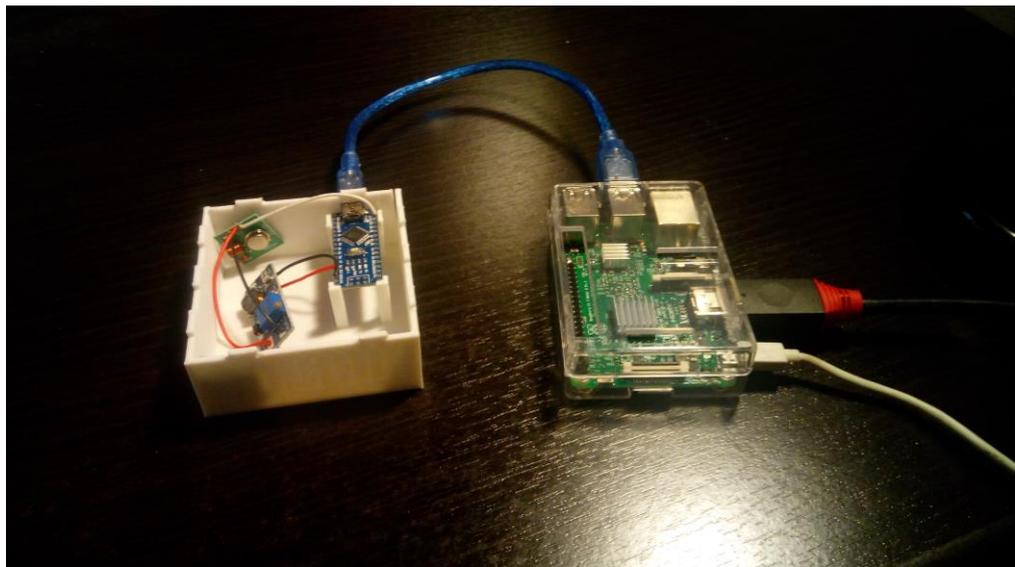


Ilustración 1: Foto del servidor (Raspberry Pi)

El servidor es la pieza central del proyecto, ya que todo gira en torno a él, a pesar de que no tendría sentido su necesidad si no fuese por ello que lo orbita, es decir, el usuario y los módulos receptores.

Esta pieza central, tiene el papel de intermediario entre las órdenes que dicta el usuario mediante la interfaz gráfica y los módulos, que reciben y ejecutan estas órdenes.

Interfaz de usuario gráfica

Para que el servidor cuente con las entradas necesarias, para saber cómo y cuándo ejecutar sus funciones, es necesario que exista una interfaz de usuario, en este caso gráfica, que no es más que el conjunto de elementos de la pantalla que permiten al usuario realizar acciones. Estas acciones, son peticiones al servidor que el mismo procesa y ejecuta dando como resultado alguna respuesta. Por ejemplo, al rellenar el formulario de inicio de sesión con un usuario y una contraseña y enviarlo, el servidor verificaría estas credenciales y decidiría si permitir el inicio de sesión o no.



Ilustración 2: Captura del inicio de sesión en la interfaz

Módulo emisor de órdenes

Una vez que el usuario realiza una orden desde la interfaz, luego de ser interpretada por el servidor, esta debe ser traducida a ondas de radiofrecuencia para que los módulos receptores sean capaces de recibirla y

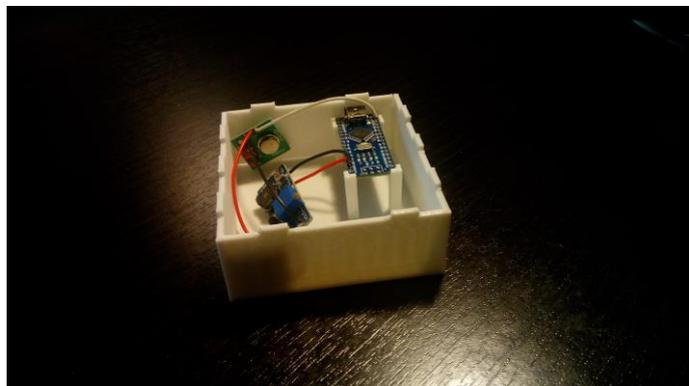


Ilustración 3: Foto del módulo emisor de órdenes

actuar en función. Para ello, el servidor al interpretar la orden del usuario genera un código compuesto de una parte única (el código identificativo del módulo receptor) y otra que indica la orden. Luego, este código se transmite al **módulo emisor de órdenes**, que, gracias a su circuito y a su programación, es capaz de transformar este código a ondas de radiofrecuencia y de transmitirlos.

Módulos receptores

Una vez se transmiten las ondas de radiofrecuencia, los receptores las captan y actúan en función de si esa onda está dirigida a ellos o no, es decir, si

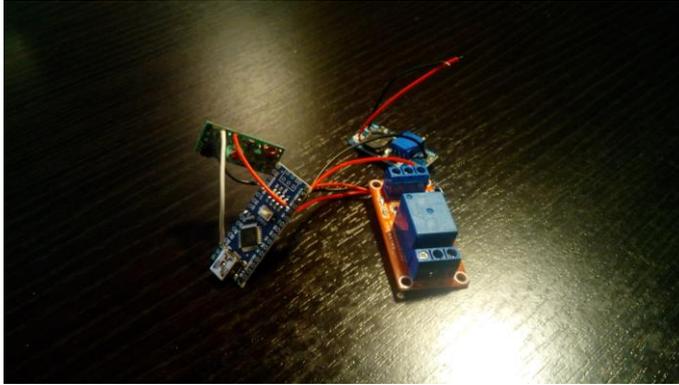


Ilustración 4: Circuito del módulo interruptor

coincide el código identificativo de la onda con el que tienen programados.

Debido a que el proyecto se trata de la creación del sistema, decidí diseñar y construir un único módulo receptor para probar la funcionalidad. Al

mismo lo he bautizado como módulo interruptor, puesto que su función es la misma que la de un interruptor. Sin embargo, por más básico que sea, es lo suficientemente capaz de demostrar que el funcionamiento general del sistema se ejecuta correctamente.

Comunicación

Para explicar la comunicación, realizaré una síntesis de todo el funcionamiento explicado anteriormente añadiendo la información pertinente, adjuntando también un diagrama para facilitar la comprensión. Primero, el usuario envía su orden desde la interfaz gráfica (1) que viaja a través de internet y es interpretada por el servidor. Segundo, el servidor (2) genera un código que es enviado al módulo emisor de órdenes (3) mediante *comunicación serial*¹. Tercero, este código es convertido y emitido en forma de ondas de radiofrecuencia que son recibidas por los módulos receptores (4).

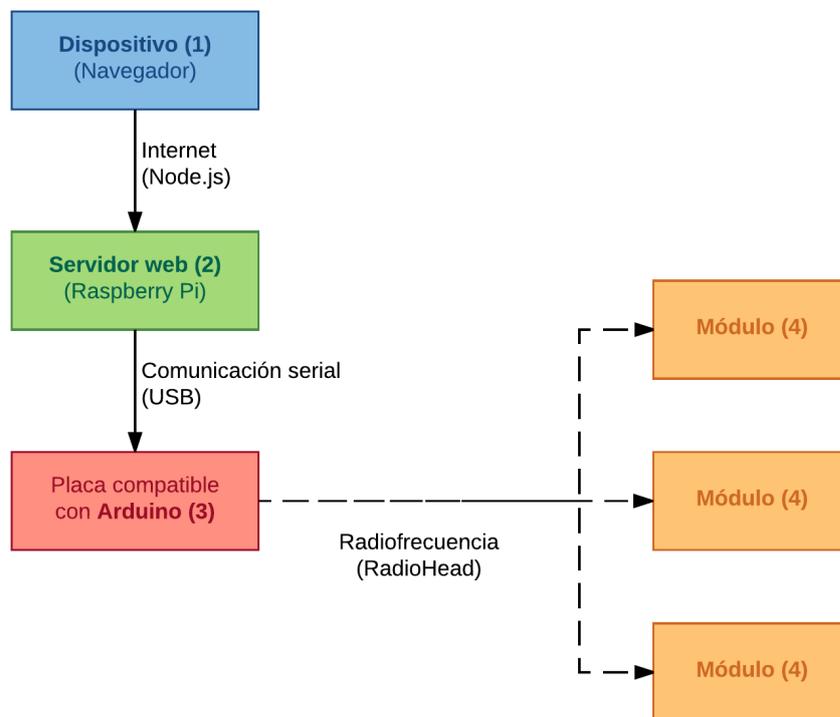


Ilustración 5: Diagrama de comunicación

¹ La comunicación serial refiere a que los bits llegan uno detrás de otro, es decir, en serie.

Cómo decía en el preámbulo del apartado del funcionamiento del sistema, aquí incluyo toda la información, en lo que a software refiere, necesaria para entender el último gran bloque del trabajo escrito.

Introducción a la programación web

La programación web es uno de los elementos clave del proyecto, puesto que la he utilizado en el servidor y en la interfaz gráfica del mismo. Para entenderla, lo primero es comprender el entorno por el cual se mueve el desarrollo web y para ello, se debe saber que en cualquier conexión web existen dos partes muy bien diferenciadas, el cliente y el servidor.

El cliente suele ser un navegador web, que se ejecuta en el dispositivo del usuario y envía peticiones al servidor. Este, por lo tanto, recibe esa petición y realiza el procesamiento, ya que en él reside el código de las diferentes páginas y las bases de datos. Una vez procesada, devuelve una respuesta, en forma de código, que es interpretada por el navegador para ser impresa en pantalla, de modo que seamos capaces de leerla e interactuar con ella.



Ilustración 6: Logo del navegador Google Chrome

En la programación web, según mi opinión, existen tres lenguajes fundamentales para escribir código: HTML, CSS y JavaScript. El primero de ellos sirve para describir la estructura y el contenido en forma de texto, así como para complementarlo con objetos e imágenes. Sus siglas provienen del inglés “HyperText Markup Language” que significa lenguaje marcado de hipertexto. El segundo, que puede estar integrado con HTML o en un archivo separado, se utiliza para definir la presentación del documento que previamente se ha estructurado con HTML, por lo cual es el complemento fundamental de este. El último, y no por ello menos importante, es JavaScript, que se está convirtiendo en uno de los lenguajes más potentes del mercado. Se

utiliza principalmente en su forma del lado del cliente, por ello, suele formar parte del navegador web, permitiendo mejoras en la interfaz de usuario con la *dinamización*² de las páginas webs.

Introducción a Node.js

Explicados ya los conceptos básicos sobre la programación web, es importante entender otro de los pilares fundamentales del proyecto: Node.js. Es un entorno Javascript ubicado en el servidor, que ejecuta este lenguaje utilizando el



Ilustración 7: Logo de Node.js

motor de procesamiento V8 desarrollado por Google. Al aprovechar este motor, Node consigue un entorno de ejecución que es capaz de *compilar*³ y ejecutar código a velocidades increíbles.

Sin embargo, la verdadera de Node.js reside en el llamado *eventloop* (bucle de eventos), ya que cuando una aplicación necesita realizar una *operación de bloqueo*⁴, Node relega la tarea al bucle con una función de respuesta, conocida en inglés como *callback* (retrollamada) que se ejecuta cuando el proceso termina.

La principal razón por la cual escogí esta tecnología, aparte de las ventajas ya comentadas anteriormente, es por una de sus librerías llamada

² La dinamización de una página web es aquello que hace que el usuario pueda interactuar con el contenido de la misma, llegando a ser capaz de modificarlo si es preciso.

³ El proceso de compilación conlleva consigo una verificación de que el código no contiene errores y una “traducción” del mismo a código máquina. Así, el ordenador es capaz de entenderlo e interpretarlo.

⁴ Una operación de bloqueo es aquella que interrumpe la ejecución del código, es decir, hasta que no se completa la operación el programa queda bloqueado.

Socket.io, que como mencione previamente permite al servidor funcionar en tiempo real.

Veo importante destacar que, a parte de esta librería, también he utilizado otras, las cuales son creadas por la extensa comunidad que envuelve a Node.js.

- **Bcrypt**, que permite encriptar contraseñas.
- **Body-parser**, que permite analizar el cuerpo de una *petición HTTP*⁵.
- **Ejs**, que permite programar plantillas con Javascript.
- **Express**, que es un *framework*⁶ minimalista y rápido para el desarrollo de aplicaciones web con Node.
- **Express-session**, que facilita el uso de las *sesiones*⁷ (necesario para el sistema de autenticación).
- **Json-file**, que permite leer y modificar archivos del tipo Json (necesario para facilitar la lectura de algunos archivos de configuración de la aplicación).
- **Pg-native**, que permite una comunicación entre Node.js y PostgreSQL (base de datos, detallado a continuación) efectiva y rápida.
- **Serialport**, que permite el acceso a los puertos seriales (necesaria para comunicarse con la placa compatible con Arduino conectada al servidor).

⁵ HTTP es sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes web y los servidores HTTP. Por lo tanto, una petición HTTP es una petición realizada al servidor mediante este protocolo.

⁶ Un framework es un entorno o ambiente de trabajo para desarrollo, que dependiendo del lenguaje normalmente integra componentes que facilitan el desarrollo organizado de aplicaciones.

⁷ Una sesión es una identificación única que se asigna a cada usuario que visita la web.

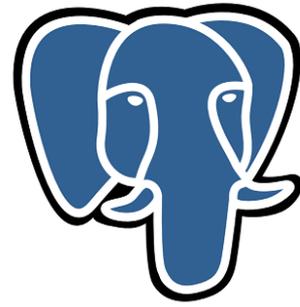
PostgreSQL (base de datos)

El concepto de base de datos se podría definir, en términos básicos, como una especie de almacén que nos permite guardar grandes cantidades de información de forma organizada con la que luego podremos trabajar fácilmente.

El sistema de gestión de base de datos que escogí para el proyecto es PostgreSQL, ya que es el sistema de código abierto (libre)

más potente del mercado. Esto se debe a que su desarrollo comenzó hace casi ya dos décadas, y durante ese tiempo: estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta.

El uso de este sistema, tiene una gran importancia, ya que, sin él, el sistema del proyecto no sería capaz de almacenar los datos de inicio de sesión y de los módulos en ningún lugar.



PostgreSQL

Ilustración 8: Logo de PostgreSQL

Hardware

Al igual que en el bloque de software, en este tercer bloque se encuentra toda la información necesaria, esta vez en lo que hardware refiere, para entender el último gran bloque.

Introducción a Raspberry Pi



Ilustración 9: Foto de una Raspberry Pi 3, con su caja y una memoria SD.

Raspberry Pi, es en esencia, una diminuta placa base resultado de un proyecto de la Universidad de Cambridge que tenía como objetivo fomentar el aprendizaje de las ciencias de la computación en niños. Lanzada al mercado en 2012, cuenta con todo el potencial de un ordenador en unas dimensiones de 84x54 milímetros. Como disco duro, utiliza una tarjeta de memoria SD y, además, cuenta con múltiples conexiones como USB, HDMI, WiFi, etcétera.

Este ordenador, da vida al servidor, ya que en él se almacena todo el software necesario para el funcionamiento del sistema.

Introducción a Arduino

Arduino es una plataforma *open source* para crear prototipos de electrónica, cuyos principios son contar con hardware y software fáciles de usar. Esto se debe a que está pensada para satisfacer las necesidades de cualquier persona que esté interesada en crear entornos u objetos interactivos.

Esta plataforma es un circuito que tiene como cerebro un microcontrolador que se programa utilizando el lenguaje de programación (Arduino Programming Language) y el entorno de desarrollo (Arduino Development Environment) de Arduino.

Sus principales ventajas se encuentran en la facilidad para crear circuitos programados y su característica de ser multiplataforma, ya que su entorno de desarrollo se puede ejecutar en sistemas operativos Windows, Macintosh OSX y GNU/Linux. Además, es un entorno fácil de usar para principiantes y lo suficientemente flexible para que usuarios avanzados lo puedan aprovechar.



Ilustración 10: Logo de Arduino

Introducción a la radiofrecuencia

La radiofrecuencia es una tecnología que utiliza ondas electromagnéticas que viajan casi a la velocidad de la luz para comunicar información de un punto a otro. Los parámetros primordiales en la comunicación con radiofrecuencia son la potencia de transmisión, la velocidad de transmisión de la información o *data rate* y la sensibilidad del receptor. La **potencia de transmisión** es la potencia de las ondas electromagnéticas que emite el transmisor, medida en watts. La **sensibilidad del receptor** es la mínima señal con la que el receptor puede decodificar.

Un ejemplo de esto se puede hacer utilizando las ondas de sonido: la potencia de transmisión sería “lo fuerte” que alguien está llamando a una

puerta y la sensibilidad de recepción sería “lo suave” que una persona puede sentir el golpe.

También, a parte de estos parámetros, vale la pena destacar que la radiofrecuencia trabaja con señales de amplitud modulada, es decir, señales que para la transmisión de información cambian (modula) la amplitud de la onda.

En el caso particular del proyecto, he decidido utilizar unos módulos que funcionan en una frecuencia pública (433Mhz) y que normalmente se utilizan en puertas de garajes y alarmas de coches.



Ilustración 11: Módulos transmisor y emisor de radiofrecuencia

Al detalle

Este es el último gran bloque del trabajo escrito, aquí procederé a explicar al detalle, cada uno de los componentes particulares del trabajo.

El servidor, al detalle

El servidor es concretamente un servidor web, programado con Node.js y utilizando una base de datos gestionada con PostgreSQL, todo ello, instalado en el mini-ordenador llamado Raspberry Pi. En el ámbito del proyecto, este servidor cumple con tres funciones básicas:

- Gestionar los usuarios, ya que proporciona seguridad y privacidad exigiendo un inicio de sesión con usuario y contraseña.
- Gestionar los módulos receptores instalados, permitiendo asignar nombres de identificación personalizados y permisos a los usuarios escogidos.
- Ser el intermediario entre el usuario y los receptores.

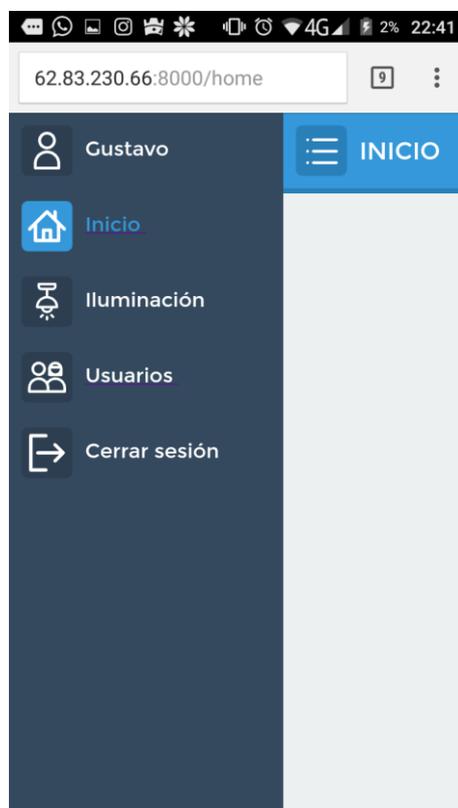
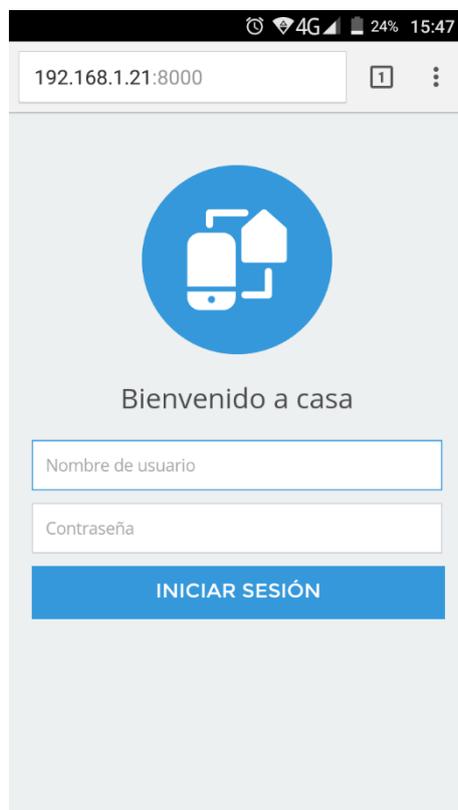
Interfaz de usuario gráfica, en detalle

La interfaz de usuario es simplemente una página web a la cual el usuario accede mediante unas credenciales previamente configuradas que le permiten ejecutar órdenes y realizar modificaciones en la base de datos de forma simple e intuitiva, ya que el servidor, programado paralelamente, se encarga de la parte difícil.

Para el diseño, he escogido tonos azules de color plano que junto al *diseño responsivo*⁸ dan una perfecta estética dirigida concretamente a los

⁸ El diseño responsivo es un diseño que responde al tamaño del dispositivo en el que se está visualizando la web, adaptando así las dimensiones del contenido y mostrando los elementos de una forma ordenada y optimizada sea cual sea el soporte.

dispositivos móviles. A continuación, en la siguiente página, adjuntaré varias capturas de la misma hechas directamente con mi móvil de uso personal.



Módulo emisor de órdenes, al detalle

El módulo emisor de órdenes consiste en una placa compatible con Arduino a la cual le he conectado un transmisor de radiofrecuencia. Como comenté en el apartado del funcionamiento, el módulo emisor de órdenes se encarga de transmitir las ondas de radiofrecuencia que incluye el código que genera el servidor. Para ello, el servidor transmite el código a la placa que está conectada por USB mediante comunicación serial⁹, y el programa que he cargado en la placa se encarga de transmitir ese código en forma de ondas electromagnéticas.

A continuación, adjuntaré un diagrama de conexiones del circuito, pero antes debo comentar un elemento del mismo llamado step up, que tiene la función de aumentar el voltaje de la corriente que alimenta el transmisor, así, este aumento en la tensión amplía el alcance del mismo.

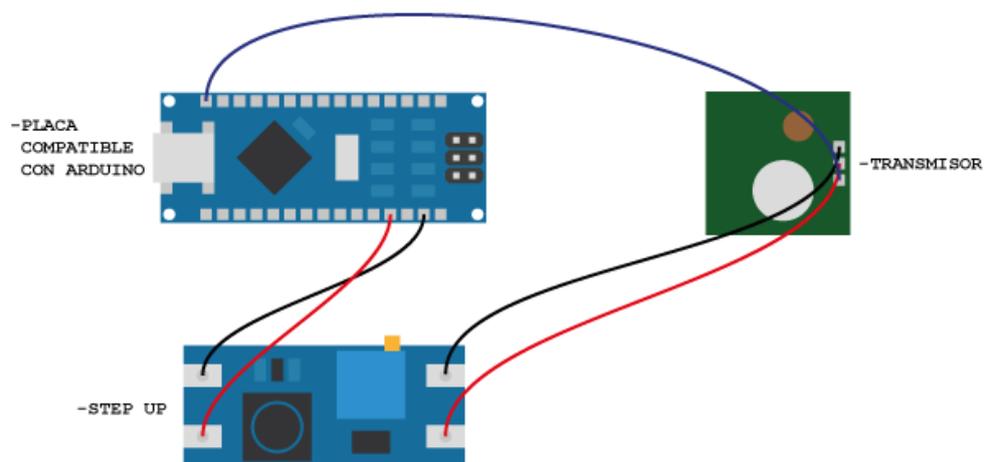


Ilustración 12: Diagrama de conexiones (módulo emisor de órdenes)

Además, el circuito está colocado dentro de una caja diseñada en *SketchUp*¹⁰ e impresa con una impresora 3D. A continuación, adjuntaré una impresión de pantalla del programa de diseño con la caja.

⁹ La comunicación serial refiere a un tipo de comunicación por cable mediante la cual los bits se envían uno detrás de otro.

¹⁰ SketchUp es un programa de diseño en 3D creado y distribuido por Google.

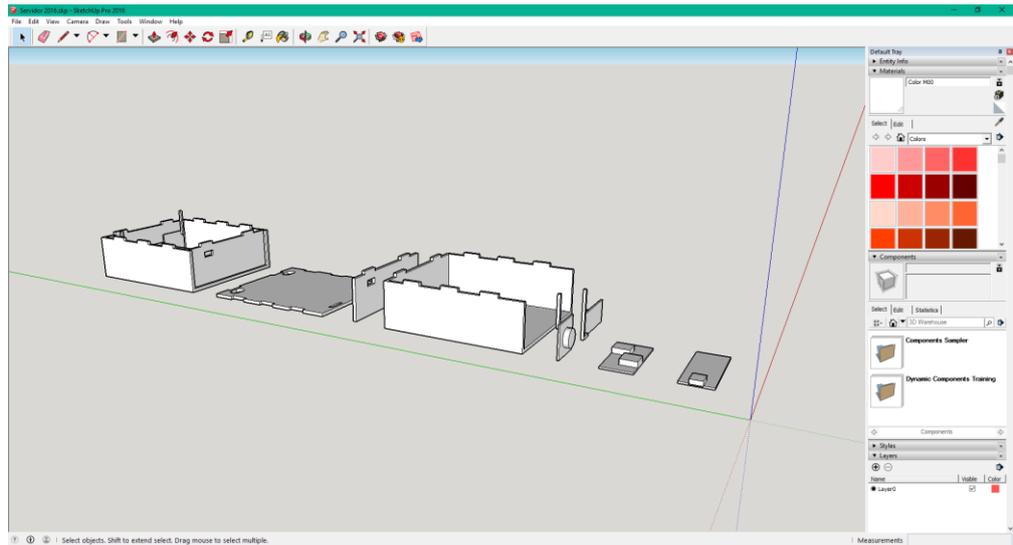


Ilustración 13: Impresión de pantalla, caja del módulo emisor de órdenes

Módulo interruptor, al detalle

El módulo interruptor, como explicaba anteriormente, fue el único que pude diseñar y construir con el margen del tiempo con el que contaba. Su circuito consta de otra placa compatible con Arduino, un relé¹¹, un step up, una batería de gran capacidad y un receptor de radiofrecuencia.

Todos estos componentes se encuentran conectados como explica el diagrama a continuación:

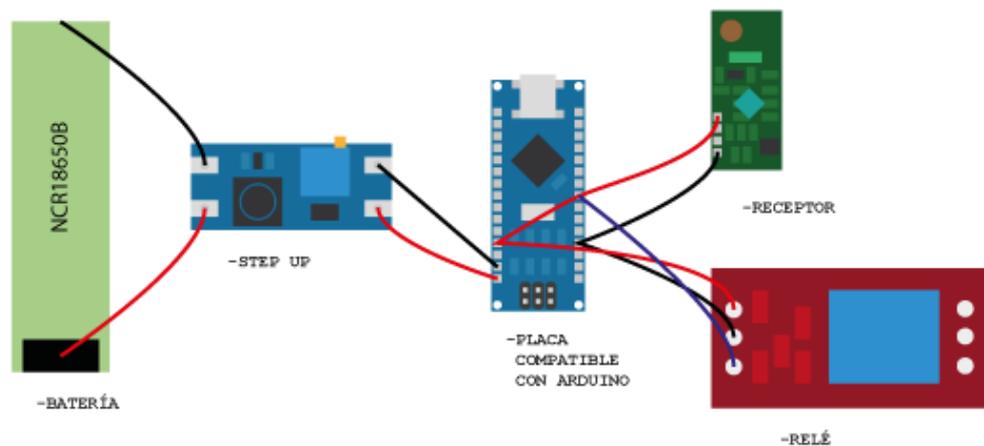


Ilustración 14: Diagrama de conexiones (módulo de iluminación)

¹¹ Un relé es un interruptor electromagnético que permite controlar y, por lo tanto, por así decirlo, “unir” dos corrientes de diferentes voltajes.

En el circuito, las baterías alimentan el microcontrolador, el receptor se encarga de trabajar con las ondas de radiofrecuencia y el step up se utiliza esta vez para aumentar el voltaje de la pila (3.7V) a 5v que es lo que necesita para trabajar Arduino y el receptor. El relé, se encarga de separar la corriente de 220v de la corriente de 5V que ofrece la batería para alimentar el microcontrolador, dando la capacidad de controlar mediante los 5V de la placa compatible, un interruptor con una conexión a 220V. Esto significa que el microcontrolador se vuelve capaz de controlar el estado de una corriente de 220v sin tener contacto directo con ella.

Al igual que el módulo emisor de órdenes, también tiene su respectiva caja diseñada en Sketchup, de la cual también adjuntare la captura de pantalla a continuación.

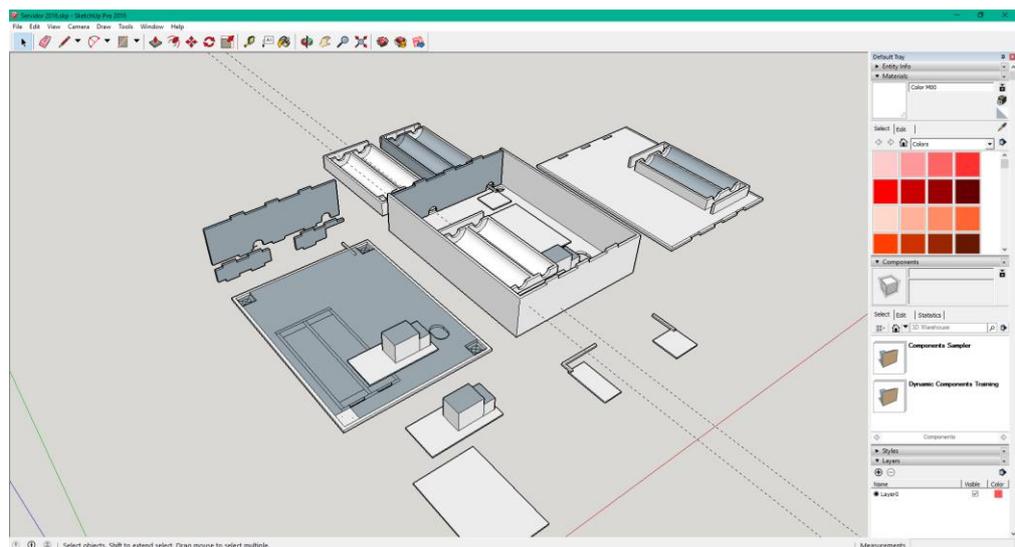


Ilustración 15: Impresión de pantalla, caja del módulo interruptor

Conclusión

Después de tres placas compatibles con Arduino quemadas, un par de transmisores y receptores de radiofrecuencia que dejé inservibles, enfados con el código y varias soldaduras fallidas, puedo decir que finalmente, y satisfactoriamente, he finalizado el proyecto.

He de decir que fue una actividad muy fructífera, que me aportó importantes conocimientos y retos, los cuales pude superar algunas veces fácilmente y otras todo lo contrario. Sin embargo, el gran problema llega ahora, ya que solamente queda concluir, lo cual es la parte que considero más difícil. Quiero decir, no se me hace fácil explicar algo que tiene una función tan práctica, que funcionando ya demuestra la propia conclusión sin escribir ni una palabra. Es por ello, que antes de terminar, justificaré dos aspectos del proyecto que he decidido no explicar, pero que no interfieren en el entendimiento del mismo, pero que son importantes para comprender la corta extensión de este trabajo escrito.

El primero de ellos, la **programación** de los microcontroladores, del servidor web y de la página web. Con una totalidad aproximada de más de 3000 líneas de código, explicarlo todo conllevaría una eternidad. Por esa razón, decidí dejarlo fuera del trabajo escrito, ya que para entenderlo se requieren conocimientos de los lenguajes utilizados, de lógica, etcétera. Aun así, en caso de que el lector los tenga, todos los archivos y las instrucciones de instalación se encuentran en una entrada de mi blog personal (gempdev.hol.es) que he abierto pensando precisamente en este nuevo año y en el trabajo.

El segundo, la profundización de las partes más técnicas, en las cuales decidí no ahondar demasiado buscando darle la mayor universalidad posible al trabajo escrito, para que así el funcionamiento del sistema sea entendido por cualquier lector.

Ahora, justificados estos aspectos, solo me queda finalizar volviendo a dar las gracias a todas aquellas personas que de alguna manera hicieron que esto fuese posible.

Tabla de ilustraciones

Ilustración 1: Foto del servidor (Raspberry Pi)	5
Ilustración 2: Captura del inicio de sesión en la interfaz	6
Ilustración 3: Foto del módulo emisor de órdenes.....	6
Ilustración 4: Diagrama de comunicación	8
Ilustración 5: Logo del navegador Google Chrome	9
Ilustración 6: Logo de Node.js	10
Ilustración 7: Logo de PostgreSQL	12
Ilustración 8: Foto de una Raspberry Pi 3, con su caja y una memoria SD. ...	13
Ilustración 9: Logo de Arduino	14
Ilustración 10: Módulos transmisor y emisor de radiofrecuencia	15
Ilustración 11: Diagrama de conexiones (módulo emisor de órdenes)	18
Ilustración 12: Impresión de pantalla, caja del módulo emisor de órdenes....	19
Ilustración 13: Foto de la caja del módulo emisor impresa	Error! Bookmark not defined.

Bibliografía

¿Qué es Arduino?, Arduino.cl [en línea].

<<http://arduino.cl/que-es-arduino/>>

Domótica, Wikipedia, la enciclopedia libre [en línea].

<<https://es.wikipedia.org/wiki/Dom%C3%B3tica>>

¿Qué es domótica?, CEDOM, Asociación Española de Domótica e Inmótica [en línea].

<<http://www.cedom.es/sobre-domotica/que-es-domotica>>

Domótica: ventajas y desventajas, blog domoactualidad [en línea].

<<http://domoactualidad.blogspot.com.es/p/ventajas-y-desventajas.html>>

Introducción a la programación web, blog de Ignacio Perez Martin [en línea].

<<https://perezmartin.es/tema-1-introduccion-a-la-programacion-web/>>

¿Qué son los frameworks?, blogNubelo [en línea].

<<http://blog.nubelo.com/que-son-los-frameworks/>>

Qué es una librería, about en español [en línea].

<<http://aprenderinternet.about.com/od/Glosario/fl/Que-es-una-libreria.htm>>

Node.js: Qué es y para qué sirve Node.js?, NetConsulting [en línea].

<<http://www.netconsulting.es/blog/nodejs/>>

Sesiones en PHP: qué son y cómo funcionan, Arume Informatica [en línea].

<<http://www.arumeinformatica.es/blog/sesiones-en-php-que-son-y-como-funcionan/>>

Técnicas para dinamizar una página web, Gestipolis [en línea].

<<http://www.gestipolis.com/tecnicas-dinamizar-pagina-web/>>

El protocolo HTTP, Herramientas web para la enseñanza de protocolos de comunicación [en línea].

<<http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html>>

¿Qué es compilar código?, El Blog de Liher [en línea].

<<http://elblogdeliher.com/que-es-compilar-codigo/>>

Radiofrecuencia básica, Cesar Augusto Zapata [en línea].

<<http://es.slideshare.net/guest3b8998/radio-frecuencia-basica>>

Principios básicos de la radiofrecuencia, Santiago Espinosa [en línea].

<<http://www.saigesp.es/rf-principios-basicos/>>

