

PFC

# Estudio Plataformas IoT

Ruben Loureiro Garrido  
12-6-2015

# Indice

1. Presentación.....	3
1.1.    Introducción.....	3
1.2.    Objetivo.....	4
1.2.1.    Objetivos del Proyecto.....	4
2. Plataformas Software.....	5
2.1.    ThingSpeak.....	5
2.1.1.    Características principales.....	6
2.1.2.    Ámbitos de aplicación.....	7
2.1.3.    Conclusiones.....	8
2.2.    Carriots.....	8
2.2.1.    Características Principales.....	9
2.2.2.    Ámbito de Aplicación.....	10
2.2.3.    Conclusiones.....	11
2.3.    Electronic Imp.....	11
2.3.1.    Características Principales.....	11
2.3.2.    Ámbitos de aplicación.....	13
2.3.3.    Conclusiones.....	13
2.4.    Spark.....	14
2.4.1.    Características Principales.....	14
2.4.2.    Ámbitos de aplicación.....	15
2.4.3.    Conclusiones.....	15
2.5.    Blaulabs.....	15
2.5.1.    Características principales.....	16
2.5.2.    Ámbitos de aplicación.....	16
2.5.3.    Conclusiones.....	17
2.6.    Thinking Things.....	17
2.6.1.    Características principales.....	18
2.6.2.    Ámbitos de aplicación.....	19
2.6.3.    Conclusiones.....	19
2.7.    Zatar.....	19
2.7.1.    Características principales.....	20
2.7.2.    Ámbito de aplicación.....	21
2.7.3.    Conclusiones.....	21
2.8.    Comparativas Plataformas Softwares.....	22
3. Campos de aplicación en IoT.....	23
3.1.    Smart Home.....	23
3.2.    Smart City.....	24
3.3.    Agricultura Inteligente.....	24
3.4.    Wearables.....	25
4. Análisis Plataformas Hardware en IoT.....	26
4.1.    Arduino.....	26
4.2.    Waspmote.....	28
4.3.    Spark.....	29
4.4.    Intel Galileo.....	30
4.5.    Zigbee32.....	32

4.6.	Comparativa plataformas HW .....	34
5.	Aplicación IoT usando Arduino + Thingspeak.....	35
5.1.	Elección Plataformas.....	36
5.2.	Funcionalidad.....	36
5.2.1.	Flujo Llegada de un Tweet.....	36
5.3.	Plataforma Hardware37 .....	37
5.3.1.	Instalación IDE Arduino.....	37
5.3.2.	Conexión WiFi Shield.....	37
5.3.3.	Montaje del Circuito.....	39
5.4.	Plataforma ThingSpeak.....	40
5.4.1.	Aplicación ThingTweet.....	41
5.4.2.	Aplicación TweetControl + ThingHttp. ....	42
5.4.3.	Aplicación TalkBack.....	45
5.4.4.	Aplicación REACT + ThingHTTP.....	47
5.5.	Pruebas realizadas.....	51
5.5.1.	Medida tomada por Arduino y enviada a ThingSpeak.....	51
5.5.2.	Envío de un Tweet desde Plataforma Thingspeak.....	51
5.5.3.	Tweet enviado a ThingSpeak para encender nuestro ventilador. ....	52
5.5.4.	Tweet enviado a ThingSpeak para encender nuestro ventilador. ....	53
5.6.	Presupuesto del Proyecto.....	54
6.	Conclusiones .....	55

## 1. Presentación

## 1.1. Introducción.

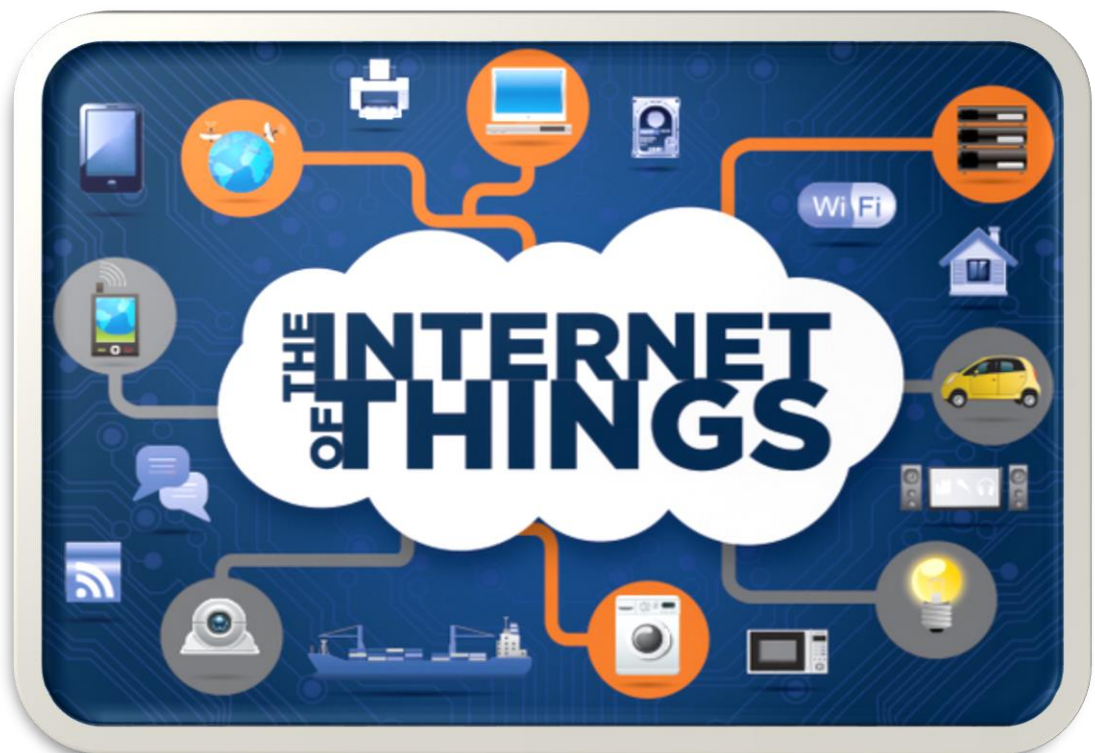
La primera vez que se propuso el concepto del Internet de las Cosas fue en 1999 por **Kevin Ashton**, en el Auto-ID Center del MIT, ahí se presentó un sistema de sensores y e identificadores de radiofrecuencia (RFID).

**“El IoT es el mundo en el que cada objeto tiene una identidad virtual propia y capacidad potencial para integrarse e interactuar de manera independiente en la Red con cualquier otro individuo, ya sea una máquina (M2M) o un humano.”**

La necesidad de los últimos años de estar constantemente conectados y poder cuantificar cualquier cosa, da lugar a lo que hoy día conocemos como Internet de las Cosas. Según Wikipedia la IOT es la interconexión de objetos cotidianos con Internet.

Nos encontramos ante una nueva revolución tecnológica, dónde objetos cotidianos del día a día con una función específica, evolucionan gracias al IoT, pasando a estar conectados y dotar de nuevas funcionalidades a estos objetos, pudiéndolos controlar y administrar desde tabletas, ordenadores o teléfonos móviles.

Los usos y aplicaciones que se pueden dar son muy variadas, y toca prácticamente todos los ámbitos de la vida humana, desde la vivienda, ciudad, salud, industria, consumo, etc.



## 1.2.Objetivo

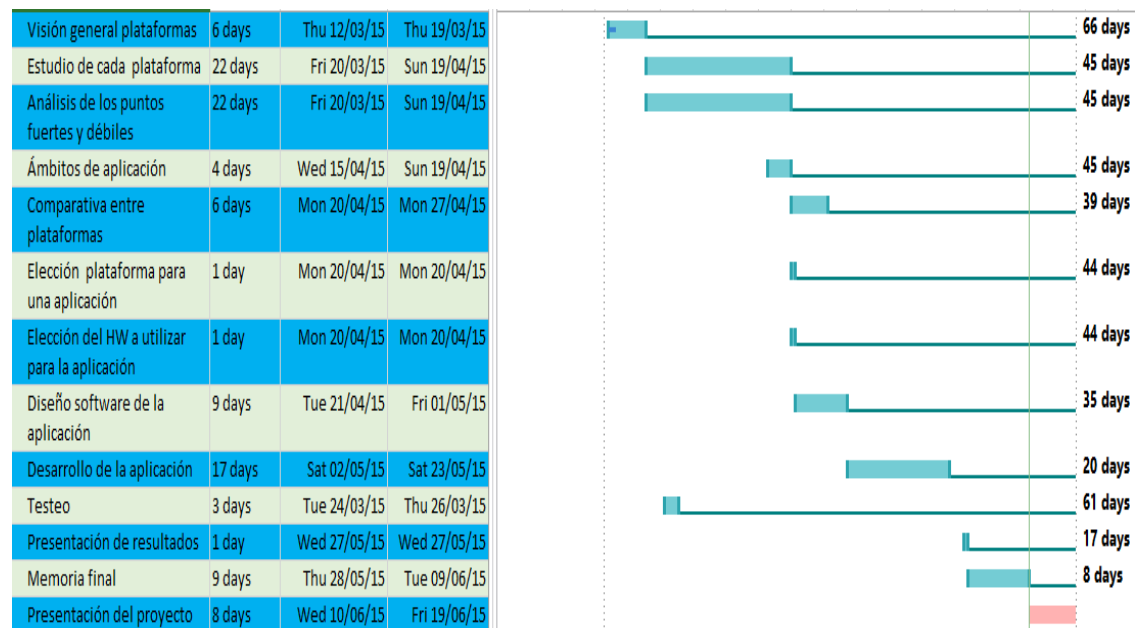
***El objetivo principal de este proyecto, consiste en dar una visión global del Internet de las Cosas, se comenzará describiendo los ámbitos de aplicación más importantes hoy día, siguiendo por el estudio de las principales plataformas software / hardware y por último se presenta una aplicación real, cuyo objetivo principal es demostrar de una forma práctica lo descrito anteriormente.***

***Con el desarrollo de la aplicación, se intentará dar una visión de cómo a través de un microcontrolador podemos conectarnos a una plataforma software y empezar a subir los datos. Otro de los puntos interesantes que queremos demostrar con la aplicación, es que hacer con esos datos y cómo tratarlos.***

### 1.2.1. Objetivos del Proyecto

- Conocer que es el Internet de las Cosas.
- Conocer los ámbitos de aplicación más importantes del IoT.
- Estudio de las principales plataformas Software del IoT.
- Estudio de las principales plataforma Hardware del IoT.
- Diseño de una aplicación real.
- Desarrollo de una aplicación usando las plataformas estudiadas.
- Tratamiento de datos de IoT.

Seguendo con el plan de trabajo los siguientes pasos a seguir serán el desarrollo de una aplicación.



## 2. Plataformas Software.

*En este punto nos vamos a centrar en el estudio de las principales plataformas software que hay actualmente en el mercado. Algunas de estas plataformas son Open Source, otras propietarias, en algunas encontraremos un ecosistema IoT, esto quiere decir que la propia plataforma nos ofrecerá software, hardware y todo lo necesario para desarrollar nuestra aplicación.*

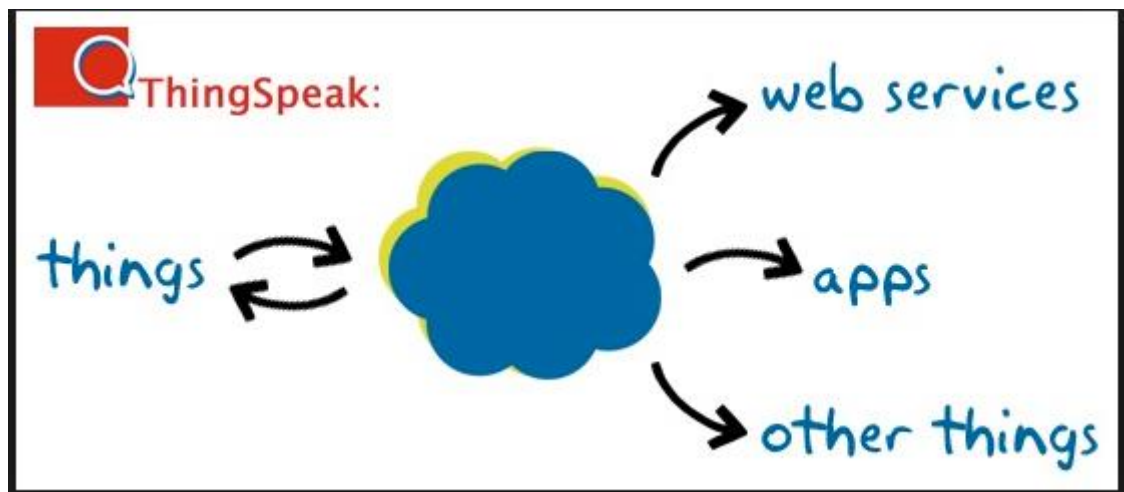
*En cada plataforma nos centraremos, en conocer sus características principales, que hardware es el idóneo para cada plataforma, sus ámbitos de aplicación y su propósito general, bien si este es orientado a la educación o por el contrario más orientado a un ámbito profesional.*

### 2.1. ThingSpeak

ThingSpeak es una plataforma abierta de aplicaciones, diseñada para permitir conectar personas con objetos. Se caracteriza por ser una plataforma Open Source con una API para almacenar y recuperar datos de los objetos usando el protocolo HTTP sobre Internet o vía LAN (Local Area Network).

Se trata de una plataforma basada en Ruby on Rails 3.0 (RoR), este es un framework de aplicaciones web de código abierto basado en Ruby, cuya arquitectura está basada en el Modelo Vista Controlador (MVC). Se caracteriza por su simplicidad a la hora de programar aplicaciones del mundo real, escribiendo menos código y con una configuración mucho más sencilla que otros frameworks. Otra de las características que hacen de RoR un framework perfecto para el desarrollo de aplicaciones es que permite el uso de meta programación, haciendo que su sintaxis sea más legible y llegue a un gran número de usuarios.

La aplicación incluye todo lo necesario para poder empezar a trabajar, desde una aplicación web en la que podremos gestionar usuarios, gestionar claves de API, gestión de canales y cartografía.



### 2.1.1. Características principales.

Destacaremos algunos puntos importantes en toda plataforma tales como su API, App (si las tiene), integración, hardware:

- **API**

Un punto importante a la hora de desarrollar cualquier proyecto es encontrar un API disponible de forma sencilla para que el desarrollador tenga los mecanismos necesarios para el desarrollo de la aplicación. En este caso, ThigSpeak dispone de una **API** la cual está disponible en [GitHub](#) para su descarga en un servidor propio. Es totalmente abierta, por lo que también se puede modificar su código fuente original y así contribuir a la comunidad con nuevas características, un principio básico en toda plataforma Open Source.

- **Canales**

La forma que tiene esta plataforma de almacenar y publicar los datos es a través de los “Chanel” (Canales). Su creación es muy simple y en un par de clicks y rellenando una serie de datos lo tendremos disponible sin mayor complicación. Nuevamente y según profundizas en esta plataforma ves que el objetivo es simplificar el trabajo.

- **Plugins**

Para extender la funcionalidad del sitio también se nos brinda la oportunidad de desarrollar plugins. Estos nos ofrecen la posibilidad de crear aplicaciones de forma nativa en nuestra plataforma ThigSpeak.

Soporta HTML, CSS y JavaScript como lenguajes de programación. Al igual que los canales los plugins pueden ser público o privados según sean nuestras necesidades.

Por último, destacar que ofrece la posibilidad de usar Google Gauge Visualization, gracias al cuál nos ofrece la posibilidad de visualizar los datos de una forma rápida y amigable, con un nivel de personalización muy amplio.

- **Integración**

Uno de los puntos fuertes en cualquier plataforma IoT, es que permita una amplia integración con diversos dispositivos Hardware y software. En este caso ThingSpeak permite la integración de su plataforma con:

- Arduino
- Raspberry Pi
- IoBridge / RealTime.io
- Electric Imp
- Móviles / Aplicaciones web
- Redes Sociales
- Análisis de datos con MATLAB

- **Apps en ThingSpeak**

Sin duda las aplicaciones que encontramos en ThingSpeak son un complemento perfecto para nuestros proyectos, dotándolos en muchos casos de unas funcionalidades muy interesantes, actualmente tienen las siguientes aplicaciones en catálogo.

- ✓ **ThingTweet** : Esta aplicación actúa como proxy permitiendo a nuestro dispositivo enviar actualizaciones de status a Twitter. Esto se consigue gracias a una llamada a la API de ThingTweet.
- ✓ **TweetControl**: Esta aplicación se pone a escuchar los hashtags de Twitter y luego permite controlar cualquier cosa que puedas imaginar. Cualquier cosa que se pueda conectar a través de ThingHTTP también se puede conectar a Twitter.
- ✓ **ThingHTTP**: Con ThingHTTP permite a un microcontrolado o dispositivo a bajo nivel conectarse a un web service a través de Internet o la propia red.  
Soporta los métodos GET,POST,PUT y DELETE, SSL y Autenticación Básica.
- ✓ **React**: Básicamente lo que realiza esa app, es ejecutar una acción cuando se cumple una cierta condición para los datos en nuestro canal. Por ejemplo, podríamos encender nuestra cafetera según llegamos a casa del trabajo, creando un **React** de Geolocalización.
- ✓ **TalkBack**: Permite que cualquier dispositivo pueda actuar sobre los comandos en cola. Por ejemplo, podremos controlar una puerta añadiendo comandos tales como OPENDOOR,CLOSEDOOR usando web interface o API. El funcionamiento podría ser el siguiente si alguien se acerca a la puerta esta se abre, si después de un par de minutos no se detecta movimiento esta se cierra.
- ✓ **TimeControl**: Con TimeControl a un determinada hora, podremos ejecutar un ThingHTTP o un ThingTweet, e incluso añadir un nuevo comando a TalkBack.

### 2.1.2. Ámbitos de aplicación.

Esta plataforma reúne a una comunidad que sobre todo se inicia en el mundo del IoT, por lo que principalmente nos encontramos con proyectos que son “prototipos” aunque por supuesto podemos encontrar proyectos que más profesionales. Por lo que principalmente esta orientado al mundo del **Smart Home**.



### 2.1.3. Conclusiones

*Estamos ante una plataforma que recomendaría para todo aquel interesado en iniciarse en el internet de las cosas. Sin duda dispone de una interfaz muy amigable con todos los mecanismos y tecnologías necesarios para cualquiera que necesite monitorizar, cualquier dispositivo.*

*Hay que destacar también el apartado dedicado a las aplicaciones, facilitan mucho la tarea a la hora de desarrollar cualquier proyecto.*

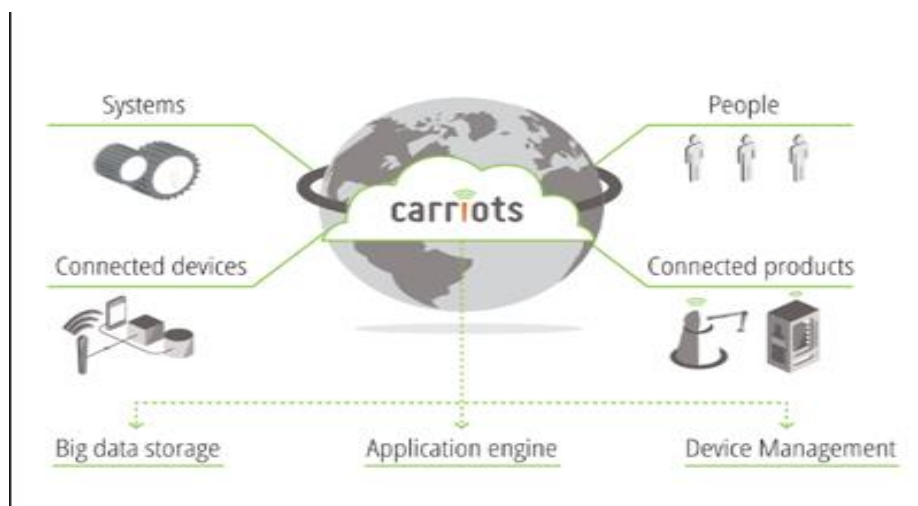
***Thingspeak será la plataforma elegida para nuestra aplicación real. Los motivos por los que se ha elegido esta plataforma son primeramente, por ser una plataforma Open Source , por su fácil configuración, el tener una API documentada es también uno de los motivos por los se ha elegido esta plataforma. Disponer de una integración sencilla con aplicaciones de terceros, como por ejemplo Twitter, u otras aplicaciones como ThingHttp de fácil configuración. Otro motivo fundamental es su compatibilidad con Arduino y el disponer de documentación referente a esta tipo de hardware.***

## 2.2. Carriots

***Carriots es una plataforma Española en la nube, que da un servicio PaaS orientada a proyectos IoT y máquina a máquina (M2M).***

***Carriots no es una plataforma open source como hemos comentado anteriormente es una empresa que ofrece un servicio PaaS. Aún así, podemos darnos de alta y registrar un máximo de 10 dispositivos de forma totalmente gratuita pero con alguna que otra restricción en lo que a funcionalidad se refiere. Para registrar más de 10 dispositivos y hacer uso de todas las funcionalidades que ofrece la plataforma sin ningún tipo de restricción hay que pagar, si bien es verdad el precio no es desorbitado siendo este de 2€ por dispositivo extra. También ofrece un servicio para conectar nuestros dispositivos a nuestra nube privada, aunque de este servicio no detalla precio.***

***Desde la propia web de Carriots aseguran que cualquier dispositivo es compatible con su plataforma siempre y cuando tengan acceso a internet, más adelante profundizaremos sobre que dispositivos HW son compatibles aunque adelante que tienen un amplio catálogo.***



### 2.2.1. Características Principales

Como hemos comentado en la introducción, Carriots es una plataforma propietaria, se caracteriza sobre todo por una gran compatibilidad de hardware, una API muy completa y documentada con una gran integración con aplicaciones de terceros. A continuación detallamos sus puntos más importantes:

- **API**

Carriots nos ofrece una API propia basada en REST, (la cual puede descargarse desde Github) para poder comunicarnos con la plataforma y gestionar los datos de una forma sencilla. REST se caracteriza principalmente por la existencia de recursos que pueden ser accedidos utilizando un identificador global. Para la manipulación de estos recursos, la comunicación de red se usa el estándar HTTP.

Respecto a los mecanismos de seguridad de la REST API destaca sobre todo el uso de API Keys se caracteriza por un Token que sustituye al clásico user+pass, Checksum es un hash de datos +fecha +clave con una clave simétrica y por último el ya de sobra conocido HTTPS.

- **Integración**

Como venimos comentando anteriormente, sin duda un punto importante en este tipo de plataformas es evaluar la integración que nos ofrece con sistemas externos. En el caso que nos ocupa la plataforma Carriots nos ofrece esta integración con sistemas externos, a través de su API REST, el PUSH de datos y con peticiones HTTP/s o sockets. Con estos mecanismos podremos tener integración con bases de datos, ERPs, CRM, data warehouse, etc.

Carriots nos ofrece ya mecanismos integrados que podrán ser accesibles mediante su SDK. La lista de mecanismos integrados son: DropBox, Twitter, Mailing, SMS internacionales y Sockets.

- **Hardware**

En este apartado Carriots ofrece un amplio catálogo de hardware compatible, la verdad que es muy completo y prácticamente soporta cualquier tipo de hardware que haya en el mercado.

- Arduino
- Raspberry Pi
- Beagle Bone
- Fez Cerbuino
- Cubie Board
- TST Gate
- TST Mote
- CloudGate
- Nanode
- Electric IMP
- TST Light
- Mobile Devices C4Max
- Mobile Devices C4EVO
- Mobile Device OBD Ongle
- Electric IMP April Development Board
- We500
- Satel Dataloggers DL170,DL171 y OWA31IETH
- Otro Hardware (Siempre y cuando pueda comunicarse con su API REST)

### 2.2.2. **Ámbito de Aplicación.**

Estamos ante plataforma que dispone de una amplia oferta en este punto, con diversos proyectos y distintos casos de usos, los ámbitos más importantes se recogen a continuación:

- Smart City
- Smart Energy
- Smart Oil
- Smart Agiculture
- Smart Buildings
- Smart Retail
- Smart Banking
- Smart Consumer
- Smart Loguistic

### 2.2.3. Conclusiones

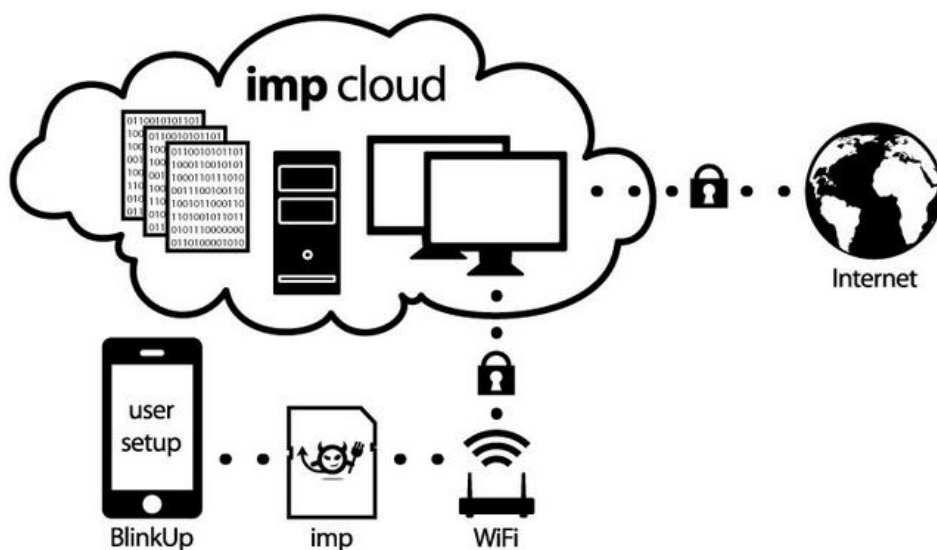
*Es una plataforma que ofrece una gran variedad de soluciones de negocio orientadas al Internet de las Cosas, en distintos ámbitos de aplicación. Dispone de una API muy bien documentada, hay que destacar también su gran compatibilidad con distintos tipos de hardware, algo que la hace muy versátil y escalable. Otro punto a favor de esta plataforma a tener muy en cuenta es su integración con aplicaciones de tercero, facilitará mucho el trabajo y lo dotará de un extra de funcionalidades.*

*Esta plataforma no está, tan orientada a prototipos, aunque trae una opción gratuita, sino que claramente su línea de negocio son aplicaciones reales basadas en el IoT en los distintos ámbitos de aplicación que dispone.*

### 2.3. Electronic Imp

*Esta plataforma abarca tanto soluciones hardware como software en forma de servicios. Se caracteriza por proveer soluciones finales con total integración hardware, sistemas operativos, APIs, servicios en la nube, con total seguridad y rapidez. Según su web comentan que reducen los costes de manera significativa. La plataforma se basa en su módulo Wifi, el cual tiene una completa integración con cualquier dispositivo hardware, y software. Básicamente este módulo Wi-Fi, actúa como un gateway para conectar un dispositivo a internet independientemente del hardware del dispositivo.*

*Esta plataforma se caracteriza sobre todo por tener un ecosistema de productos muy completa y variada, del que a continuación mostraremos sus productos más relevantes y así poder hacernos una idea de lo completa que es esta plataforma*



### 2.3.1. Características principales

**ImpOs** es el core de la plataforma Electronic Imp que a través del dispositivo y su agente online provee los mecanismos necesarios para satisfacer nuestras necesidades a la hora de desarrollar cualquier aplicación.

Dispone de una Api abierta la cual nos permite gestionar la conectividad , energía, seguridad, notificaciones, etc. El código de nuestra aplicación correrá sobre una máquina virtual que nos ofrece impOs

Cloud Es la pieza central de la plataforma Electronic imp, para conectar nuestros dispositivos a través de internet, se le identifica mediante un único agente online, para así poder tener libertad total a la hora de programar un dispositivo. La plataforma Cloud nos ofrece un servicio Saas, nos ofrece almacenamiento de datos, visualización de reportes, es el agente quien gestiona este tipo de comunicaciones todo ello bajo el protocolo HTTPS.

- **API**

El uso de una API abierta permite la personalización de nuestra plataforma y adaptarla a nuestras necesidades, en este caso Electronic imp ha elegido el lenguaje Squirrel. Se trata de un Api que hace un tratamiento de objeto muy variados. Por otro lado, la API dispone de los métodos necesarios de seguridad para proteger las comunicaciones entre internet y el dispositivo.

- **Hardware**

Como hemos comentado Electronics Imp comenzó con la implementación de los módulos Wi-Fi cuya función era de actuar como un Gateway para conectar dispositivos a Internet sin importar el hardware del dispositivo.

Actualmente, en el mercado dispone de tres módulos:

- ✓ **Imp001:** Fue el primer módulo creado se compone de un procesador, Wifi y un antena. A diferencia de los demás módulos este incluye una ranura para un tarjeta SD.
- ✓ **Imp002:** Es una evolución del módulo anterior de un tamaño más reducido y se compone de un procesador, Wifi Soc y una antena
- ✓ **Imp003:** Este módulo es el que mayor rendimiento Wifi tiene y más sencillo es su integración con la plataforma Electronic Imp. A diferencia de los demás módulos este no incluye antena. Tiene un tamaño diez veces menor que su predecesor imp002.

Cada módulo esta optimizado para un consumo mínimo de energía, esto permite que puedan operar durante años con tan solo un par de baterías AA. Finalmente estos módulos corren en impOs y pueden ser ejecutados en cualquier hardware con la instalación de un agente.

- **Integración**

Electronic Imp ofrece una solución extremo a extremo haciendo posible que sea fácil conectar cualquier dispositivo a internet de una

forma muy sencilla y rápida, a través de una infraestructura, que nos permite esa integración pudiéndonos centrar en la aplicación en sí.

**ImpOs** es el core de la plataforma Electronic Imp que a través del dispositivo y su agente online provee los mecanismos necesarios para satisfacer nuestras necesidades a la hora de desarrollar cualquier aplicación. Dispone de una Api abierta la cual nos permite gestionar la conectividad, energía, seguridad, notificaciones, etc. El código de nuestra aplicación correrá sobre una máquina virtual que nos ofrece **impOs**

Cloud Es la pieza central de la plataforma **Electronic imp**, para conectar nuestros dispositivos a través de internet, se le identifica mediante un único agente online, para así poder tener libertad total a la hora de programar un dispositivo.

La plataforma Cloud nos ofrece un servicio **Saas**, nos ofrece almacenamiento de datos, visualización de reportes, es el agente quien gestiona este tipo de comunicaciones todo ello bajo el protocolo HTTPS.

### 2.3.2. Ámbitos de aplicación

La plataforma Electronic Imp, no se centra en un único campo de aplicación, por lo que diversifica un poco su oferta ofreciendo diversas soluciones en el mundo del IoT, sobre todo en los siguientes ámbitos:

- Smart Logistic
- Smart Home
- Smart Consumer
- Smart Industry

### 2.3.3. Conclusiones

*Electronic imp es sin duda una plataforma muy a tener en cuenta en el mundo del IoT, que nos ofrezca un ecosistema en el que se nos facilita tanto el software como el hardware siempre es un punto a favor.*

*Es una plataforma que se caracteriza por ser escalable, teniendo en cuenta que aparte de usar sus módulos y su agente podemos usar cualquier tipo de dispositivo. Además tiene a nuestra disposición diferentes herramientas para adaptar nuestros dispositivos a las necesidades cambiantes del proyecto. Por tanto ese es un punto muy a tener en cuenta a la hora de realizar cualquier proyecto, llegar a un punto del proyecto y ver que no es escalable puede tirar por tierra horas de trabajo.*

## 2.4. Spark

*Para empezar hablando de la plataforma Spark.io, tenemos que comenzar por el núcleo del proyecto que nació en la plataforma de crowdfunding kicstarter. Se trata de Spark Core, un dispositivo que permite que cualquier cosa tenga Wifi con el fin de poder comunicarnos con ella.*

*Spark Core se basa en Arduino por lo que es un punto a favor disponer una enorme comunidad y documentación a la hora de abordar cualquier proyecto.*

*La idea de Spark Core es crear una plataforma de desarrollo para dotar de inteligencia a objetos, es aquí donde arduino nos da todo ese poder y los mecanismos necesarios a la hora de programar.*

*Con la compra de este módulo, el cual tiene un precio de 39\$, viene incluido un servicio en la nube, Spark Cloud que nos permite desarrollar y actualizar nuestro dispositivo de forma remota. Este servicio es gratuito cuando adquirimos cualquier módulo hardware.*



### 2.4.1. Características Principales

Nos encontramos ante una plataforma tipo ecosistema, que auna hardware y software, por lo que nos ofrece todo lo necesario para poder desarrollar nuestra aplicación sin tener que salir de esta plataforma.

- **API**  
API basada en REST, Todas las peticiones de los dispositivos SPark, van a través de los servidores API usando TLS. Es una API bastante completa y puede ser consultada desde (<http://docs.spark.io/api/>).
- **Hardware**  
Como hemos comentado anteriormente, Spark nació con Spark Core, lógicamente han ido evolucionando y sacando nuevos productos hardware, aquí solo nos centraremos en el más popular y extendido Spark Core.

**The Spark Core:** Es el primer módulo de la compañía que permite dotar de Wifi a cualquier objeto, sus características principales son:

- ✓ Se puede conectar a una placa standard.

- ✓ Certificación FCC/CE/IC.
- ✓ STM32F103 ARM cortex M3
- ✓ CC3000 Wi-Fi module
- ✓ Hardware Open Source

- **Integración**

Toda integración se deberá hacer a través de la API que provee Spark Cloud. Con Spark Cloud podremos conectar nuestro hardware a la web. A través de un Gateway seguro conectaremos nuestro hardware y podremos interactuar con él mediante APIS y servicios web.

#### **2.4.2. Ámbitos de aplicación**

Quizás sea porque están comenzando en esto del IoT , en mi opinión es una plataforma que promete en el campo del Smart Home, aunque a día de hoy lo que más uso se le da para tema educativos o prototipos, no hay que perder de vista esta plataforma para temas de Smart Home.

#### **2.4.3. Conclusiones**

*Es una plataforma que abarca tanto software como hardware, ideal para iniciarse en el mundo del Internet de las cosas, debido a su facilidad de configuración, por lo que acorta los tiempos de aprendizaje en esta plataforma.*

*Dispone de una gran escalabilidad y una gran comunidad que crece cada día más.*

### **2.5. Blaulabs**

***Empresa Española con sede en Barcelona, es una plataforma SaaS modular dedicada al IoT, centrándose en la monitorización y análisis en tiempo real de los dispositivos y sensores.***

***Es una plataforma desde la cual podemos capturar cualquier tipo de datos desde cualquier dispositivo haciendo uso de la API abierta que se nos facilita. Otros de los puntos importantes, es su base de datos Time series capaz de gestionar altos volúmenes de datos en tiempo real.***

***Es una solución completamente ampliable y modular, es totalmente flexible permitiendo un gran número de configuraciones entre las que destacan dashboards con diferentes gráficos, informes, analíticas, etc.***





### 2.5.1. Características principales.

La plataforma Blaulabs es una plataforma modular, el conjunto de esos módulos se agrupan en torno a Blaulabs Suite cuya línea de negocio se basa en dar soluciones tipo Smart City, Gestión Energética, Smart Manufacturing, etc.

- **API**

A estas alturas poco nos debe sorprender que se use una API basada en REST. Desde la propia web nos facilita información de como utilizar esta API, se usa los lenguajes Python y Java.

Hace especial hincapié en como usar esta API a la hora de interactuar con TimeSeriesDB, se nos recomienda que usemos las librerías oficiales como la mejor forma de interactuar con esta base de datos.

- **Hardware**

Dado el negocio al que va orientado, el catálogo de hardware compatible es bastante limitado, hay un apartado de consultas en el que podemos preguntar por otro tipo de hardware pero no aporta ninguna información adicional acerca de otros hardware disponibles, los que admite a día de hoy son los siguientes:

- Satel OWA31IETH
- Satel SenNet DL160
- Satel SenNet DL161
- GridPoint
- GridAgent
- GridLink

### 2.5.2. Ámbitos de aplicación

Esta plataforma tiene una línea de negocio muy marcada, principalmente se centra en **energía, Smart city e infraestructuras**. . Por tanto, ya nos ofrece una solución para este tipo de negocios, principalmente se basan en la gestión eficiente de los datos recogidos por los sensores, entraremos a analizar las diferentes soluciones que nos aporta esta empresa:

- **BlauEnergy 4.0:** Orientada a la gestión de la energía, a través de la captación y almacenamiento de datos mediante sensores. La finalidad de esta solución es mejorar la eficiencia energética, reducir costes y reducir emisiones.
- **Smart City:** El módulo Smart city busca gestionar de forma eficiente los servicios que se prestan en la ciudad. Con este módulo se pretende dotar de inteligencia a toda esa infraestructura básica para poder desarrollar un modelo de ciudad más eficiente.
- **Infraestructuras inteligentes:** Solución orientada tanto empresas públicas como privadas, que ofrece una gestión más eficiente de los sistemas de transporte, redes eléctricas (Smart Grid), agua y telecomunicaciones.

### 2.5.3. Conclusiones

*Plataforma con una línea de negocio muy marcada, esto puede ser un impedimento pero a la hora de extender el mercado pero a la vez un punto a favor debido a su especialización.  
Recomendable sobre todo para proyectos de ciudades inteligentes y gestión de la energía.*

## 2.6. Thinking Things

*Estamos ante una plataforma del internet de las cosas, que no solo es una plataforma software sino que nos ofrece tanto hardware como software para empezar a trabajar.  
Thinking Things es la apuesta de Telefónica para el internet de las cosas, es una plataforma muy nueva, apenas lleva unos meses y todavía están en muchos aspectos en fase beta. Me parece adecuado comentar esta plataforma, por diversos aspectos, el más importante es la sencillez de su hardware a la hora de montarse prácticamente como si fueran piezas de lego, el segundo aspecto es la conectividad obviamente aprovechándose de su experiencia en las telecomunicaciones se olvidan de la conectividad WiFi y apuestan por la red GSM, es una gran ventaja ya que tendremos una conectividad global.*



### 2.6.1. Características principales

Estamos ante una plataforma un tanto peculiar, se basa en unos módulos tipo LEGO, compuestos básicamente de una batería, un sensor y un módulo de conectividad GSM.

Por supuesto, nos ofrece una plataforma software con una interfaz muy fácil de usar. La conectividad es otra novedad respecto a otras plataformas, a través de la red móvil y con una SIM M2M de telefónica se obtiene conectividad casi en cualquier parte del mundo.

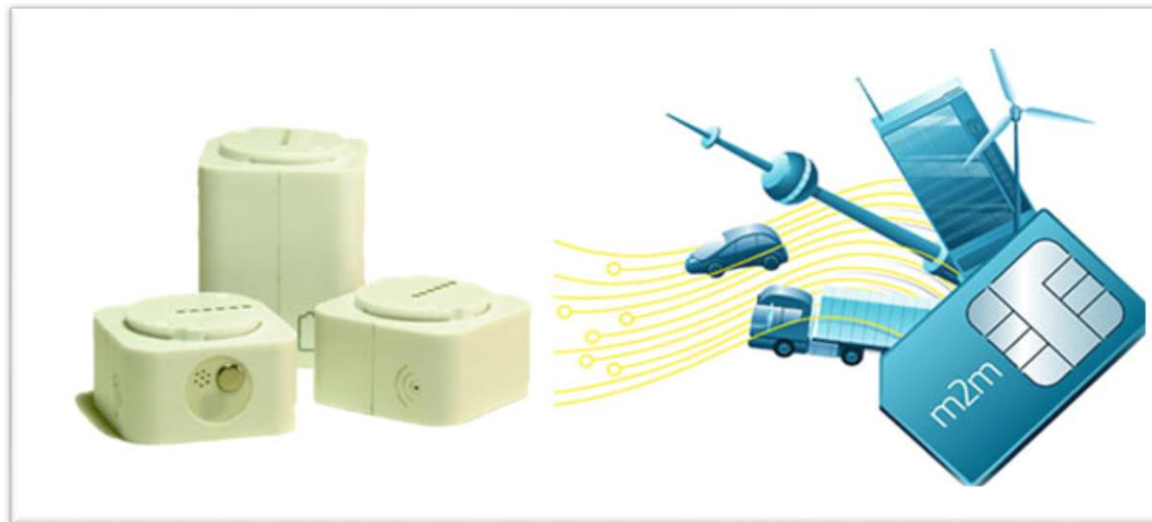
- **API**

Se trata de una API basada en REST, con la cual podremos conectar nuestro stack a las necesidades de nuestra empresa, o integrarlo con nuestro software

- **Hardware**

Sin duda es la base de esta plataforma, por originalidad, sencillez y facilidad de uso creo que es merecido que se comente esta nueva forma de interconectar objetos.

Se trata de diferentes cubos de plásticos que se conectan como si fueran piezas de Lego.



El stack se compone de diversos módulos, en la base está la batería, la cual puede recargarse mediante un microUSB, la duración depende del uso que se le dé, según comentan en la web puede alimentar un stack durante un mes conectándose cada hora. Se puede conectar más de un módulo de batería para alargar la vida.

Justo encima del módulo de batería se encuentra conectado el módulo de conectividad, este módulo es el core, envía los datos de todos los módulos a la

página web. Como comentamos al principio usa la red móvil y trae una tarjeta SIM.

Y justo encima del módulo Conectividad se encuentran conectados los módulos “sensores”. Actualmente solo tienen a la venta el módulo Ambiente el cual mide la temperatura del aire, la humedad del aire y la luz ambiente. Próximamente sacarán a la venta otros como el sensor de presencia.

### 2.6.2. Ámbitos de aplicación

Es una plataforma que debido a que es todavía muy nueva su uso es meramente educativo, y muy buena opción para introducirse en el mundo del IoT creando algún prototipo de tipo Smart HOME.

### 2.6.3. Conclusiones

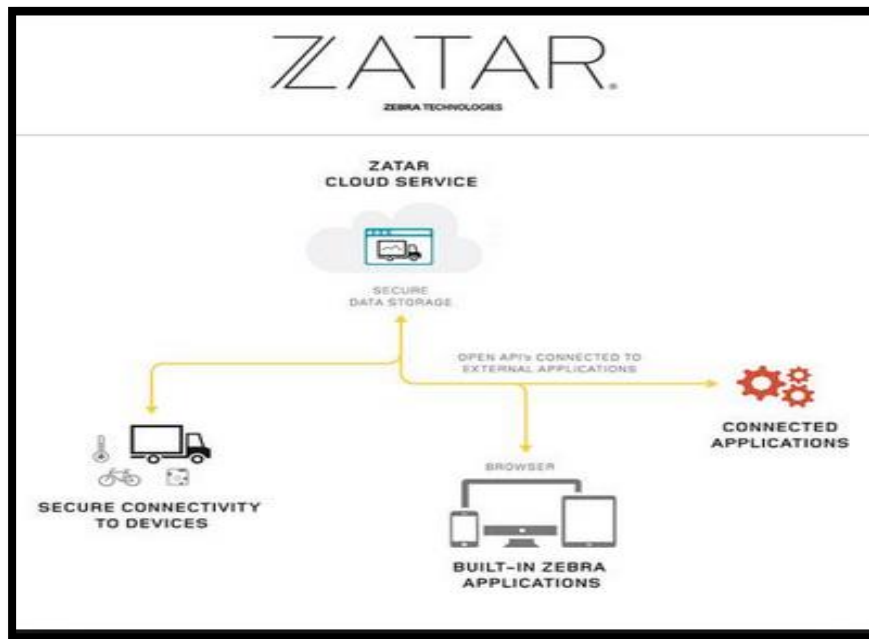
*Al ser una plataforma tan nueva, y que tan solo tiene un sensor lanzado al mercado el ámbito de aplicación es bastante limitado, pero es una buena opción para iniciarse en este mundo por su sencillez y un precio asequible unos 99€.. Aún así es una plataforma a tener en cuenta, sobre todo por un próximo lanzamiento **Thinking Things Open**, que es una placa industrial compatible con Arduino, que promete muchas posibilidades.*

*Su configuración sencilla, y el hecho de no tener que depender de una conexión a Internet ya que trae su propia conectividad, la hacen una plataforma muy atractiva para iniciarse, y muy a tener en cuenta en un futuro no muy lejano.*

## 2.7. Zatar

***Zatar es una plataforma IoT orientada para construir aplicaciones empresariales, basado en cloud computing. Esta solución incluye componentes básicos de IoT como son los sensores, RFID y código de barras para que cualquier empresa pueda tener la información que necesite y gestionarla a través de la web.***

***Como viene siendo habitual en este tipo de plataformas, usa una API basada en REST para acceder a todas las propiedades y métodos de los dispositivos.***



### 2.7.1. Características principales

A pesar que no trae novedades diferenciadoras respecto a las demás plataformas estudiadas, cabe destacar su marcado perfil empresarial y sobre todo los dispositivos a los que va orientado principalmente y como veremos más adelante a impresoras. Entre las características a destacar, destacaremos las siguientes:

- **Accesibilidad:** Accesible desde cualquier lugar y desde cualquier smartphone, ordenador o Tablet. Al estar basado en la nube permite a las empresas gestionar un gran número de sensores ambientales, impresoras, lectores RFID, balizas, etc.
- **Control del dispositivo:** Personalización de cada dispositivo, también se nos permite comunicarnos e interactuar remotamente siendo la compartición de datos con otros dispositivos de forma securizada . Por supuesto, también se nos permite acceder a los datos en tiempo real y una última novedad que no habíamos visto en otras plataformas es la posibilidad de crear **grupos de dispositivos** para acceder de ofmra colaborativa.
- **Escalable:** La plataforma Zatar nos permite trabajar con una gran número de dispositivos, trabajando con las principales estándares de APIs como por ejemplo las basadas en REST, CoAP y JSON.
- **Seguridad y Privacidad:** Los protocolos principales de comunicación son HTTPS y SSL. Los intercambios van encriptados mediante RSA.
- **Asequible de usar:** Bajo coste .

- **Hardware**

Los dispositivos y sensores, aportan diferentes datos que pueden ser de gran utilidad para las empresas, incluso estos sensores y dispositivos pueden compartir datos entre ellos lo que aumentan las opciones de mejoras de las empresas. Zatar se caracteriza por el uso de RFID readers(Radio Frequency IDentification), provee soporte para un gran número de impresoras Zebra y también da soporte para Raspberry Pi.

Hace uso del Zatar Gateway para conectarse a internet. Este actúa como si fuera un modem del IoT. A través de este Gateway podemos conectarnos a internet via WIFI, Ethernet o 3G.

- **Integración**

A pesar que Zatar no es una plataforma Open source, si que esta abierta a los desarrolladores para que creen aplicaciones para la plataforma y así puedan extender la funcionalidad de la misma más allá de la funcionalidad de la web.

### **2.7.2. Ámbito de aplicación**

Estamos ante una plataforma realmente orientada a logística, lo que es un punto a favor para ciertas empresas debido a su especialización.

### **2.7.3. Conclusiones**

*Es una plataforma muy interesante para el sector de Logística debido a su alta especialización en esta área. Dándole una gran versatilidad y escalabilidad en esta área.*

*Es de las pocas plataformas que usan la tecnología RFID, tiene gran integración mediante aplicaciones gracias a la apertura de su API para que otros desarrolladores trabajen sobre ella.*

## 2.8. Comparativas Plataformas Softwares

Plataforma	HW	Ámbito	Ventajas	Desventajas
Thingspeak	Arduino, Spark, Raspberry Pi, Electronic Imp	Smart Home Prototipos	Interfaz. App. Integración redes sociales	Documentación limitada a cierto HW.
Carriots	Arduino Raspberry Electronic Imp Beagle TST Industrial	City Energy Oil Agiculture Buildings Retail Banking Consumer Loguistic	Integración redes sociales Hw compatible Ámbitos de aplicación.	Poca Documentación
Spark	Spark	Smart Home Prototipos	Ideal para iniciarse. Escalable Comunidad al alza.	Compatibilidad HW.
Electronic Imp	Electronic Imp. Otros	Loguistic Consumer Home Industry	HW. Ámbitos de aplicación Ecosistema propio. Escalable.	No recomendable para iniciarse.
Bluelabs	SATEL Grid	City Energy Manufacturing	Especialistas en Smart City. Aplicaciones orientadas a Smart City.	HW limitado.
Thinking Things	Propietario	Smart Home Prototipos	Fácil Configuración. Integración fácil entre HW- plataforma Ideal para prototipos. Ideal para iniciarse	Poca variedad Sensores. Sol es posible usar su HW.

Zatar	RFID	Logística	Integración con impresoras. Escalable. Precio.	Software propietario.
-------	------	-----------	--	-----------------------

### 3. Campos de aplicación en IoT

*Una vez que se han presentado diferentes plataformas software, es momento que hablemos de los principales ámbitos de aplicación del Internet de las cosas.*

*El IoT ha evolucionado tanto en los últimos años que prácticamente a día de hoy abarca cualquier campo de la vida que podamos imaginar, sin embargo, aquí describiremos los que mayor evolución y protagonismo están alcanzando estos últimos años.*

#### 3.1. Smart Home

El Smart Home o Casa inteligentes es sin duda uno de los campos de aplicación, en el que el internet de las cosas más ha apostado en los últimos años y en el que más proyectos se llevan a cabo por empresas y particulares. Las aplicaciones en esta área son muy variadas desde control de temperatura de la vivienda, poder subir y bajar persianas remotamente, pasando por frigoríficos inteligentes que nos avisan cuando se nos acaba algún alimento incluso puede hacer un pedido por nosotros a la tienda si así lo deseamos. Muchas son las empresas que están apostando por Smart Home, por ejemplo la empresa guipuzcoana Wattio, ofrece una solución final de Smart Home, lista para el usuario final la conecte en su hogar.

Pero por supuesto el rey en este tipo de tecnología es sin duda Zigbee, con una amplia experiencia y con una red de sensores muy utilizada en este ámbito.



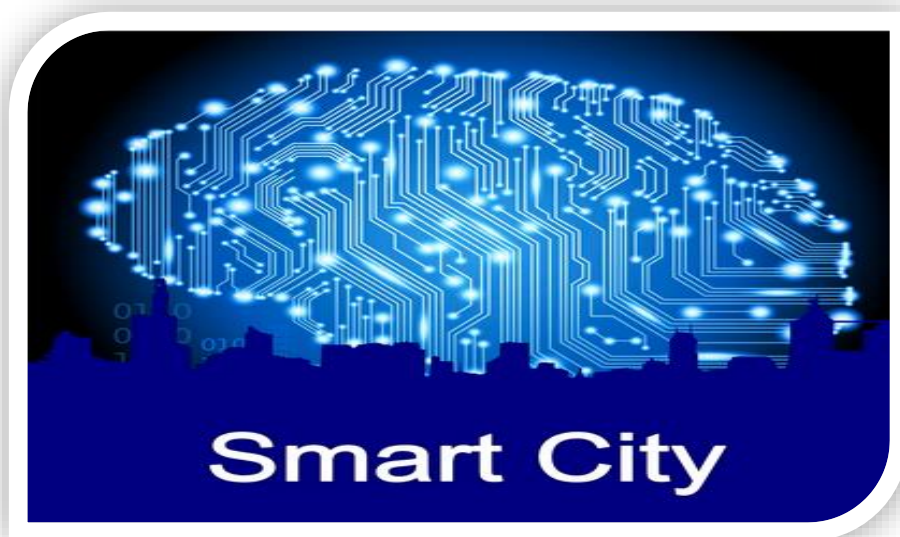


### 3.2. Smart City

Las ciudades inteligentes es otro campo por el que empresas e instituciones están apostando últimamente. El concepto es muy simple, tener la mayor cantidad de datos para facilitar la vida al ciudadano.

Sus aplicaciones van desde el Smart Parking, indicando al conductor donde hay un hueco libre para poder aparcar, el Smart Traffic indicando en tiempo real el tráfico en la ciudad y rutas alternativas, pasando por una gestión eficiente del alumbrado de la ciudad.

Esto son solo unos ejemplos de lo que se pretende implementar con esta tecnología, su futuro es muy prometedor sobre todo viendo como muchas ciudades y empresas están apostando claramente por convertir las ciudades en un ecosistema más inteligente y conectado. Muchas empresas están especializándose en esta área, de las que hemos comentado anteriormente destacan Blaulabs y Carriots , son dos empresas que se el Smart City es un área en el que se esetan centrando.



### 3.3. Agricultura Inteligente

A pesar de que le esta costando arrancar en este medio, probablemente por ser un área en el que la evolución tecnológica nunca ha sido una prioridad, si que se está intentando revolucionar la forma en la que trabajan los agricultores. Por ejemplo tenemos una empresa Española Libelium, que comercializa las placas Wasp mote que setan apostando por la Agricultura inteligente.

Sus aplicaciones van desde monitorizar los cultivos, herramientas de soporte para la toma de decisiones, controlar automáticamente riego, protección de heladas, fertilización..

La agricultura inteligente se convertirá en el campo de aplicación más importante en los países predominantemente agrícola.

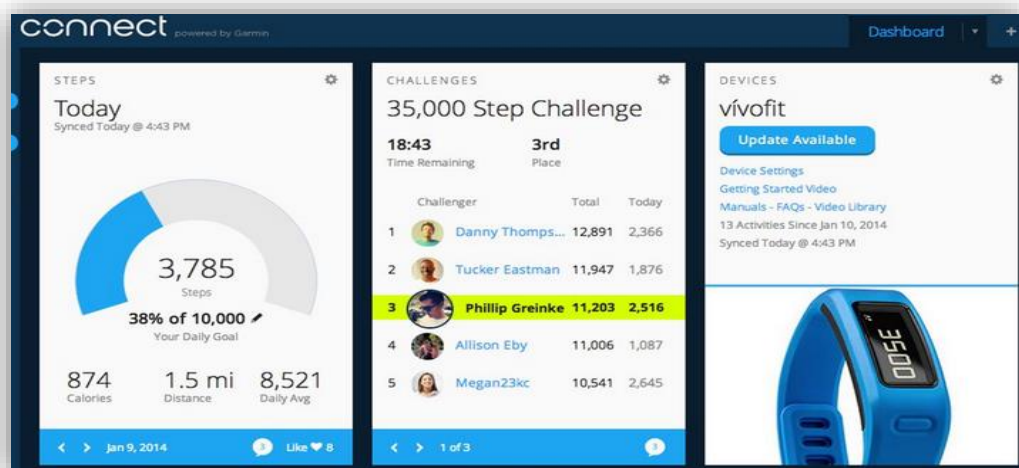


### 3.4. Wearables

Son dispositivos de consumo, en la actualidad están tomando una gran popularidad, debido al crecimiento del sector.

Encontramos pulseras que nos monitorizan durante todo el día y nos dicen los pasos que hemos andado y las calorías consumidas, relojes inteligentes que se conectan con nuestro Smartphone, en el ámbito deportivo también hay un mercado al alza desde relojes que monitorizan nuestro pulso, hasta nuestra cadencia a la hora de correr..

Empresas como Garmin con su producto Vivofit que es una pulsera que monitoriza los pasos que damos al cabo del día y a través de una web podemos visualizar una gran cantidad de datos como distancia recorrida, calorías quemadas..



## 4. Análisis Plataformas Hardware en IoT

*Tras analizar plataformas software y su ámbito de aplicación, es el momento que hablemos que hardware vamos a elegir a la hora de aventurarnos en cualquier proyecto. Para ello debemos tener claro, cual será la finalidad del proyecto, si es para ámbito educativo o un prototipo entonces la duración de la batería puede que no nos preocupe, si por el contrario vamos hacia una solución real aplicado a un problema concreto, ese será un punto crítico, a parte de otros muchos como tamaño, escalabilidad, etc.. Es por eso que aquí intentaremos dar una visión de los diferentes tipos de hardware que podemos usar en IoT y el uso y finalidad de cada uno de ellos.*

### 4.1. Arduino

Arduino es una plataforma de hardware libre, sin duda la más popular del mercado con una amplia comunidad que hace de esta solución la más apropiada para iniciarse en el mundo del hardware libre.

Se basa en una placa con un microcontrolador, con entradas y salidas analógicas y digitales, además posee un entorno de desarrollo propio, con el fin de facilitar el desarrollo de proyectos.

La placa de Arduino se caracteriza por el uso de un microcontrolador Atmel AVR, siendo el Atmel 328 uno de los más utilizados, este por ejemplo es el que usa el Arduino UNO rev3, una de las placas más utilizadas por la comunidad open-hardware.

Otra característica fundamental a la hora de adquirir una placa Arduino es su bajo coste y su puesta en funcionamiento. En el mercado encontramos kits de iniciación con una gran documentación que facilita al usuario principiante, iniciarse de una forma fácil en este mundo. Esto es clave porque el usuario obtiene resultados prácticamente nada más conectar la placa, lo que provoca que se siga investigando a la vez que se van adquiriendo grandes conocimientos, por lo que la curva de aprendizaje es claramente al alza.

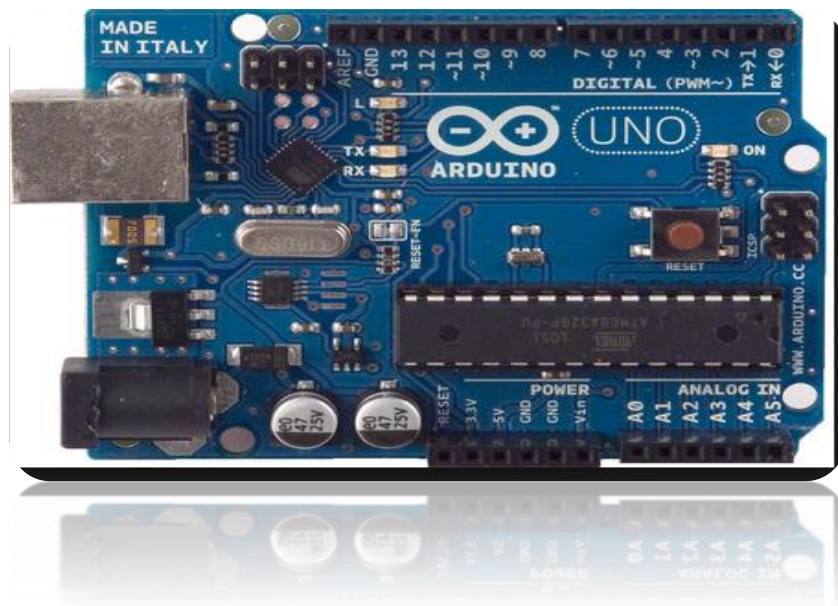
Arduino es multiplataforma, es decir, admite los sistemas operativos más populares que hay en la actualidad, tales como Windows, Mac Os X y Linux. Sin duda una gran ventaja ya que a la hora de elegir una placa hardware, el sistema operativo no va a ser una limitación.

A la hora de adquirir una placa Arduino, puede surgirnos la duda de cuál elegir dado que en el mercado encontramos una gran oferta de modelos con diferentes características, sin duda, en un usuario que se está iniciando se recomienda que tengamos en cuenta el número de entradas y salidas, especialmente prestando mayor atención a las analógicas, que son las que frecuentemente limitan nuestro proyecto. A continuación se muestra un cuadro comparativo con las placas arduino oficiales más populares del mercado:

Modelo	I/O digitales	Entradas analógicas	Salidas PWR	UART	Memoria	Precio
Uno r3	16	6	6	1	32kb	20€
Mini 05	14	6	8	1	32kb	17€
Leonardo	20	12	7	1	32kb	23€
Mega r3	14	6	7	1	256kb	18€
YUN	20	12	7	1	32kb	70€

**Nota :** Precios orientativos, ya que en internet puedes encontrar una gran variedad de precios.

La principal novedad que trae la placa YUN respecto con las demás placas antes comentadas, es que esta placa ya trae integrada conexión Wifi y Ethernet por lo que no es necesario adquirir los Shields para dotar de esta funcionalidad a las demás placas.



### Conclusiones:

*Arduino es sin duda una perfecta elección para todo aquel que se quiere iniciar en el mundo de los microcontroladores, su bajo coste, su amplia comunidad y su fácil acceso a la plataforma la hacen ser la mejor opción para iniciarse en este mundo. Esto no quiere decir que no se puedan hacer proyectos de calidad, ni mucho menos, solo hace falta darse una vuelta por la web para poder comprobar como con estas pequeñas placas se logran proyectos de una alta calidad.*

*Es ideal también para realizar prototipos en el mundo del IoT la gran mayoría de las plataformas IoT traen compatibilidad con estas placas hardware por lo que es un candidato muy serio a la hora de aventurarnos en el internet de las cosas.*

***Es por ello que se ha elegido para nuestra aplicación, siendo un candidato perfecto para demostrar mediante un prototipo los usos y posibilidades que tienen dentro del Internet de las cosas.***

## 4.2. Wasmote

Creada por la empresa española Libelium, estamos ante una plataforma modular open source, cuya finalidad es la de construir redes inalámbricas de bajo consumo.

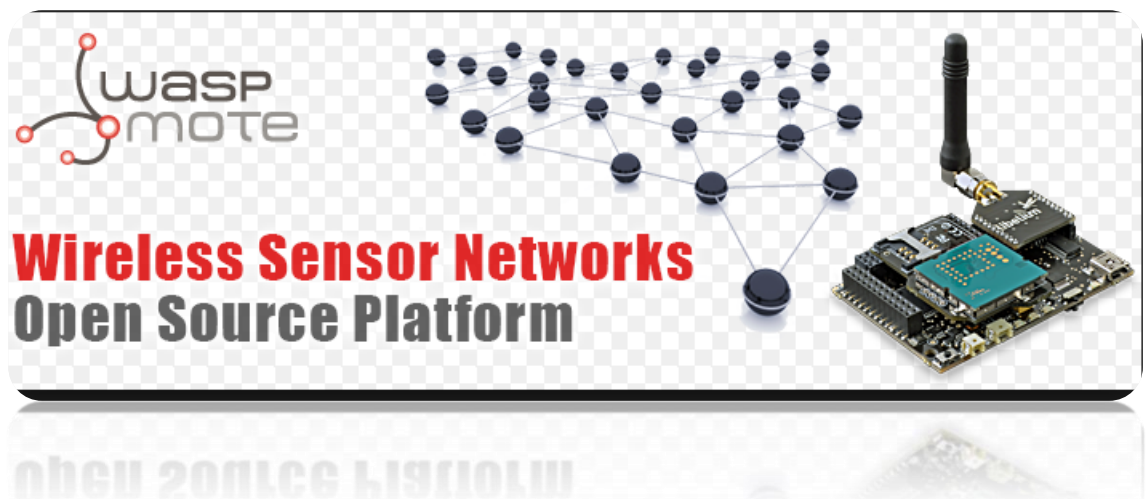
Prácticamente la plataforma se compone de la placa Wasmote con un microcontrolador de la familia Atmel, memoria, batería, acelerómetro y sockets para ir añadiendo módulos.

A nivel software API y compilador open source, usa el protocolo de comunicación Zigbee con alcances de hasta 40 Km, dispone de diferentes módulos para dotarles de diferentes medios de comunicación, tales como, Bluetooth, GPS, y GPRS.

La placa Wasmote usa el mismo IDE que Arduino, por lo que el mismo código es prácticamente compatible en ambas plataformas solo variando algunas líneas de código. Como podemos intuir, Libelium fabrica estos módulos basándose en estándares de hardware abierto, aunque intenta meter un cariz diferenciador dándole un toque de mayor robustez, la facilidad para incorporar diferentes sensores y la posibilidad de operar a largas distancias.

A diferencia de Arduino, cuya plataforma esta más orientada a la creación de proyectos “caseros” o prototipos, Wasmote es un dispositivo especialmente orientado para la creación de Redes Sensoriales Inalámbricas que necesitan de una larga duración, cuya finalidad es desplegarse en escenarios reales.

Los ámbitos de aplicación de estos módulos son para Smart City, por ejemplo se han aplicado ya en proyectos de Smart Parking, monitorizar la red de alcantarillado en Valencia,, para medir la calidad del aire en Santander. También se aplica en el mundo agrícola para medir la irrigación, el crecimiento de las cosechas, además detectar sucesos meteorológicos adversos.





### Conclusiones:

*Estamos ante una plataforma hardware destinada a proyectos reales, por lo tanto es destinada a un ámbito profesional por lo que a diferencia de Arduino no es una plataforma para iniciarse. Le avalan que es una plataforma que ya se ha aplicado a con éxito en proyectos reales, por otro lado su gran variedad de sensores es un punto a favor. También compartir el mismo IDE que Arduino es mi opinión es un punto a favor, poder realizar prototipos en arduino y poder migrar con pocas modificaciones a Waspnote puede ser un punto interesante a tener en cuenta. También el no tener que aprender un nuevo IDE para gente que viene de la plataforma Arduino, hace que el paso de una plataforma a otra se pueda realizar de una forma sencilla.*

### 4.3. Spark

Volvemos a mencionar a Spark, anteriormente lo conocimos por su plataforma software, ahora hablaremos de su hardware que es la parte más importante de esa plataforma y en la que se basa todo su núcleo.

Es una placa de un tamaño reducido que se compone de un módulo WiFi, cuya finalidad es la de dotar de conexión a Internet prácticamente a cualquier cosa. Es una empresa que nació gracias a la financiación de la popular plataforma Kickstarter.

Se caracteriza por ser un dispositivo fácil de instalar y que se puede programar sin cables, gracias a su conexión WiFi. Combina la simplicidad de un Arduino con todo el poder del chip ARM cortex M3, considerado uno de los mejores módulos disponibles en la actualidad.

Dado que Spark Core es un dispositivo que se caracteriza por estar siempre conectado a Internet, es una gran ventaja, se nos facilita también una vez adquirido el producto una nube de la compañía Spark, desde la cual podemos realizar desarrollos, actualizar y otras ventajas a nivel de software.

Es una plataforma totalmente Open Source, todos los diseños de firmware y hardware son de código abierto, lo que permite una integración libre en diversos proyectos. Hay que destacar que Spark Core usa estándares abiertos como HTTP, TCP o TLS/SSL.

Si queremos ampliar la funcionalidad existe la posibilidad de ampliarlo con los SHIELD o escudos. Incluso el propio Spark Core comercializa un Shield que permite añadir cualquier escudo que encontremos en el mercado, sobretodo esta orienta a ser utilizados con Arduino.



#### Conclusiones:

*Estamos ante un dispositivo para ser una gran herramienta de aprendizaje en el mundo de los microcontroladores y en la internet de las cosas, dada su facilidad de configuración, apoyo de una comunidad que cada día va creciendo más y también a su precio que ronda los 30€.*

*Su uso es muy similar a lo que nos propone Arduino, esta más orientado a ser una plataforma en la que poder desarrollar prototipos y en la que tengamos nuestro primer contacto con la internet de las cosas, aunque como todo dependerá de nuestro nivel de conocimientos para poder desarrollar proyectos más o menos profesionales.*

#### 4.4. Intel Galileo

Se trata de una nueva familia de placas compatibles con Arduino que trabajan bajo la arquitectura Intel. ES uno de los primeros productos de Intel para introducirse en el mercado del IoT, trae funciones muy similares a Raspberry Pi o Arduino.

Galileo es un microcontrolador basado en el procesador iQuark SoC X1000 que funciona a 400MHZ, es realmente una CPU de 32 bits tiene una arquitectura similar a la de los antiguos Pentium. Es la primera placa basada en arquitectura Intel y cuyo diseño está pensado para que sea compatible con Arduino. Trae compatibilidad con los Shield que usa Arduino UNO rev3 y se puede usar el IDE de Arduino. Es multiplataforma por lo que se puede usar tanto en Linux, Mac y Windows.

Esta placa ejecuta un sistema operativo Linux libre, que contiene las librerías de software de Arduino, permitiendo una mayor escalabilidad y reutilizar el software ya existente.

Además de esa compatibilidad con Arduino, Galileo tiene una serie de puertos y características que son estándares en la industria del PC por lo que va un paso más allá de los Shields de Arduino. Destacan por ejemplo, puerto Ethernet a 100Mb, espacio Micro SD, memoria flash a 8Mb, un puerto RS-232 serial.

Destacar que gracias a su gran conectividad, potencia de proceso y el uso de SDK sencillos permite desarrollar software capaz de conectar cualquier dispositivo a Internet, muy interesante es su uso para domótica proyectos que necesiten de un PC de bajo consumo y coste.



### Conclusiones:

*Estamos ante una placa muy similar a la Arduino UNO rev3, ideal para acceder al mundo de los microcontroladores y destinado a estudiantes o personas que se inician en este mundo. Su precio es algo menor de 70 euros, algo caro respecto a las muchas otras opciones que encontramos en Arduino, aunque también es más potente. Su campo de uso es similar al de Arduino, proyectos docentes, prototipos o gente que se inciaa en el mundo de los microcontroladores.*

*Sinceramente no la recomendaría para alguien que esta iniciándose, hay otras opciones en el mercado con similares funcionalidades, y sobre todo con una amplia comunidad y documentación, fundamental cuando uno da sus primeros pasos en este mundo.*



#### 4.5. Zigbee

Zigbee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación para su comunicación con radiodifusión digital de bajo consumo.

Basado en el estándar IEEE 802.15.4.

El siguiente cuadro nos dará una idea general de porque Zigbee está orientado sobre todo a domótica:

Estándar	Ancho de Banda	Consumo de potencia	Ventajas	Aplicaciones
Wi-Fi	Hasta 54Mbps	400ma transmitiendo, 20ma en reposo	Gran ancho de banda	Navegar por Internet, redes de ordenadores, transferencia de ficheros
Bluetooth	1 Mbps	40ma transmitiendo, 0.2ma en reposo	Interoperatividad, sustituto del cable	Wireless USB, móviles, informática casera
ZigBee	250 kbps	30ma transmitiendo, 3ma en reposo	Batería de larga duración, bajo coste	Control remoto, productos dependientes de la batería, sensores, juguetería

Con esta estandarización se ofrece a los fabricantes, integradores, desarrolladores.. la opción de poder trabajar bajo un enfoque basados en estándares a la hora de introducir nuevos productos dedicados a la domótica o automatización del hogar.

Zigbee se convierte en uno de los principales protagonistas en el mundo de la domótica, debido a su bajo consumo, su sistema de comunicaciones vía radio y su integración sencilla que permite que no se use mucha electrónica para la fabricación de sus nodos. El estándar ZigBee proporcionará la red de comunicaciones, seguridad con algoritmos empotrados y servicios de apoyo para la capa de aplicación.

Se definen tres tipos de dispositivos ZigBee según su papel en la red:

- **Coordinador ZigBee (ZC):** Es el dispositivo más completo debe haber uno por red, se encarga de controlar la red y los caminos que deben seguir para conectarse entre ellos, requiere memoria y capacidad de computo.
- **Router ZigBee(ZR):** Interconecta dispositivos separados en topología de red.
- **Dispositivo final ZigBee (ZED):** Posee la funcionalidad necesaria para comunicarse con su nodo padre (ZC o ZR), pero no puede transmitir información destinada a otros dispositivos. Con esto se consigue que este tipo de nodo pueda estar dormido la mayor parte del tiempo. Este tipo de dispositivo tiene unos requerimientos mínimos de memoria por lo que su coste se reduce significativamente.

Zigbee tiene un programa de certificación, que asegura que los productos son interoperables y de calidad, actualmente tienen más de 1000 productos certificados. Uno de los campos en los que más investiga ZigBee es en el Smart Home, cumpliendo con una amplia gama de estándares con el fin de que los dispositivos funcionen de una manera conjunta de forma inteligente.



**Conclusiones:**

*ZigBee es una tecnología ampliamente usada en entornos profesionales y también de aficionados. Este tipo de plataforma hardware lo clasificamos dentro del estándar radio. Destacaría sobre todo que su ámbito de aplicación es el mundo profesional, no recomendaría iniciarse con este tipo de tecnología debido a que no están sencillo como por ejemplo sea Arduino, también dadas sus características su uso esta más extendido a proyectos como SMART HOME, redes de sensores.*

## 4.6. Comparativa de Plataformas Hardware

Plataforma	Ámbito de aplicación	Ventajas	Desventajas	Micro	Memoria	Alimentación
Arduino	Prototipos, educación	Comunidad. Extensa Documentación Precio. Variedad dispositivos. Ideal para iniciarse.	En proyectos más profesionales quizás no sea la plataforma adecuada.	Familia ATmel: ATmega328	Modelos de 32Kb y otros como Mega de 256Kb.	<b>5V-9V</b>  <b>Consumo según modelo</b>
Waspote	Smart City, Smart agricultura	Amplia variedad de sensores. Bajo consumo. Compatibilidad con IDE Arduino.	Diseñado para unos requerimientos específicos. No apta para gente que se esta iniciando.	ATMega 1281	128KB	<b>3,3V -4,2V</b>  <b>ON:</b> 15mA <b>Sleep:</b> 55uA <b>Hiberna:</b> 0,7ua
Spark	Smart Home, Educación, Prototipos	Comunidad al alza. Facilidad de configuración. Precio. Ideal para iniciarse.	Poca variedad de dispositivos propios aunque permite compatibilidad con otros fabricantes.	ARM-32-Cortex	128KB	<b>3V-6V</b>  <b>Max:</b> 300mA <b>Min:</b> 50mA
Intel Galileo	Prototipos, Educación	Arquitectura Intel. Compatible con Arduino. Potencia de cómputo. Buena opción para iniciarse.	Precio. Documentación excasa. Su ámbito de aplicación y su competencia hacen que otros fabricantes tomen ventaja.	Intel Quark.S0 Cx1000	512KB	7V-15V  <b>MAX:</b> 800mA
ZigBee	Smart Home	Protocolo de comunicación. Bajo consumo. Variedad de sensores y dispositivos.	No recomendable para iniciarse.	Según fabricante	Según fabricante	Según fabricante

## 5. Aplicación IoT usando Arduino + Thingspeak.

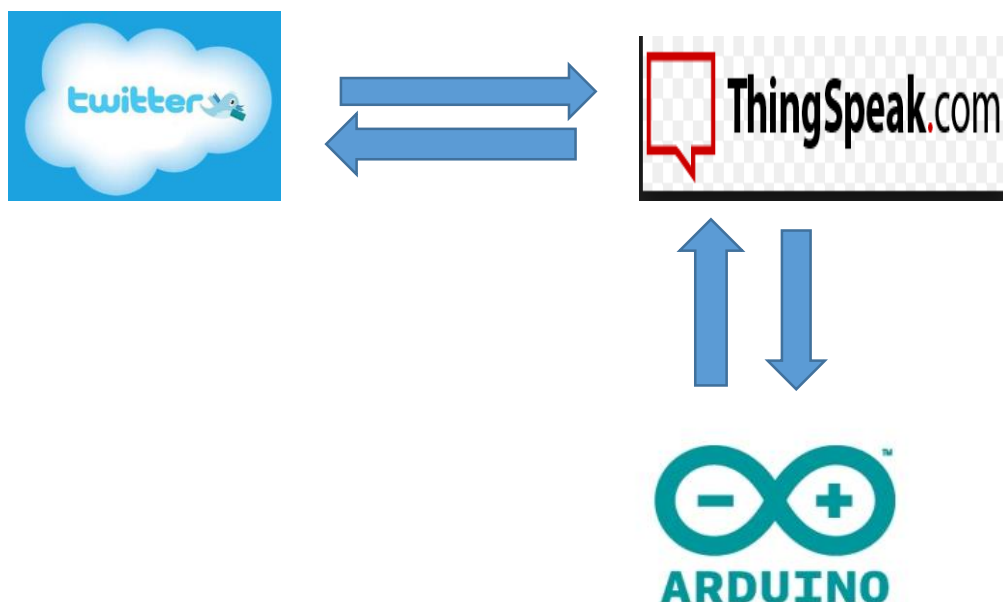
*En esta sección se intentará dar un enfoque práctico de todo lo anteriormente comentado, haciendo uso de una placa Arduino UNO rev3 al cuál, le añadimos el Shield WiFi Shield de la compañía Arduino, para dotarle de comunicación Wifi.*

*Haremos uso de la plataforma ThingSpeak, esta aplicación se basará en recrear un prototipo, basándonos en el área del Smart Home, midiendo la temperatura de una habitación y subiendo los datos medidos a la plataforma Thingspeak.*

*Además lo integraremos con la red social Twitter, mandando tweets a la red social por cada medida de temperatura. Finalmente activaremos un ventilador, recreando lo que sería un sistema de refrigeración, para disminuir la temperatura de nuestra habitación.*

*Con esta aplicación queremos sumergirnos en el mundo del SMART HOME, dando una solución real, que es la monitorización de temperatura en una vivienda. En este caso, monitorizaremos una habitación.*

*Al ser una solución real, esta podría ser comercializada, como parte de un proyecto global de SMART HOME, o simplemente vendiéndola por módulos como es el caso de activar un sistema de refrigeración o en invierno de calefacción. Si fuera un proyecto global, por ejemplo, podríamos bajar las persianas o desplegar un toldo si la temperatura vemos que asciende, las posibilidades son muy variadas y el mercado esta en pleno auge.*



## 5.1. Elección Plataformas

A continuación, se detallarán los motivos del porque se ha elegido las plataformas tanto software como hardware para el desarrollo de nuestro prototipo.

El principal motivo para la elección de la plataforma software ThingSpeak, es sin duda su categoría de Open Source, para este tipo de proyecto es sin duda uno de los mejores candidatos tanto a nivel económico como a nivel de documentación. Otro punto a favor es su interfaz amigable, y de fácil configuración, como además su integración con aplicaciones de terceros véase por ejemplo la red social twitter.

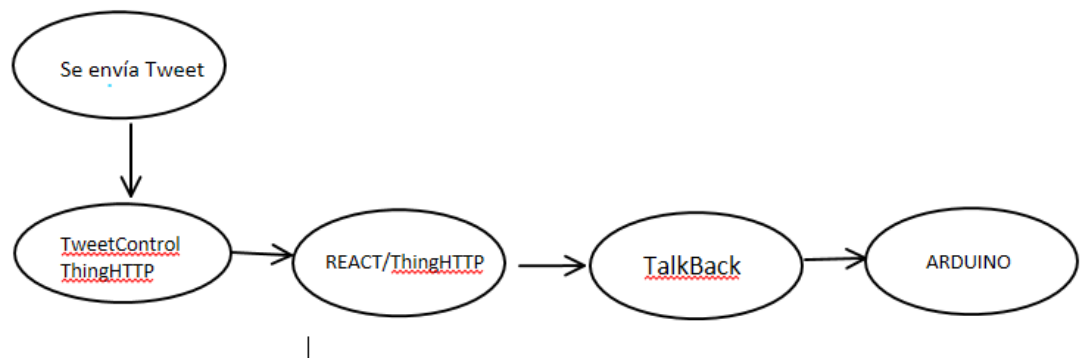
A la hora de elegir que hardware usar, la elección ha sido más sencilla, sin duda Arduino. Plataforma con un gran recorrido y una documentación muy extensa, y una gran comunidad que ayudan a todo aquel que tiene cualquier problema. Su precio también es un punto a favor, de lo más económico que hay en el mercado.

## 5.2. Funcionalidad

Nuestra aplicación realizará las siguientes funciones:

- Medir la temperatura ambiente de una habitación mediante un sensor de temperatura.
- Subir los datos a la plataforma lot Thingspeak.
- Integración con la red social Twitter.
- Mandar Tweet desde la red social Twitter a la plataforma Thingspeak
- Una vez llegue el tweet a la plataforma mediante el uso de las aplicaciones ThingHttp, TweetControl , React y Talkbacks activarán nuestro ventilador.
- Aunque con TweetControl y ThingHttp hubiera sido suficiente para activar el ventilador, se han usado las demás aplicaciones para mostrar las funcionalidades que trae esta plataforma.

### 5.2.1. Flujo Llegada de un Tweet



### 5.3. Plataforma Hardware

Al disponer previamente de una placa arduino uno rev3, hemos aprovechado dicha placa y hemos adquirido un WiFi Shield oficial para dotarle de comunicación Wifi y así poder comunicarse con la plataforma Software Thingspeak. De no haber tenido la placa Arduino UNO rev 3 se hubiera optado por la placa Arduino YUN ya que su precio es más económico que adquirir todo lo anteriormente descrito.

#### 5.3.1. Instalación IDE Arduino.

Para poder comenzar a trabajar con nuestra placa Arduino, lo primero que tenemos que hacer es descargarnos el IDE de Arduino, antes de continuar, hay que aclarar que la compañía de Arduino está dividida en dos, arduino.cc y arduino.org, a la hora de tener que descargar el IDE debemos tener esto en cuenta, aunque ambos IDE funcionan perfectamente para ambas placas. En nuestro caso descargamos el software de la página arduino.org, siendo la versión del IDE actual la 1.7.3. <http://www.arduino.org/downloads> La instalación es guiada por lo tanto es muy sencilla, una vez instalado conectamos nuestra placa por USB y esperamos a que sea reconocida por nuestro equipo.

#### 5.3.2. Conexión WiFi Shield.

Antes de conectar nuestro Shield debemos actualizar el mismo a la última versión software, según el sistema operativo que usemos la operación varía de uno a otro. En nuestro caso hemos usado el sistema operativo Windows 8.1.

Lo primero que debemos hacer es descargar el software FLIP de Atmel (<http://www.atmel.com/tools/FLIP.aspx>), una vez instalado pondremos el WiFi Shield en modo programación, simplemente será cerrar el jumper que viene por defecto abierto en nuestro Shield.



Una vez puesto el jumper en modo programación, lo conectamos a nuestro ordenador, a través de la conexión micro usb, (todavía no se debe conectar a la placa arduino).

Abrimos una línea de comandos, y nos poscionamos en la ruta donde hemos intalado Flip. A continuación ejecutamos:

```
batchisp.exe -device AT32UC3A1256 -hardware usb -operation erase f  
memory flash blankcheck loadbuffer c:\wifi_dnld.elf program verify start  
reset 0
```

Obtendremos si todo ha ido correcto lo siguiente:

```
AT32UC3A1256 - USB - USB/DFU  
  
Device selection..... PASS  
Hardware selection..... PASS  
Opening port..... PASS  
Reading Bootloader version..... PASS 1.0.2  
Erasing..... PASS  
Selecting FLASH..... PASS  
Blank checking..... PASS 0x000000 0x3ffff  
Parsing ELF file..... PASS wifi_dnld.elf  
WARNING: The user program and the bootloader overlap!  
Programming memory..... PASS 0x000000 0x2a10b  
Verifying memory..... PASS 0x000000 0x2a10b  
Starting Application..... PASS RESET 0  
  
Summary: Total 11 Passed 11 Failed 0
```

Tras esto, reseteamos nuestro Shield y esperamos a que nuestro equipo lo detecte de nuevo, y volvemos a ejecutar otro comando:

```
batchisp.exe -device AT32UC3A1256 -hardware usb -operation erase f  
memory flash blankcheck loadbuffer c:\wifiHD.elf program verify start reset  
0
```

```
AT32UC3A1256 - USB - USB/DFU  
  
Device selection..... PASS  
Hardware selection..... PASS  
Opening port..... PASS  
Reading Bootloader version..... PASS 1.0.2  
Erasing..... PASS  
Selecting FLASH..... PASS  
Blank checking..... PASS 0x000000 0x3ffff  
Parsing ELF file..... PASS wifiHD.elf  
WARNING: The user program and the bootloader overlap!  
Programming memory..... PASS 0x000000 0x3fe2b  
Verifying memory..... PASS 0x000000 0x3fe2b  
Starting Application..... PASS RESET 0  
  
Summary: Total 11 Passed 11 Failed 0
```



Ya tendríamos nuestro WiFi Shield actualizado, antes de conectarlo a nuestra placa de Arduino debemos volver a poner el jumper en la posición que traía por defecto.

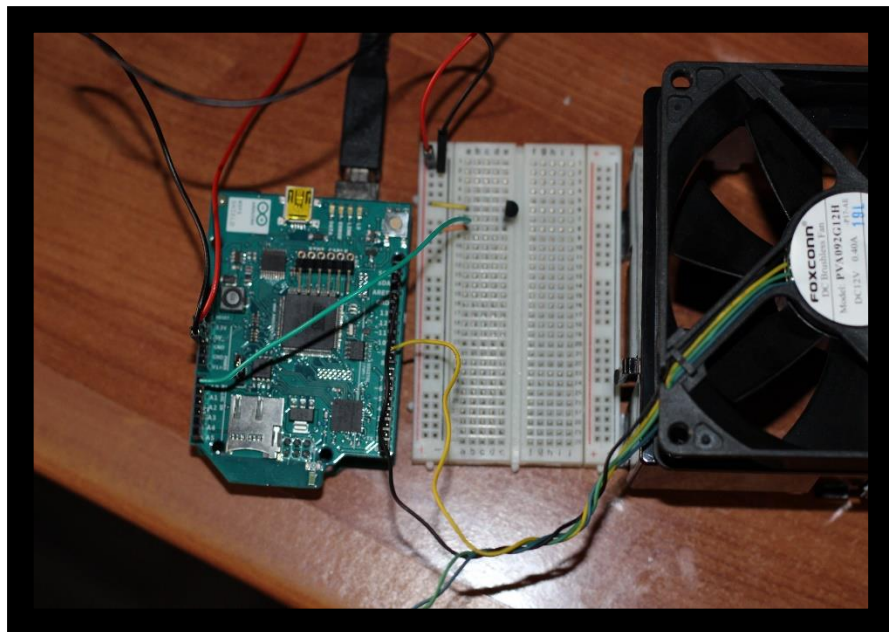
Por último, y si todo ha ido correcto y a la hora de hacer cualquier prueba nos sale como error WIFI SHIELD NOT PRESENT, si nuestra actualización fue correcta, hay que cerciorarse de que el Shield este bien conectado a nuestra placa Arduino, al principio puede dar miedo y no hayamos apretado lo suficiente, recomiendo revisar esta parte ya que nos ahorrará mucho tiempo.

### 5.3.3. Montaje del Circuito

Ya conectado el WiFiShield a nuestra placa Arduino UNO rev 3, nos apoyaremos de un protoboard, para la conexión del sensor de temperatura y nuestro ventilador.

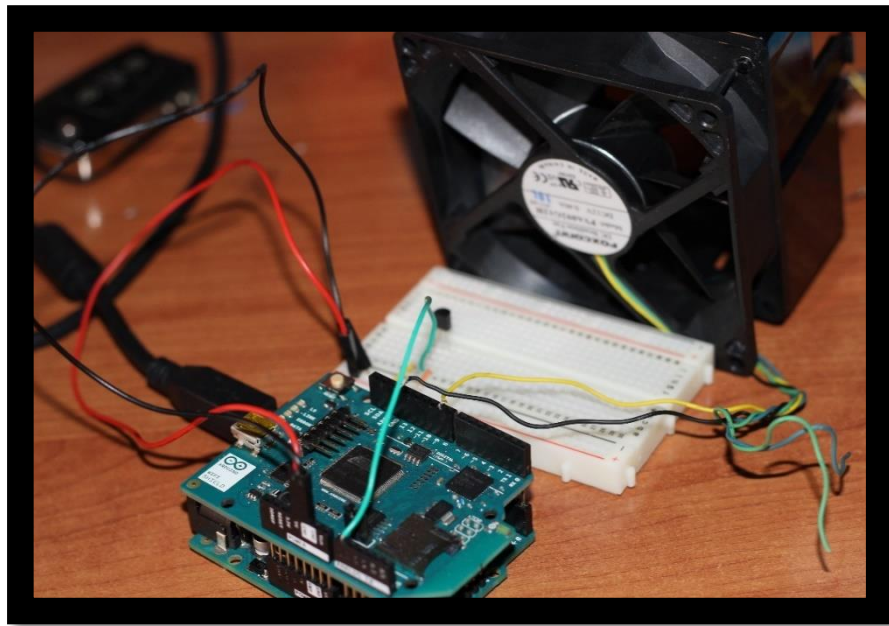
- Usaremos el pin A0 para el sensor de temperatura.
- El pin 9 para el ventilador conectándolo también a GND.
- Pin 5V y GND para la alimentación.

Como podemos observar en la imagen la conexión quedaría de esta forma:



**Montaje de nuestro dispositivo**





**Montaje de nuestro dispositivo**

#### 5.4. Plataforma ThingSpeak

Una vez montada la parte hardware pasaremos a configurar nuestra plataforma software. Lo primero que tenemos que hacer es darnos de alta en la plataforma, una vez dado de alta, nos vamos a la pestaña de Channels y creamos nuestro canal, rellenamos los siguientes campos:

<b>Name</b>	PFC Temperatura	
<b>Description</b>	PFC UOC sensor que mide la temperatura de una habitación.	
<b>Metadata</b>		
<b>Tags</b>	Temperatura	
<b>Latitude</b>	<input type="text"/>	
<b>Longitude</b>	<input type="text"/>	
<b>Elevation</b>	<input type="text"/>	
<b>Make Public?</b>	<input type="checkbox"/>	
<b>URL</b>	<input type="text"/>	
<b>Video ID</b>	<input type="text"/>	<input type="radio"/> YouTube <input type="radio"/> Vimeo
<b>Field 1</b>	Temperatura	<input type="checkbox"/> remove field
<b>Field 2</b>	Tweet	<input type="checkbox"/> remove field

Con esto ya tenemos creado el canal donde se volcarán las medidas registradas por nuestro sensor de temperatura, un dato importante que tenemos que anotar para poder subir los datos es la WRITE API KEY, la cuál a la hora de programar nuestro Arduino deberemos incluirla para poder subir los datos a la plataforma.

#### 5.4.1. Aplicación ThingTweet

Haciendo uso de las aplicaciones que nos brinda esta plataforma, usaremos la aplicación thingTweet, la cual nos permite integrar la red social Twitter con Thingspeak. El proceso como muestra la imagen es el siguiente:

Una vez introducida nuestra cuenta y la contraseña ya tenemos integrada nuestra cuenta con thingspeak, nos facilitará una API key que deberemos usar a la hora de programar en Arduino.

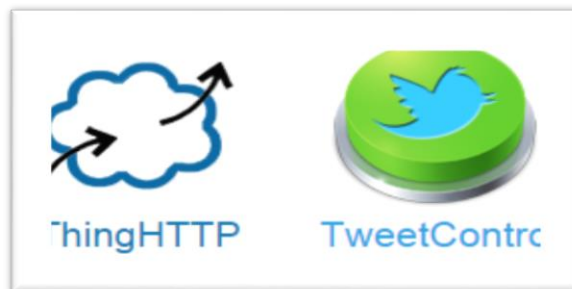
Twitter Account	API Key	Action
UocPFC	MOXRRG593GJZ11H W	<div>Regenerate API Key</div> <div>Unlink Account</div>

De la imagen anterior tenemos que tener en cuenta a la hora de programar en Arduino la API Key, la cual es la que nos permitirá que nuestro arduino envíe un tweet con el valor de la temperatura.

Como se puede ver en la siguiente imagen nuestro arduino va mandando tweets a nuestra cuenta de Twitter.



#### 5.4.2. Aplicación TweetControl + ThingHttp.



Ya tenemos nuestro Arduino subiendo datos a Twitter, nuestro siguiente paso es poder enviar un Tweet y que este llegue a la plataforma ThingSpeak para poder posteriormente procesar ese tweet.

Para ello lo primero que tenemos que hacer será hacer uso de la aplicación **ThingHttp**, que nos dará comunicación entre nuestra cuenta de Twitter y la plataforma **ThingSpeak**.

A la hora de configurar tenemos que tener en cuenta los siguientes campos:

- URL: <http://api.thingspeak.com/update>
- Method: POST

<b>Name</b>	<input type="text" value="Tweet"/>
<b>API Key</b>	<input type="text" value="ZHI7TAAGNEB5WU16"/>
<b>URL</b>	<input type="text" value="http://api.thingspeak.com/update"/>
<b>ITTP Auth Username</b>	<input type="text"/>
<b>ITTP Auth Password</b>	<input type="text"/>
<b>Method</b>	<input type="text" value="POST"/>

- **Body:**api\_key=WRITE\_API\_KEY&field=%%trigger%%&status%%status%  
**WRITE\_API\_KEY=** Es la llave de nuestro canal la conseguimos en Chanel en la parte DATA improt/export.  
 El campo **field** es el campo donde volcará el tweet (trigger), en nuestro caso hemos usado el **field2**, ya que el field1 es donde vuelca los valores de la temperatura.

<b>Body</b>	<input type="text" value="api_key=F4O6CQO4CEDUEIF8&amp;field2=%%trigger%%&amp;status=%%status%%"/>
-------------	--

Ya tenemos la comunicación entre la plataforma Thingspeak y Twitter, ahora nos queda que el tweet llegue al field2 de nuestro canal y se muestre en una ventana. Para ello, lo primero que tenemos que hacer es irnos a APP→TweetControl→New TweetControl. Debemos introducir el usuario de nuestra cuenta de Twitter, el trigger y la el ThingHttp que creamos anteriormente.

El trigger consistirá en el tweet que enviaremos desde nuestra cuenta, a la hora de enviar el Tweet es necesario que sigamos la siguiente nomenclatura para que la aplicación pueda filtrar: **@thingspeak trigger**.

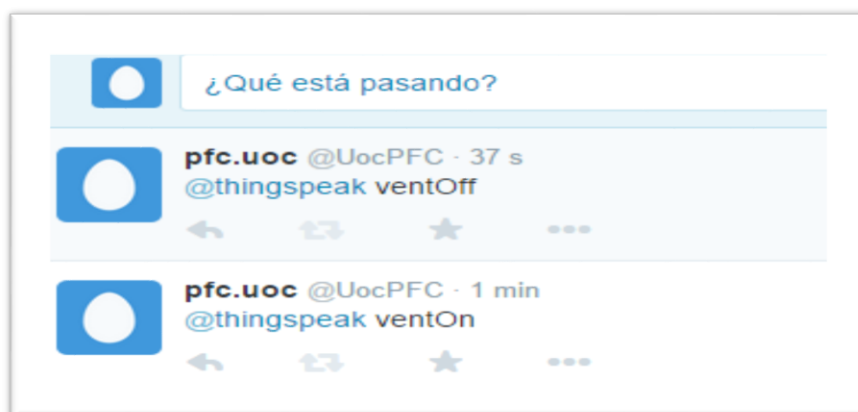
Debemos crear dos TweetControl uno para activar el ventilador y el otro para apagarlo, los trigger correspondientes serán: **@thingspeak ventOn** y **@thingspeak ventOff**.

Twitter Account	<input type="text" value="UocPFC"/>
Trigger	<input type="text" value="ventOn"/>
ThingHTTP Action	<input type="text" value="Tweet"/>
<input type="button" value="Save TweetControl"/>	

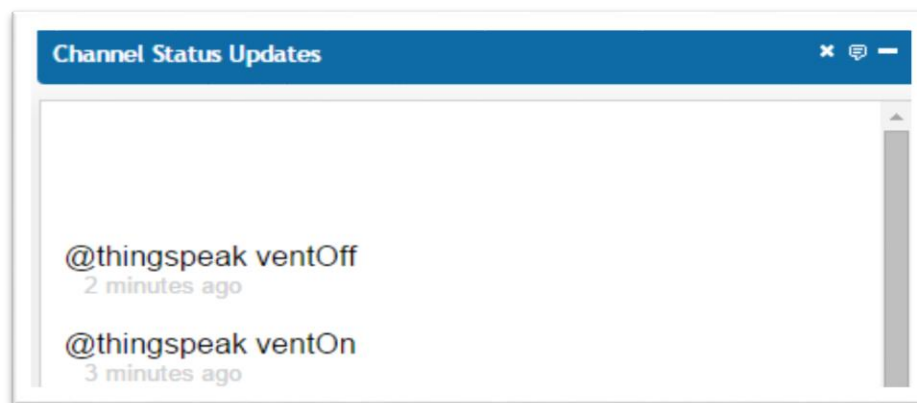
☐ Anonymous TweetControl

Twitter Account	<input type="text" value="UocPFC"/>
Trigger	<input type="text" value="ventOff"/>
ThingHTTP Action	<input type="text" value="Tweet"/>
<input type="button" value="Save TweetControl"/>	

Ya tenemos todo listo para poder probar, desde nuestra cuenta de Twitter enviaremos los siguientes tweets: **@thingspeak ventOn** y **@thingspeak ventOff**.



Estos dos tweet deben llegar a nuestra canal de Thingspeak, como podemos observar en la imagen ambos llegan correctamente.



A través de esta dirección

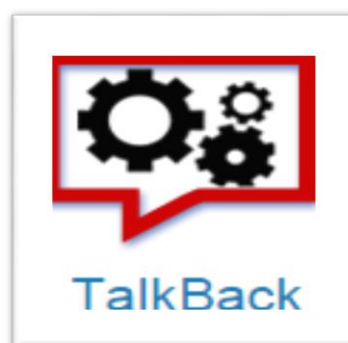
<http://api.thingspeak.com/channels/37205/feed.json?key=F4O6CQO4CEDUEIF8>

Podemos comprobar como el campo field2 se ha actualizado según le han llegado los diversos tweets, esto nos servirá más adelante para poder activar nuestro ventilador.

```
2015-05-26T16:16:46Z", "entry_id": 293, "field1": null, "field2": "venton"},
{"created_at": "2015-05-26T16:17:17Z", "entry_id": 294, "field1": null, "field2": "ventoff"}]}
```

Antes de continuar explicando el proceso para poder activar nuestro ventilador, hay que aclarar que en este punto ya sería suficiente programando directamente en Arduino y leyendo el campo field2, pero para complicarlo un poco más y dar una imagen mayor de la plataforma vamos a seguir usando las aplicaciones TalckBacks y React, por supuesto deberemos volveremos a tener que utilizar ThingHttp.

#### 5.4.3. Aplicación TalkBack



**TalkBack** será la aplicación que interactúe finalmente con nuestro Arduino. Será el encargado de mandar los comandos de encendido y apagado a nuestra placa. A la hora de configurarlo debemos rellenar el **Chanel** y los comandos a enviar al dispositivo, tal cual aparece en la siguiente imagen:

The screenshot shows the configuration page for a device named 'TalkBack vent'. The 'API Key' is '2UTZBYGPKALWB1GC'. The 'Log to Channel' is set to 'PFC Temperatura (37205)'. Under the 'Commands' section, there are two command entries. The first entry is for 'Position 1' with 'Command ID: 99276' and a 'Command string' of 'TURN\_ON'. The second entry is for 'Position 2' with 'Command ID: 99288' and a 'Command string' of 'TURN\_OFF'. A 'Remove command' link is present below the first command.

Name	TalkBack vent
API Key	2UTZBYGPKALWB1GC
Log to Channel	PFC Temperatura (37205)
Commands	<div><div>Position 1 Command ID: 99276 Command string TURN_ON <a href="#">Remove command</a></div><div>Position 2 Command ID: 99288 Command string TURN_OFF</div></div>

Tenemos que tener en cuenta, que nuestra intención es que se active el ventilador cuando enviamos un tweet, por lo que alguien debe de decirle a **TalkBack** que ha llegado un tweet para activar/apagar el ventilador y mande el comando TURN\_ON o TURN\_OFF a nuestro dispositivo.

Bien, es aquí donde finalmente entra en escena nuestra última aplicación llamada REACT que será la encargada de comunicarse con **TalkBack** para decirle que le ha llegado un tweet para encender/apagar nuestro dispositivo. Recordemos que ese tweet ya lo tenemos volcado en nuestro campo field2 gracias a la aplicación **TweetControl**.



#### 5.4.4. Aplicación REACT + ThingHTTP



Antes de continuar hablando acerca de la aplicación REACT, vamos a repasar todo lo que hemos hecho hasta ahora.

Nuestra intención es enviar un Tweet desde Twitter para que nuestro ventilador se encienda/apague. Para ello usando **TweetControl** y **ThingHttp**, hemos conseguido hacer llegar el tweet correspondiente a nuestro canal y guardarlo en el campo **field2**. En este punto tenemos por ejemplo **field2=ventOn**.

Ahora necesitamos que alguien le diga a **TalkBack** que se ha recibido ese dato para que mande el comando TURN\_ON a nuestro dispositivo y así active el ventilador. Es aquí donde entra en juego REACT.

Ya puestos en situación, REACT será quien le comunique la llegada del tweet a **TalkBack**. Para ello necesitamos otra vez de la funcionalidad de **ThingHTTP**, que nos permitirá que REACT y **TalkBack** se comuniquen.

Debemos configurar dos nuevos **ThingHttp**:

- **Request ventOn:**

Los campos a rellenar aquí serán los siguientes:

**URL:** [https://api.thingspeak.com/talkbacks/99278/commands/TURN\\_ON](https://api.thingspeak.com/talkbacks/99278/commands/TURN_ON)

El número 99278 corresponde al comandID del comando TURN\_ON de la aplicación TalkBack.

**Mehod:** PUT

**Body:** api\_key=2UTZBYGPKALWB1GC&command\_string=TURN\_ON&position=1

En el body debemos poner la API Key de la aplicación TalkBack corresponde a: 2UTZBYGPKALWB1GC, le indicamos el comando y la posición del mismo.

<b>Name</b>	Request ventOn
<b>API Key</b>	QLSDUQFIYW4IXQOP
<b>URL</b>	https://api.thingspeak.com/talkbacks/99278/commands/T
<b>HTTP Auth Username</b>	
<b>HTTP Auth Password</b>	
<b>Method</b>	PUT ▼

<b>Body</b>	api_key=2UTZBYGPKALWB1GC&command_string=TURN_ON&position=1
-------------	--

- **Request ventOff:**

Los campos a rellenar aquí son los mismos que antes variando el commandID y la posición:

**URL:** [https://api.thingspeak.com/talkbacks/99288/commands/TURN\\_OFF](https://api.thingspeak.com/talkbacks/99288/commands/TURN_OFF)

**Method:** PUT

**Body:** api\_key=2UTZBYGPKALWB1GC&command\_string=TURN\_OFF&position=2

<b>Name</b>	Request ventOff
<b>API Key</b>	81T7E2YHYHDB506H
<b>URL</b>	https://api.thingspeak.com/talkbacks/99288/commands/T
<b>HTTP Auth Username</b>	
<b>HTTP Auth Password</b>	
<b>Method</b>	PUT ▼
<b>Content Type</b>	
<b>HTTP Version</b>	1.1 ▼

<b>Body</b>	api_key=2UTZBYGPKALWB1GC&command_string=TUR N_OFF&position=2
-------------	---

Es hora de configurar REACT, para ello debemos configurar los siguientes campos:

<b>React Name</b>	React ventOn
<b>Condition Type</b>	String
<b>Test Frequency</b>	On Data Insertion
<b>Condition</b>	If channel PFC Temperatura (37205)
	field 2 (Tweet)
	contains
	ventOn

<b>Action</b>	ThingHTTP
	then perform ThingHTTP
	Request ventOn
<b>Options</b>	<input type="radio"/> Run action only the first time the condition is met <input checked="" type="radio"/> Run action each time condition is met

Explicamos los campos que se han rellenado:

- **Condition Type: String**  
Elegimos String porque como hemos venido comendo nuestro campo se aloja en **field2=ventOn**
- **Test Frequency: On Data Insertion**  
Se consultará cuando se introduzca un dato
- **Condition:** La condición será, si en nuestro canal en el campo field2 contiene ventOn activa una acción.
- **Action:** La acción será mediante ThingHttp creado anteriormente en este caso **Request ventOn**. Esto le dirá a TalkBack que ha

llegado un tweet que ordena que se active el ventilador, por lo que TalkBack le dirá a nuestro dispositivo que active el mismo.

Ya hemos creado la orden para que encienda nuestro ventilador solo nos queda tratar la petición de apagado.

<b>React Name</b>	React ventOff
<b>Condition Type</b>	String
<b>Test Frequency</b>	On Data Insertion
<b>Condition</b>	If channel PFC Temperatura (37205)
	field 2 (Tweet)
	contains
	ventOff
<b>Action</b>	ThingHTTP

<b>Action</b>	ThingHTTP
	then perform ThingHTTP
	Request ventOff
<b>Options</b>	<input type="radio"/> Run action only the first time the condition is met <input checked="" type="radio"/> Run action each time condition is met

Con esto ya podremos enviar nuestro tweet **@thingspeak ventOn** y así activar nuestro ventilador, por el contrario si queremos apagarlo debemos mandar el siguiente tweet **@thingspeak ventOff**.

## 5.5. Pruebas realizadas

En este punto mostramos mediante imágenes, una serie de pruebas realizadas con el fin de mostrar la funcionalidad de nuestra aplicación.

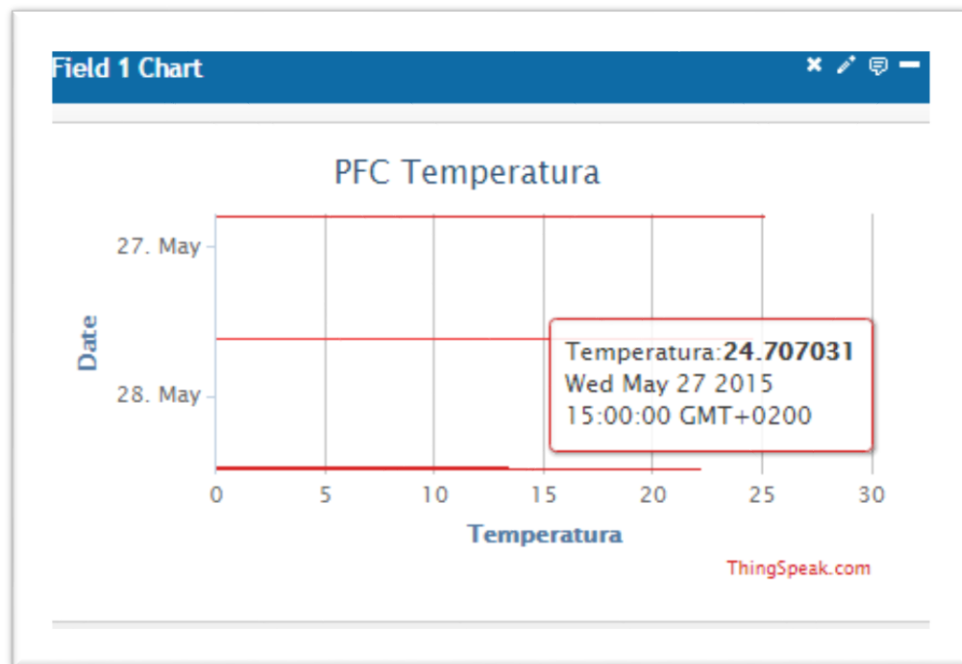
### 5.5.1. Medida tomada por Arduino y enviada a ThingSpeak.

En este punto podemos observar en la primera imagen, referente al Serial monitor de Arduino, como primero se conecta a nuestra red WiFi, mide la temperatura de nuestra habitación y posteriormente sube el dato medido a la plataforma Thingspeak. En este caso la medida tomada son 24,7 grados.

```
Attempting to connect to SSID: WLAN_67BA
sensor Value: 153Volts: 0.75, degrees C: 24.71
Connecting to ThingSpeak...
```

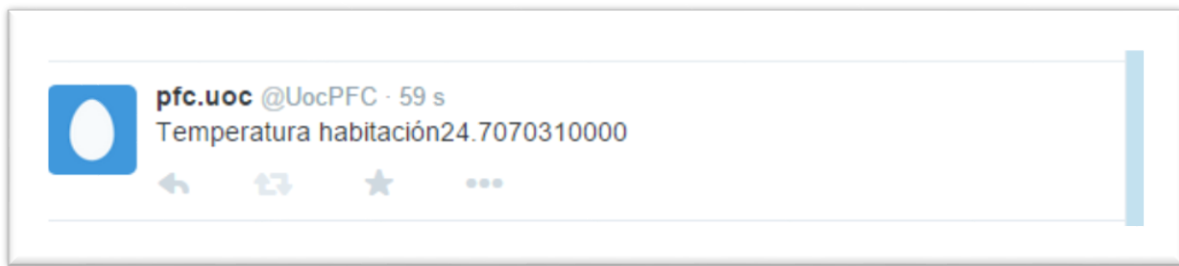
```
Conectado
Conectando a ThingSpeak...
```

```
Manda Tweetapi_key=MOXRREG593GJZ11HW&status=Temperatura habitaciÃ³n24.7070310000
```



### 5.5.2. Envío de un Tweet desde Plataforma Thingspeak

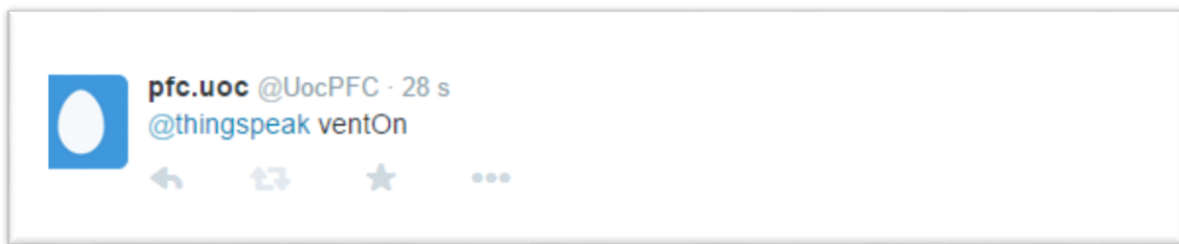
En esta imagen se muestra como tras subir la medida a nuestra plataforma ThingSpeak, haciendo uso de la aplicación ThingTweet, se envía el Tweet con la medida tomada, en este caso la medida tomada son 24,7 grados.



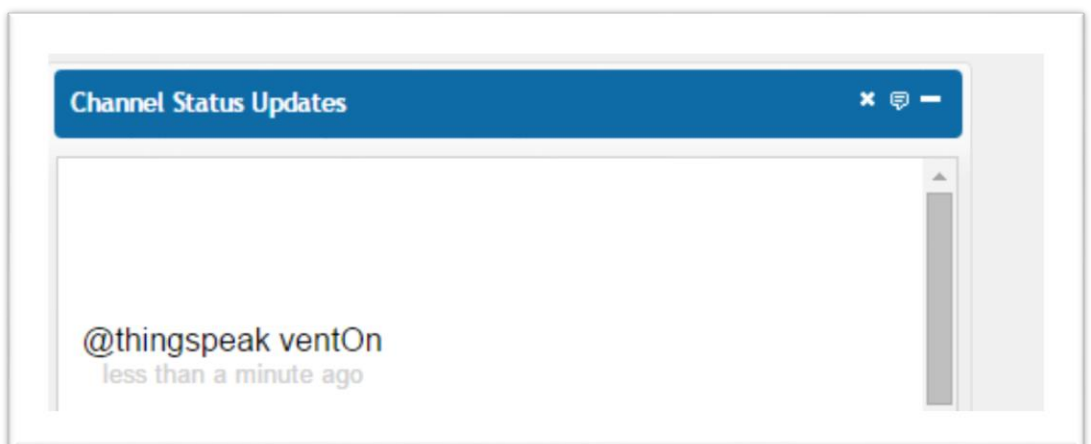
```
{"created_at": "2015-05-28T10:21:47Z", "entry_id": 333, "field1": "24.7070310000", "field2": null}, {
```

### 5.5.3. Tweet enviado a ThingSpeak para encender nuestro ventilador.

Aquí vemos como enviamos un Tweet desde la red social Twitter:



En la siguiente imagen, vemos como llega el Tweet a nuestra plataforma y se enciende el ventilador:

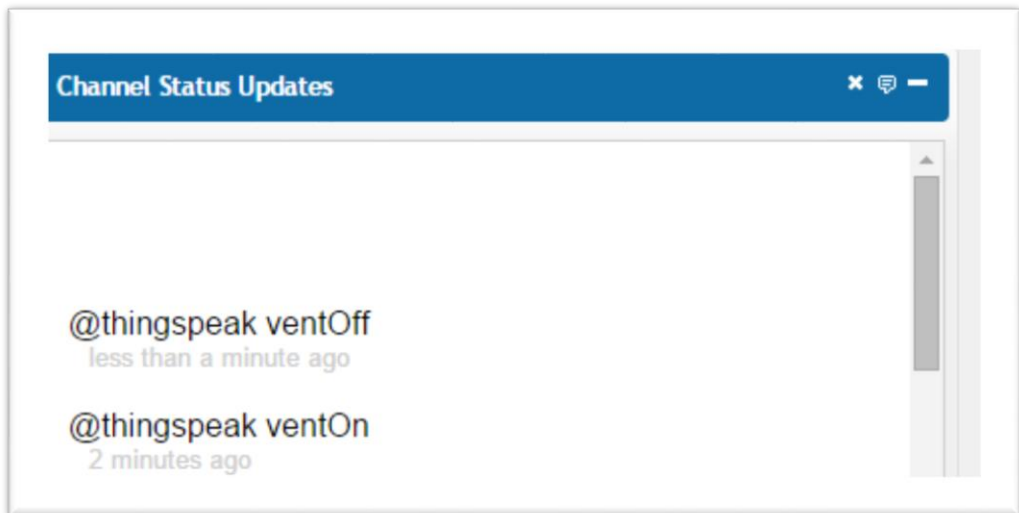


#### 5.5.4. Tweet enviado a ThingSpeak para encender nuestro ventilador.

Haremos lo mismo para apagar nuestro ventilador.



Observamos como llega a nuestra plataforma y se apaga nuestro ventilador.



### 5.6. Presupuesto del Proyecto

Estamos ante un proyecto académico, por lo que el coste estaría bastante de lo que sería un proyecto real a nivel corporativo.

Aún así, vamos a hacer una simulación de lo que podría costarnos realizar este mismo proyecto, si quisiéramos comercializarlo.

Los perfiles que necesitaríamos serían los siguientes:

- Jefe de Proyecto: 100 €/hora
- Consultor Analista: 60€/hora
- Programador: 40€/hora
- Técnico soporte: 35€/hora
- Tester: 35€/hora

Tarea	Recurso	horas	Precio hora	Precio total
Kick Off	Jefe Proyecto	2	100€	200
Toma Requisitos	Consultor	4	60	240
Diseño Software	Consultor	14	60	840
Desarrollo	Programador	40	35	1400
Testeo	Tester	12	35	420
Depuración Código	Programador	8	35	140
Memoria Final	Jefe Proyecto	18	100	1800
Presentación	Jefe Proyecto	2	100	200
Soporte	Técnico	40	0	0
<b>TOTAL</b>				<b>5240</b>

Material	Cantidad	Precio
Arduino UNO rev3	1	21,55€
WIFI-SHIELD	1	74,96€
Sensor Temperatura	1	2,48€
Ventilador	1	3,25
Placa Board	1	1,99€
Cable USB	1	2 €
Cables conexiones	1	0,5 €
<b>Total</b>		<b>106,73€</b>

El precio total ascendería a **5346,73 €**. Aún así estamos en el supuesto que hubiera que hacer un estudio, muchas empresas ya venden dispositivos plug and play listos para conectar con unos precios muy competitivos.



## 6. Conclusiones

El desarrollo de este proyecto final de carrera, parte de la idea de profundizar en una nueva revolución tecnológica que poco a poco se esta instaurando en nuestro día a día., lo cual me ha llevado a adquirir una serie de conocimientos, muy interesantes tanto a nivel teórico como a nivel práctico.

En un primer momento, la idea del proyecto era hacer únicamente un estudio de las diversas plataformas software, tras hablar con mi tutor me recomendó en la medida de lo posible, incluir una aplicación al proyecto. La idea resultó atractiva, pero siendo sincero, no acababa de convencerme sobre todo por la limitación de tiempo y siendo la entrega del PFC en un solo semestre. Finalmente he de decir que fue una decisión acertada.

También se han incluido como ha podido comprobar el lector, un estudio de las diferentes plataformas hardware además de sus ámbitos de aplicación, así como una demostración de pruebas realizadas y una simulación de lo que sería un presupuesto real.

Respecto al estudio de las plataformas Software, en Internet se encuentra una gran cantidad y variedad de plataformas, se ha optado por elegir aquellas que a mi juicio parecen más interesantes o que tienen alguna característica especial, obviamente no están todas, pero si se ha intentado representar aquellas que son Open Source como ThingSpeak, o como hemos nombrado a lo largo del documento las que comportan un ecosistema como Electronic Imp y Spark(con hardware y software), o aquellas con un ámbito de aplicación específico como Zatar en el mundo de la logística.

En las plataformas Hardware era imprescindible nombrar a Arduino, sobre todo por su gran popularidad y presencia en el sector de la educación. La intención al igual que con las plataformas software es mostrar una variedad de placas y tecnologías, y las diversas funcionalidades que se le pueden dar. Por ejemplo usaríamos ZigBee para un proyecto Smart Home, así como Waspote para un proyecto de agricultura inteligente, Arduino o Intel Galileo para proyectos educativos o prototipos.

Respecto a la aplicación, hay que tener en cuenta que no es el objetivo principal de este proyecto, este proyecto no nace con la idea de desarrollar una aplicación, si que es verdad que esta sección es un punto importante del proyecto, cuya finalidad es aplicar y demostrar de una forma práctica los conocimientos adquiridos durante todo el proyecto. Por supuesto también, adquirir nuevos conocimientos como conocer el IDE de Arduino y la forma de como se programa.

Por último, comentar que este proyecto me ha servido para poder investigar sobre una tecnología en concreto, aplicar esos conocimientos diseñando una aplicación y desarrollándola. Por supuesto todos los conocimientos adquiridos durante la carrera han ayudado a crear una base técnica que ha ayudado en la consecución de este proyecto. Pero sin duda lo más importante en el transcurso de la carrera es como una vez finalizada esta, uno se da cuenta de como se ha transformado la forma de pensar y analizar cualquier problemática.

Finalmente quiero agradecer a mi tutor José Lopez Vicario, el tiempo y los consejos dedicados durante todo el semestre, así como las mejoras necesarias que debía ir aplicando en las diferentes fases del proyecto.

# ANEXO

- Código Fuente Arduino

```
TempThingspeak$
1 #include <SPI.h>
2 #include <WiFi.h>
3 #include "HttpClient.h"
4
5 WiFiClient client;
6
7 //Configuración Conexión WIFI
8 char ssid[] = "WLAN_67BA"; //SSID (name)
9 char pass[] = "3fb0a16d52b089a81c89"; //password WPA
10 int status = WL_IDLE_STATUS;
11
12 // Configuración ThingSpeak
13 char thingSpeakAddress[] = "api.thingspeak.com";
14 String writeAPIKey = "F406CQ04CEDUE1F8";
15 String thingTweetAPIKey = "MOXPRG593GJZ11HW";
16 const int updateThingSpeakInterval = 16 * 1000; // Intervalo de tiempo para actualizar Thingspeak (numero de seg * 1000 = intervalo)
17
18 //Cinfiguración TalkBack
19 String thingSpeakAPI = "api.thingspeak.com";
20 String talkBackAPIKey = "2UTZBYGPKALWB16C";
21 String talkBackID = "2191";
22 const int checkTalkBackInterval = 16 * 1000;
23
24 //Variables
25 const int sensorPin = A0;
26 long lastConnectionTime = 0;
27 boolean lastConnected = false;
28 int failedCounter = 0;
29
30 void setup()
31 {
32
33     // Setup On-board LED
34     pinMode(9, OUTPUT);
35
36     // Start Serial for debugging on the Serial Monitor
37     Serial.begin(9600);
38
39
40     // Verifica que esta presente WIFI SHIELD:
41     if (WiFi.status() == WL_NO_SHIELD) {
42         Serial.println("WiFi shield not present");
43         // no continua Acaba el programa:
44         while (true);
45     }
46
47     String fv = WiFi.firmwareVersion();
48     if ( fv != "1.1.0" )
49         Serial.println("Actualiza el firmware");
50
51     // Intenta conectar a la red:
52     while (status != WL_CONNECTED) {
53         Serial.print("Intentando conectar SSID: ");
54         Serial.println(ssid);
55         // Conectar a la red WPA:
```

```

56     status = WiFi.begin(ssid, pass);
57
58     // Espera 10 segundos para conectar:
59     delay(10000);
60
61 }
62
63 }
64
65 void loop()
66 {
67     // Leer valor de la entrada analogica A0
68     int sensorVal = analogRead(sensorPin);
69
70
71     //Temperatura
72     // Envía valor de sensor al puerto serie
73     Serial.print("Valor sensor: ");
74     Serial.print(sensorVal);
75
76     // Convertimos valor ADC
77     float voltage = (sensorVal / 1024.0) * 5.0;
78
79     // Enviamos el valor de ADC leído al puerto serie
80     Serial.print(", Volts: ");
81     Serial.print(voltage);
82
83     // convierte volt en temperatura
84     Serial.print(", grados C: ");
85     float temperature = (voltage - .5) * 100;
86     Serial.println(temperature);
87     String analogValue0 = String(temperature, DEC);
88
89     // Imprimimos en monitor
90     if (client.available())
91     {
92         char c = client.read();
93         Serial.print(c);
94     }
95
96     // Desconectando de Thingspeak
97     if (!client.connected() && lastConnected)
98     {
99         Serial.println("...desconectando");
100        Serial.println();
101
102        client.stop();
103    }
104
105    // Actualiza ThingSpeak
106    if(!client.connected() && (millis() - lastConnectionTime > updateThingSpeakInterval))
107    {
108        updateThingSpeak("field1="+analogValue0);
109        updateTwitterStatus("Temperatura habitación: "+analogValue0);

```

```

110     delay(10000);
111 }
112 if(talkBackCommand == "TURN_OFF"){
113     digitalWrite(4, LOW);
114     Serial.println("OFF");
115 }
116 if(talkBackCommand == "TURN_ON"){
117     digitalWrite(4, HIGH);
118     Serial.println("ON");
119 }
120 lastConnected = client.connected();
121 }
122
123 void updateThingSpeak(String tsData)
124 {
125     if (client.connect(thingSpeakAddress, 80))
126     {
127         client.print("POST /update HTTP/1.1\n");
128         client.print("Host: api.thingspeak.com\n");
129         client.print("Connection: close\n");
130         client.print("X-THINGSPEAKAPIKEY: "+writeAPIKey+"\n");
131         client.print("Content-Type: application/x-www-form-urlencoded\n");
132         client.print("Content-Length: ");
133         client.print(tsData.length());
134         client.print("\n\n");
135
136         client.print(tsData);
137
138         if (client.connected())
139         {
140             Serial.println("Conectando a ThingSpeak...");
141             Serial.println();
142
143             failedCounter = 0;
144             Serial.println("Conectado");
145         }
146         else
147         {
148             failedCounter++;
149
150             Serial.println("Conexión fallida a Thingspeak (" +String(failedCounter, DEC)+")");
151             Serial.println();
152         }
153     }
154     else
155     {
156         failedCounter++;
157
158         Serial.println("Conexión fallida a Thingspeak (" +String(failedCounter, DEC)+")");
159         Serial.println();
160
161         lastConnectionTime = millis();
162     }
163 }

```

```

170 void updateTwitterStatus(String tsData)
171 {
172     if (client.connect(thingSpeakAddress, 80))
173     {
174         // Create HTTP POST Data
175         tsData = "api_key="+thingTweetAPIKey+"&status="+tsData;
176
177         client.print("POST /apps/thingtweet/1/statuses/update HTTP/1.1\n");
178         client.print("Host: api.thingspeak.com\n");
179         client.print("Connection: close\n");
180         client.print("Content-Type: application/x-www-form-urlencoded\n");
181         client.print("Content-Length: ");
182         client.print(tsData.length());
183         client.print("\n\n");
184
185         client.print(tsData);
186
187         lastConnectionTime = millis();
188
189         if (client.connected())
190         {
191             Serial.println("Conectando a ThingSpeak...");
192             Serial.println();
193
194             failedCounter = 0;
195         }
196         else
197         {
198             failedCounter++;
199
200             Serial.println("Conexión fallida a Thingspeak("+String(failedCounter, DEC)+")");
201             Serial.println();
202         }
203     }
204     else
205     {
206         failedCounter++;
207
208         Serial.println("Conexión fallida a Thingspeak("+String(failedCounter, DEC)+")");
209         Serial.println();
210
211         lastConnectionTime = millis();
212     }
213
214     delay(10000);
215 }
216
217 void getTalkBack() {
218     String tsData;
219     tsData=talkBackID + "/commands/execute?api key=" + talkBackAPIKey;
220

```

```

221 if (!client.connected() && (millis() - lastConnectionTimeTalkBack > getTalkBackInterval())){
222     if (client.connect("api.thingSpeak.com", 80)){
223         client.println("GET /talkbacks/"+tsData+" HTTP/1.0");
224         client.println();
225         lastConnectionTimeTalkBack = millis();
226     if (client.connected()){
227         Serial.println("GET TalkBack command");
228         Serial.println("Connecting to ThingSpeak");
229         Serial.println("Server response");
230         Serial.println();
231         failedCounterTalkBack = 0;
232         while(client.connected() && !client.available()) {delay(2000)}; //waits for data
233         while (client.connected() || client.available()){
234             char charIn = client.read();
235             talkBackCommand += charIn;
236             Serial.print(charIn);
237         }
238         talkBackCommand = talkBackCommand.substring(talkBackCommand.indexOf("_CMD_")+5);
239
240         Serial.println("Disconnected");
241         Serial.println(talkBackCommand);
242         Serial.println();
243     }
244     else{
245         failedCounterTalkBack++;
246         Serial.println("Connection to ThingSpeak failed (" +String(failedCounterTalkBack, DEC)+")");
247         Serial.println();
248         lastConnectionTimeChannels = millis();
249     }
250 }
251 else{
252     failedCounterTalkBack++;
253     Serial.println("Connection to ThingSpeak Failed (" +String(failedCounterTalkBack, DEC)+")");
254     Serial.println();
255     lastConnectionTimeTalkBack = millis();
256 }
257 }
258 client.stop();
259 Serial.flush();
260
261 }

```

## BIBLIOGRAFÍA

<http://postscapes.com/internet-of-things-software-guide#protocols>

<http://www.datamation.com/open-source/35-open-source-tools-for-the-internet-of-things-1.html>

<https://thingspeak.com/>

<https://www.carriots.com/>

<https://www.particle.io/>

<https://electricimp.com/>

<http://www.blaulabs.com/es/>

<http://www.thinkingthings.telefonica.com/>

<http://www.zatar.com/>

<https://es.wikipedia.org/>

<http://www.libelium.com/es>

<http://www.arduino.cc/>

<http://www.intel.es/content/www/es/es/do-it-yourself/galileo-maker-quark-board.html>

<http://www.zigbee.org/>