Universidad del Azuay
Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

Prototipo de monitoreo ambiental aplicando el Internet de las cosas con Arduino y Cloud Computing

Trabajo de graduación previo a la obtención del título de Ingeniero de Sistemas y Telemática

Autores:

Francisco David Salgado Castillo

David Santiago Coello Moncayo

Director: Ing. Oswaldo Merchán Manzano, Msc**.**

Cuenca, Ecuador
2015

Dedicatoria

Le dedico este trabajo a Eliana Moncayo, que gracias a su apoyo incondicional pude alcanzar esta meta tan importante en mi vida, fue el pilar de mi vida universitaria que siempre estuvo ahí para impulsarme a seguir adelante y me alentó en los momentos más difíciles.

A Lucía Robalino, quien siempre me motivó a seguir adelante y así, al transformarse en mi compañera de vida, tuve la dicha de tener a alguien que esté ahí siempre que lo necesite y dándome su total apoyo y ánimo en todo momento.

A Jaime Matovelle, Juan Sebastián Landy, Juan Pablo León, Danny Noboa, Darío Espinoza, Santiago Vásquez, Cesar Vele, Francisco Salgado, Juan Esteban Calderón, Nataly Pizarro, Fabián Atariguana, Byron Guamán, William León, Verónica Peñafiel, Víctor Córdova, Henry Portilla, que gracias a esta carrera tuve la suerte de conocerlos y llegar a considerarlos en mi vida como mis amigos y que sin ellos mi vida universitaria y sobre todo la personal no hubiera sido la misma y que ahora conforman una parte muy importante en mi vida.

A Rómulo Astudillo, Giovanni Cordero, Andrés Bustamante, Javier Coello y Christian Aguilera, que son mis primeros amigos de la vida y que siempre estuvieron pendientes de mí, apoyándome y dándome fuerzas en los momentos difíciles de la carrera.

David Santiago Coello Moncayo

Dedicatoria

A mi familia y amigos, quienes me acompañaron y apoyaron durante todo este camino en la Universidad. A mis hermanas y hermano por todo lo que hemos podido compartir. A mi Mamá, Luz Marina, una incansable luchadora llena de amor que ha hecho de mí la persona que soy hoy. Y a mi Papá, Francisco, con quien tuve la maravillosa suerte de tenerlo como profesor; por su constante guía, enseñanza y cariño.

Francisco David Salgado Castillo

Agradecimiento

A nuestro Director, el Ingeniero Oswaldo Merchán, por su paciencia y apertura. Un admirable académico que nos llena de orgullo el haber trabajado con él.

Al personal del IERSE, en especial al Ingeniero Chester Sellers y al Ingeniero Omar Delgado; por su guía y apoyo en la realización de este trabajo y con quienes esperamos trabajar nuevamente en futuros proyectos.

Y a la Universidad del Azuay, la cual nos ha impartido una formación profesional y humana; con la que esperamos servir de manera correcta a nuestra sociedad.

ABSTRACT

This graduation work is about the development of an electronic prototype for monitoring the environmental variables: temperature, humidity, noise and CO concentration based on an approach focused on the concept of the "Internet of Things" Cloud Computing, and Wireless Sensor Networks (WSN). The prototype is made of Arduino free hardware plate which communicates via Ethernet and Wi-Fi through the Internet to ThingSpeak platform for processing data in the cloud, and then displaying the corresponding monitoring graphics in a web browser.

UNIVERSIDAD DEL AZUAY
Dpto. Idiomas

Translated by,
Lic. Lourdes Crespo

RESUMEN

El presente trabajo de titulación consiste en el desarrollo de un prototipo electrónico para el monitoreo de las variables ambientales: temperatura, humedad, ruido y concentración de CO, desde un abordaje enfocado al concepto de la “Internet de las Cosas”, Computación en la Nube, y Redes de Sensores Inalámbricos (WSN). El prototipo está realizado con la placa de hardware libre Arduino que se comunica por medio de Ethernet y WiFi con el internet a la plataforma ThingSpeak para el procesamiento de los datos en la nube. Finalmente, se realiza la gestión de la información que permite mostrar los gráficos de monitoreo correspondientes en un navegador web.

Índice de Contenidos

Índice de Figuras

Índice de Tablas

Índice de Listados

1. CAPÍTULO 1: El Internet de las Cosas

Introducción

> *We can make objects intelligent. We can make them think and speak. Pundits have dubbed this 'physical computing', 'ubiquitous computing' or 'ubicomp', or 'the Internet of Things'. Whatever you choose to call it, we are really talking about making magical things, enchanted objects.*
>
> Adrian McEwen & Hakim Cassimally

En este capítulo se expondrá sobre el concepto del Internet de las Cosas o **IoT** por sus siglas en inglés, una breve historia, tendencias y expectativas, servicios, aplicaciones y además de un análisis de la arquitectura que forma una red IoT y de los distintos protocolos de comunicación que se pueden utilizar.

1.1.Definición

El Internet de las Cosas, **IoT** por sus siglas en inglés ("*Internet of things*"), ha recibido varias denominaciones pero se puede describir como una red global de dispositivos inteligentes que pueden "sentir" e interactuar con su ambiente empleando el internet para su comunicación e interacción con usuarios y otros sistemas. (Doukas, 2012)

El uso de la palabra "Internet" en el término de IoT hace referencia a una metáfora: en la misma manera que las personas utilizan la web hoy en día, en un futuro las cosas también se comunicarán entre ellas, usarán servicios, proveerán datos y generarán valor agregado.

La idea del Internet de las Cosas sugiere que, en lugar de tener un pequeño número de dispositivos de computación muy poderosos como laptops, teléfonos inteligentes, tablets, etc., se podría tener un gran número de aparatos u objetos que sean "menos poderosos", objetos como un par de zapatos, un

espejo, un paraguas, un brazalete, una silla, etc., que monitoreen, registren y procesen datos relacionados a lo que sucede a su alrededor, su interacción con el medio ambiente, etc. Esta información puede provocar un mayor impacto en nuestras vidas que el que normalmente se obtendría con los dispositivos convencionales mencionados anteriormente. (McEwen & Cassimally, 2014)

1.2.Breve historia

Un término popular utilizado anteriormente que describía de manera similar el mismo concepto fue la "computación ubicua" o simplemente "ubicomp", que también refleja el alto número de objetos que podrían contener alguna tecnología de computación. (McEwen & Cassimally, 2014)

El término "*Internet of Things*" fue popularizado por el Auto-ID Center en el Instituto Tecnológico de Massachusetts (MIT), el cual en 1999 comenzó a diseñar y propagar una compañía con una infraestructura de identificación por radio-frecuencia (RFID). En 2002, su cofundador y antiguo jefe Kevin Ashton fue citado en la revista Forbes diciendo, "Necesitamos una internet de las cosas, una manera estandarizada en que las computadoras puedan entender al mundo real". (Mattern & Floerkemeier) Este artículo fue titulado "*The Internet of Things*" y fue el primer uso documentado del término en un sentido literal.

1.3.Estándares y Expectativas

Desde entonces, IoT fue ganando mayor importancia como una tendencia que daría forma a la definición de varios estándares en el sector de las Tecnologías de la información y de las Comunicaciones (ICT). A tal punto que organizaciones como la Unión Internacional de Telecomunicaciones (ITU) (Unión Internacional de Telecomunicaciones, s.f.), el IEEE (IEEE, s.f.), la ISO/IEC JTC 1 (ISO/IEC, s.f.) (Normalización en el campo de la tecnología de la información) han formado sus propios grupos de trabajo, resoluciones e iniciativas de estándares globales que permitirán a los proveedores de servicios

de todo el mundo ofrecer una amplia gama de servicios que se esperan de esta tecnología.

Las expectativas son enormes. Gartner, una empresa de consultoría de TI en Connecticut, calcula que cerca de 26 mil millones de dispositivos estarán conectados a Internet para el año 2020. Otra consultora, ABI Research de Nueva York, cree que la cifra será de 30 mil millones, mientras que Cisco Systems, la firma de equipamiento para redes en California, espera que haya no menos de 50 mil millones. Cisco está tan embelesado del Internet de las Cosas que se ha instalado un "contador de conexiones" en su sitio web. El 11 de julio de 2014, el número de "cosas" conectadas al internet fue más de 12,8 mil millones y contando. (CISCO, 2014)

Una gráfica publicada recientemente por zdnet.com muestra el “boom” que ha logrado IoT. La línea de tendencia para las búsquedas de Google para el término "Internet of Things" (Fig. 1.1) se mantuvo esencialmente plana hasta mediados del 2013. Empezó a subir constantemente en octubre y luego se disparó en enero de 2014 (ZDNet, 2014). Parece haber alcanzado un pico con la adquisición de Google por $ 3.200 millones en enero de los laboratorios Nest, un negocio de emprendimiento o *start-up* de Silicon Valley que fabrica termostatos “inteligentes” conectados a Internet y detectores de humo / monóxido de carbono para el hogar. (Tech Crunch, 2014)

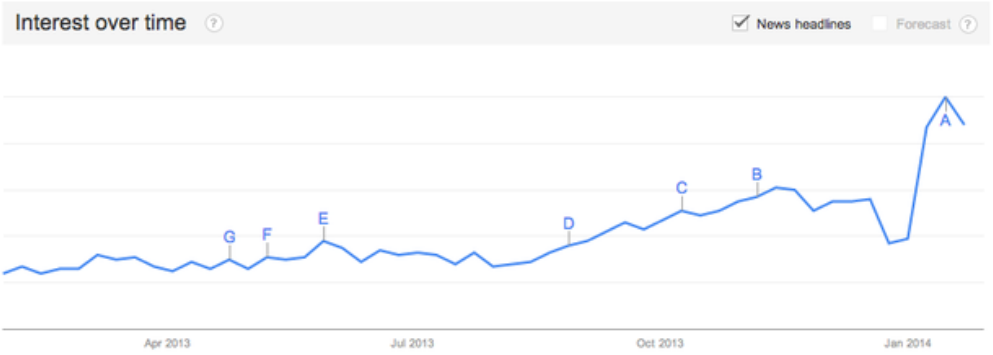


Figura 1.1. Tendencias en la búsqueda en Google para "*internet of things*". (Google Trends, 2014)

1.4. Interacción con el Internet

Las funciones básicas de un dispositivo o componente que pertenezca al IoT se describen como las siguientes: (Doukas, 2012)

- Colectar y transmitir datos: el dispositivo puede sentir el ambiente (hogares, cuerpo humano) y colectar información relacionada con él (temperatura e iluminación) y transmitirla a diferentes dispositivos (celular, computadora) hacia el Internet.
- Accionar dispositivos basados en disparadores de eventos o *triggers*: Se puede programar que accione otros dispositivos (apagar las luces o apagar la calefacción) basándose en condiciones establecidas por el usuario.
- Recibir información: procedente de la red a la que pertenecen o través del internet.
- Asistencia en comunicación: los dispositivos pueden asistir en la comunicación entre otros nodos de la misma red.

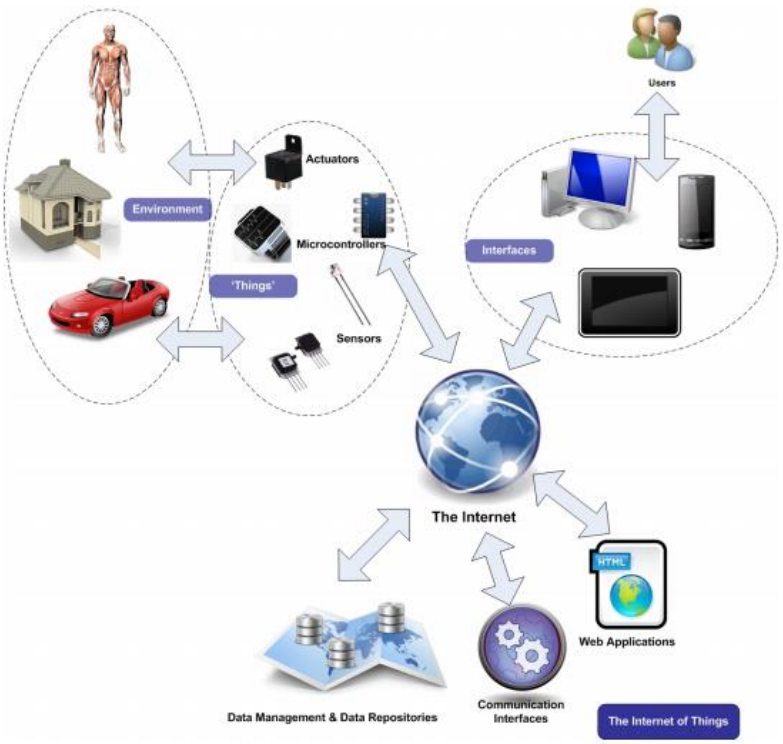


Figura 1.2. Ilustración del "Internet de las Cosas". Las cosas constan como sensores, actuadores que interactúan con el ambiente. El internet permite a los

usuarios gestionarlos y aprovechar sus datos sobre varias interfaces. (Doukas, 2012)

Se puede resumir a estos componentes en la siguiente ecuación simplificada:

Objeto Físico

+

Controlador, Sensor y Actuadores

+

Internet

=

Internet de las Cosas

Una ecuación para el Internet de las Cosas. (McEwen & Cassimally, 2014)

Una de las maneras más sencillas de conectar estos dispositivos al internet es utilizando una tarjeta Arduino, un hardware de código abierto u *open-source* utilizado para el desarrollo de prototipos de electrónica que incluye software, así mismo, *open source* (Arduino, s.f.). Con la Arduino se pueden crear estos objetos interactivos o distintos ambientes para permitir la comunicación entre ellos. La tarjeta Arduino se describe con mayor detalle en los capítulos siguientes

Lo que hace a los dispositivos IoT diferentes a aquellos con sensores ordinarios es básicamente la habilidad para comunicarse directa o indirectamente con el internet. Los sensores generan muchísimos datos que necesitan ser gestionados de cierta forma. En general, la memoria embebida es bastante limitada, por lo que se utilizan soluciones alternativas como almacenar datos en tarjetas de memoria, o en computadoras en casos en los que los sensores estén directamente conectados a ellas. Debido a que los sensores pueden ser integrados a dispositivos con mejores capacidades de red, la información se podría almacenar en línea. De esta forma, se puede solucionar el problema de la limitación de espacio de almacenamiento y, al mismo tiempo,

acceder a los datos desde cualquier lugar, cuando sea, y utilizando las aplicaciones web apropiadas.

La figura 1.3 ilustra las principales características de las “cosas” y su interacción con los servicios del internet. (Doukas, 2012)

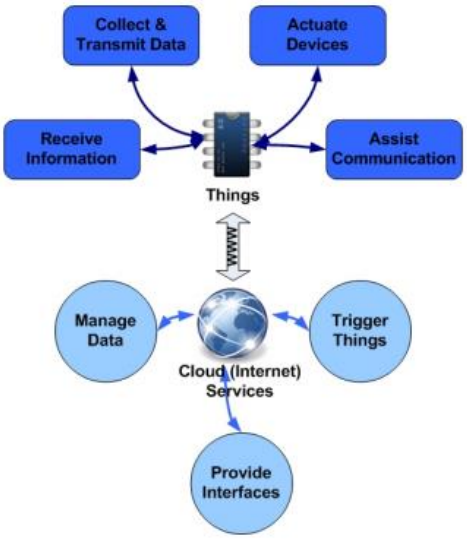


Figura 1.3. Ilustración de las características principales de las “Cosas” en redes IoT y los servicios de internet respectivos. (Doukas, 2012)

La mayoría de estas aplicaciones web proveen las interfaces para el intercambio de datos con otras aplicaciones y, en particular, son muy útiles las interacciones con aplicaciones móviles. Por lo que es posible que, por ejemplo, un iPhone o un teléfono Android pueda comunicarse directamente al dispositivo IoT. (Doukas, 2012)

También se pueden usar las plataformas web para enviar datos de regreso a los dispositivos. Los datos pueden ser disparadores condicionales, instrucciones para la activación de actuadores e interruptores, o simple información del internet, como el pronóstico del clima o una cuenta de red social que pueda ser mostrada en un interfaz como una pantalla LCD.

Las plataformas basadas en web que permiten la ya mencionada funcionalidad para la gestión de datos e intercambio de información, están

basadas en La Computación en la Nube o *Cloud Computing*. Más información sobre plataformas de la Nube se presentará en el capítulo siguiente.

1.5. Aplicaciones

En la cita inicial de este capítulo, McEwen y Hakim mencionaban a ciertos "objetos encantados", refiriéndose a cómo los dispositivos conectados, o cosas que, más allá de sus especificaciones funcionales, tienen capacidades de comunicación y procesamiento mucho mayores a la necesidad de una lámpara ordinaria, un paraguas o un espejo. Las aplicaciones de los componentes y dispositivos IoT pueden servir a satisfacer los requisitos planteados en situaciones como:

- Gestión de desperdicios
- Planificación urbana
- Sensores ambientales
- Domótica
- Herramientas digitales o *Gadgets* de interacción social
- Respuesta a emergencias
- Compras inteligentes
- Automatización en el hogar
- Muchas otras. (Alexandra Institute, 2011)

Como declara Charalampos Doukas en su libro, la imaginación es el límite. Y esa es otra razón por la que se considera que el Internet de las Cosas es algo realmente mágico.

1.6. De IT a IoT

Según la Cisco VNI (*Visual Network Index*), se predice el aumento del tráfico o de dispositivos a lo largo de algunos años. En la figura 1.4, se ve este rápido crecimiento, en el que actualmente existen cerca de 400 millones de dispositivos conectados al internet, y para dentro de 4 años, la cantidad sobrepasará las mil millones de unidades.

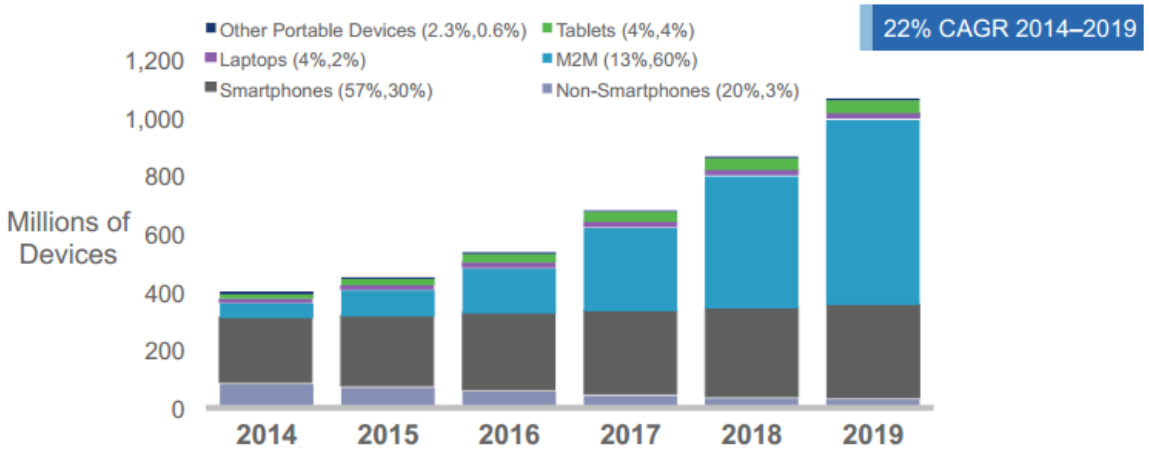


Figura 1.4. Índice de redes visuales. (Chiang, 2015)

Se observa que la mayor parte de dispositivos, en el año 2015, corresponden a los teléfonos inteligentes o *smartphones*, y una menor parte para dispositivos M2M (*Machine to Machine*) o máquina a máquina, los cuales son precursores del Internet de las Cosas. Para el año 2019, estas M2M dominarán, en cuyo caso no se requerirá de un ser humano en el intermedio del ciclo de comunicaciones entre los dispositivos.

1.6.1. I & T

Se puede definir el alcance lo que significan las letras “I” y “T” de IoT o internet de las cosas. Para “I” se puede extender más allá del concepto reducido del internet, se incluye también la red de dispositivos móviles, comunicaciones M2M directas que no utilizan la red troncal de internet o el *backbone* de internet. La palabra clave, realmente, sería “conectado.” Una conexión a una red de redes interoperable.

Las “cosas” corresponden a un amplio rango de dispositivos que tienen las siguientes capacidades, mejor definidas, o un subconjunto de estas:

- Comunicarse con el mundo exterior, a través de un puerto o una antena, operando a cierto valor de frecuencia.
- "Sentir" su ambiente mediante todo tipo de sensores (movimiento, velocidad, temperatura, humedad, etc.).
- Cambiar algo en su ambiente, actuadores (controladores embebidos).
- Realizar cómputos básicos o extensos (procesadores).
- Almacenamiento suficiente (memoria) para comunicar lo que ha captado mediante los sensores.
- Dependiendo de la aplicación, puede tener algún modo de interacción, como una pantalla táctil.

1.7.Definición de IoT por la ITU-T

La ITU-T define al internet de las cosas como una infraestructura de red global dinámica con capacidades auto configurables, basadas en protocolos de comunicación estándares e interoperables en donde las "cosas" físicas y virtuales:

- tienen identidades, atributos físicos y personalidades virtuales,
- utilizan interfaces inteligentes y
- se integran a la perfección en la red de información. (ITU-T, s.f.)

1.8.Arquitectura de IoT

Se puede ver a la arquitectura de IoT de diferentes maneras, una de las cuales es comenzar con aplicaciones potenciales de ciudades inteligentes, transporte inteligente, edificios inteligentes, consumo de energía inteligente, etc. Como se observa en la figura 1.5.

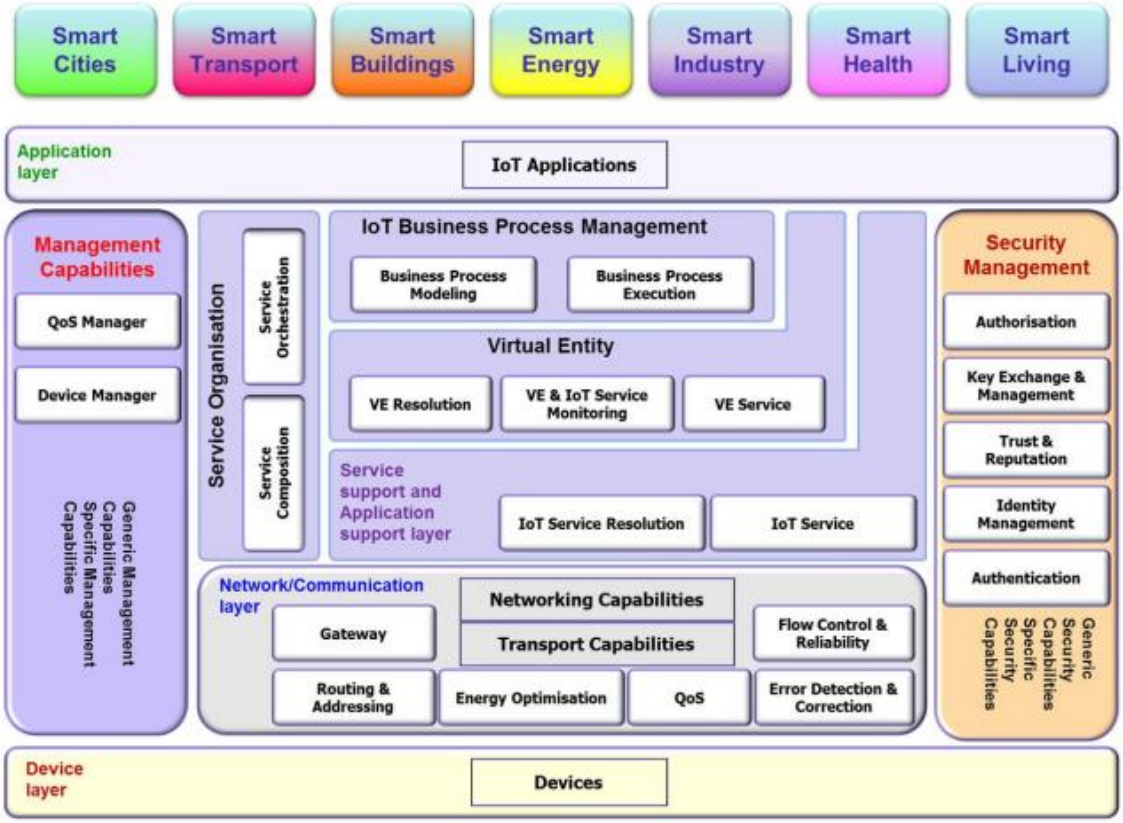


Figura 1.5. Modelo de arquitectura IoT de la ITU-T. (Chiang, 2015)

En cada bloque de aplicaciones potenciales se incluye la palabra "*Smart*" o "inteligente". Este término hace referencia, en este caso, a la capacidad de conectarse a una red. Para soportar algunas o todas estas capacidades, se necesitan aplicaciones específicas para las "cosas" (capa de aplicación) las cuales se construyen sobre una variedad de diseños, incluyendo capacidades de gestión, capacidades de seguridad y muchos otros aspectos.

Esto es solo un punto de vista en particular y existen muchos otros que están en trayectos paralelos, algunos coinciden y otros difieren en ciertos detalles. Las funciones que se encuentran debajo de la capa de aplicación, consisten en aquellas que se encargan de la práctica del problema, organización del servicio y de los datos. En contraste con aquellas que se encargan de las redes de comunicaciones, en la base de la arquitectura, se asienta la capa de dispositivos, las "cosas" o dispositivos físicos.

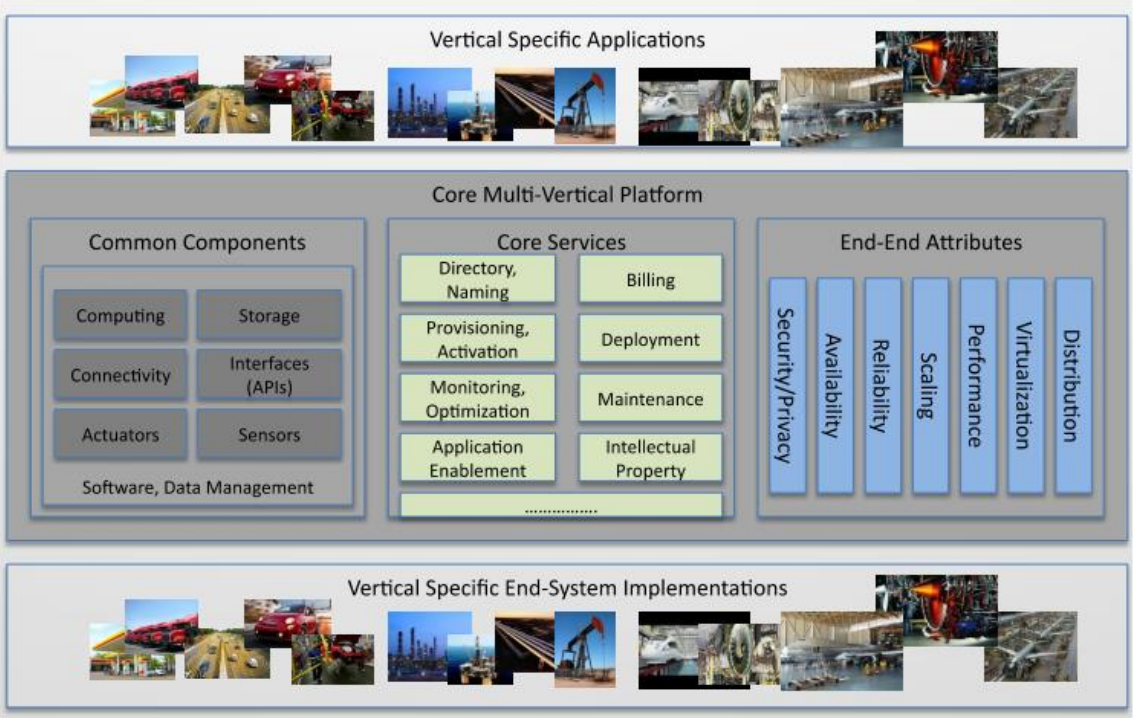


Figura 1.6. Otro modelo de arquitectura IoT. (Chiang, 2015)

En el otro modelo de arquitectura de la figura 1.6, uno más bien orientado a la parte comercial, también se tiene aplicaciones específicas verticales. Se puede observar una variedad de servicios que se ofrecen para soportar esas aplicaciones. En los componentes comunes se encuentran algunas de las características ya mencionadas, la habilidad de cómputo, almacenamiento, conectividad, etc.

También se pueden observar todo tipo de servicios. Por ejemplo, la capacidad de nombrar a estas cosas, facturar y cobrar por los servicios, proveer diferentes servicios, monitorearlos y optimizarlos.

Un tercer modelo, mostrado en la figura 1.7, se ilustra, de manera concéntrica, con diferentes sectores de servicios orientados a diferentes grupos de aplicaciones. Este modelo se orienta un poco más a las localizaciones en las que estas aplicaciones corren o se ejecutan. De manera similar a las anteriores, el modelo se asienta sobre los diferentes tipos de dispositivos, orientados a un servicio en particular. Por ejemplo, para los servicios de energía los dispositivos son turbinas, generadores, etc. Para el hogar y consumo son las consolas de juegos, lavadoras, etc. Y así para el resto de servicios.

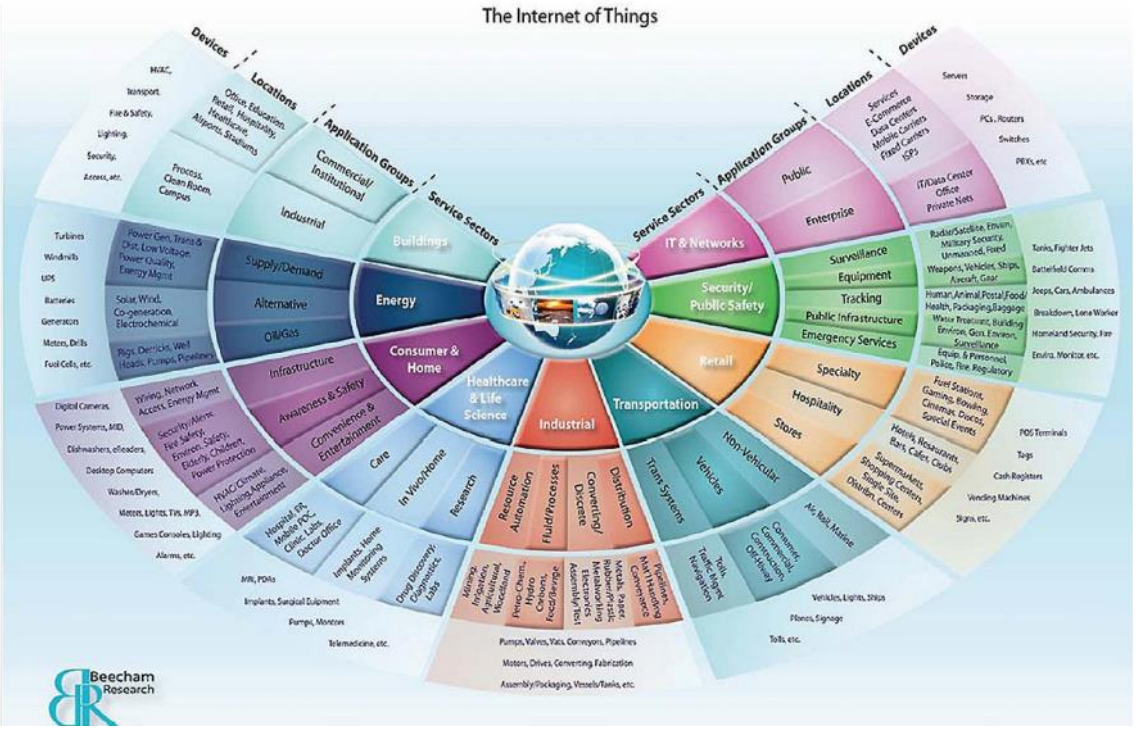


Figura 1.7. Otro modelo de arquitectura IoT. (Beecham Research, s.f.)

1.9. Clasificación de las aplicaciones de IoT

Las aplicaciones se pueden dividir en dos grupos: orientadas al consumidor y orientadas a la industria.

Orientadas al consumidor:

1. Dispositivos prendas o *Wearables*: dispositivos de realidad virtual, relojes, gafas y otros artículos de vestimenta.
2. Hogar y edificaciones.

Orientadas a la industria

1. Salud: hospitales, centros de salud, etc.
2. Transporte: vehículos públicos y privados.
3. Redes Inteligentes (*Smart grids*) y energía: fuentes de energía renovables, distribución, etc.
4. Municipal: vigilancia, parquímetros, recolección de desechos etc.
5. Ambiente y agricultura
6. Manufactura y logística

Sin importar cómo se clasifiquen estas aplicaciones de IoT, un aspecto clave que surge son sus necesidades heterogéneas, en todas las siguientes dimensiones:

- **Energía**: Las aplicaciones de IoT pueden tener requerimientos de energía muy distintos para los dispositivos. En términos de energía de transmisión y energía para mantener su operación. Algunas de estos pueden alimentarse con baterías e implantarse dentro del cuerpo humano donde se requieren protocolos IoT extremadamente eficientes para el manejo de la energía. En contraste con otros que solamente necesitan conectarse a un tomacorriente y el costo de la energía no representa un factor importante.
- **Retraso**: Algunas aplicaciones pueden ser críticas en términos del tiempo, en las que deben proveerse comunicaciones con latencias muy bajas y con muy poco retraso (que varía en el tiempo) o *jitter*. Mientras otras puedan tolerar retrasos de días hasta semanas e incluso meses. Se tiene entonces un gran rango que va desde los milisegundos a los meses.
- **Salida (*Throughput*)**: Se puede requerir una alta salida (en KBps) como un video para una cirugía remota, o una baja para registrar el valor medido por un sensor.
- **Costo**: Las aplicaciones pueden tener gran costo y tener implicaciones en las finanzas de una compañía mientras que otras pueden ser tan económicas que se pueden comprar en gran cantidad.
- **El ser humano:** Aunque se hable del internet de las cosas, todavía se requiere de una persona que realice el control y examine los datos. En otros casos, este proceso es automatizado y no existe una persona en el ciclo de trabajo. (Chiang, 2015)

Para el Profesor de la Universidad de Princeton de E.E.U.U., Mung Chiang, de ello se puede inferir que la necesidad heterogénea de las aplicaciones IoT representa un gran desafío para los investigadores de este campo y que para cada aplicación específica se debe realizar un análisis para establecer el nivel de cada una de estas necesidades o condiciones. Por ejemplo, la tabla 1.1 muestra aplicaciones en una ciudad inteligente y sus diferentes atributos.

	Medidores inteligentes	***Salud con apoyo de procesos electrónicos (eHealth)***	**Transporte inteligente (ITS)**	**Vigilancia**
Mobilidad	ninguna	Peatonal/Vehicular	Vehicular	Ninguna
Tamaño del mensaje	Pequeño (pocos Kb)	Medio (Diferentes casos)	Medio (Depende de versiones)	Grande
Patrón de tráfico de datos	Regular	Regular/irregular	Regular/irregular	Regular
Densidad de dispositivo	Muy alta (más de 10000 por celda)	Media	Alta	Baja
Requerimientos de latencia	Baja (hasta horas)	Media (segundos)	Muy alta (pocos milisegundos)	Media (<200 ms)
Requerimientos de eficiencia de energía	Alta (medidores alimentados por baterías)	Alta (dispositivos alimentados por baterías)	Baja	Baja
Confiabilidad	Alta	Alta	Alta	Media
Requerimientos de seguridad	Alta	Muy alta	Muy alta	Media

Tabla 1.1. Necesidades operativas para servicios IoT en una ciudad inteligente. (Chiang, 2015)

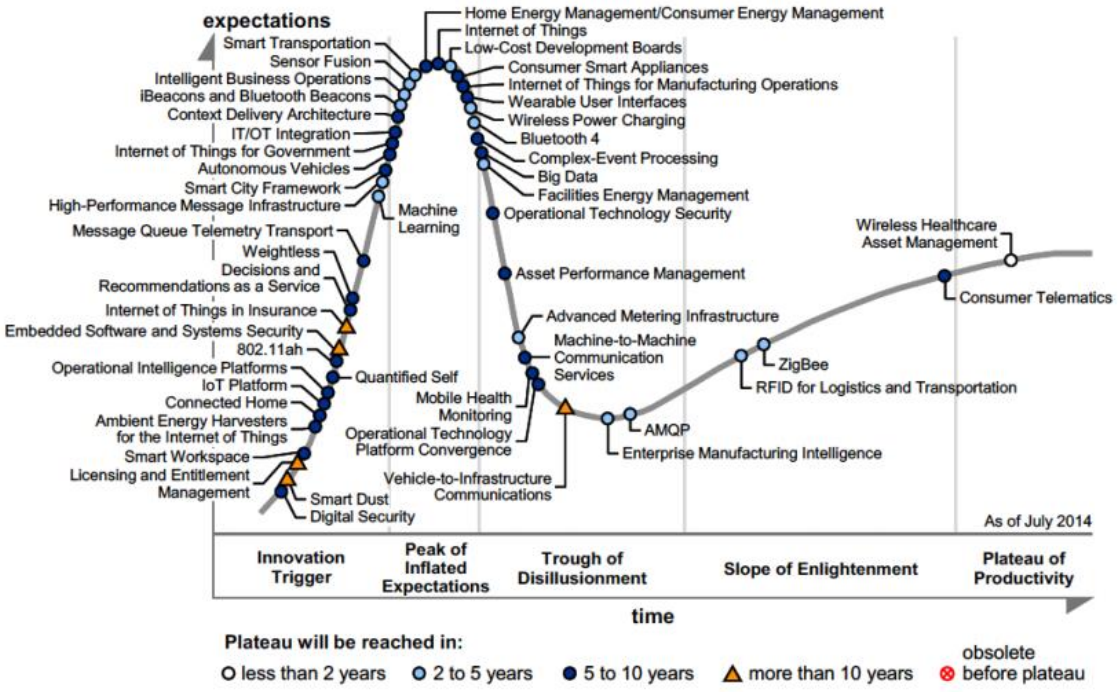


Figura 1.8. Gráfica del ciclo de sobre-expectativa del Internet de las Cosas. (Gartner, 2014)

La figura 1.8 muestra una gráfica realizada por la empresa consultora y de investigación de las tecnologías de la información, Gartner, correspondiente al mes de julio del año 2014, en la que se puede observar una representación visual lo que las personas en la industria llaman ciclo de sobre expectativa del internet de las cosas en los últimos años.

El típico ciclo de sobre-expectativa se desarrolla así: se comienza con poca popularidad y luego de algunos años todos están hablando del tema. En seguida, las personas notan que es algo difícil y quizás no sea tan beneficioso y que existen varios desafíos para su despliegue y entonces caen las expectativas y existe cierta desilusión. Luego de este periodo y, de una manera mucho más lenta que el disparo inicial de innovación y de la fase de inflación, se da un crecimiento firme o fase de productividad, con optimismo, pues no todos los ciclos de sobre-expectativa tendrán esta fase (Gartner, 2014). Se espera que IoT efectivamente la tenga, donde se estará a un nivel mucho más alto de donde se comenzó pero seguramente por debajo del pico de inflación.

IoT denota varias dimensiones. Cuando se hace la pregunta de en dónde se encuentra actualmente la gráfica, realmente se hacen varias de ellas,

dependiendo del tipo de servicios IoT al cual se refiere, ya que algunos de estos servicios están todavía terminando la fase de disparo, otros se encontrarán ya en el pico y otras estarán ya decayendo. Según la gráfica, muchos de los ítems tendrán un ciclo de sobre-expectativa de 2 a 5 o de 5 a 10 años y otros quedarán obsoletos antes de la fase de producción de este ciclo.

1.10. Capas de IoT

La figura 1.9 muestra las capas del Internet de las Cosas. En este caso, la manera en la que se definen las capas de trabajo es diferente a cómo se definen capas en la red comunicaciones en las que cada una sirve una función bien definida (capa física, de enlace, de red, de transporte, de aplicación). El término de capa se emplea en este aspecto para mostrar las separaciones de distinto modo de funcionamiento. Existe una capa de soluciones verticales que incluye ventas, de trasporte, industrial, etc. Y debajo de ella, están algunos servicios creados por IoT, como por ejemplo DasS (*Data as a service*). Más abajo, se tienen los servicios núcleo de IoT como configuración y gestión, análisis de los datos, aprovisionamiento de servicios y fijación de su precio, etc. A ésta le sigue la capa subyacente de red y de dispositivos IoT que incluyen todas las "cosas". (Chiang, 2015)

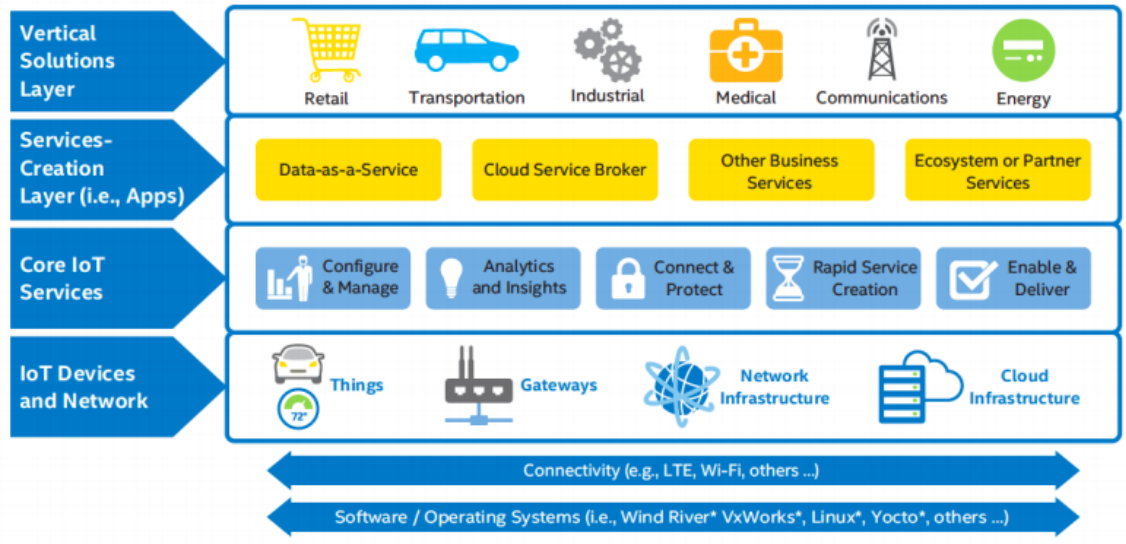


Figura 1.9. Capas y API de IoT. (Chiang, 2015)

Cada capa vertical está limitada por otra, los servicios proveídos por el módulo se ven reflejados en la capa superior. Existe una separación de intereses, es decir, una división del trabajo, y cada capa tendrá una visibilidad y un control limitados sobre lo que debería pasar en otra capa.

1.11. Necesidades IoT

Para que las aplicaciones se corran como IoT, hay varios conjuntos de necesidades que brevemente se clasifican en:

- Creación de redes
- Dispositivos
    - Hardware
    - Diseño UI/UX (interfaz de usuario/experiencia de usuario)
    - Prototipos: Raspberry Pi, Arduino, etc.
- Servicios
    - Aplicaciones
    - Economía y finanzas

1.11.1. Desafíos en la creación de redes de investigación para IoT.

El desarrollo de redes para el Internet de las Cosas requiere de investigación continua, pero se presentan los siguientes desafíos en este proceso:

1. **Escalabilidad Masiva**: El internet en sí es altamente escalable, pero en IoT se habla de una escalabilidad extremadamente sorprendente, varios órdenes de magnitud mayor de dispositivos conectados que seres humanos en el planeta.

2. **Alta heterogeneidad e Interoperabilidad**: No solo son varios dispositivos, sino también pueden ser muy distintos entre ellos.

Algunos pueden ser grandes, pequeños, con diferentes demandas de energía y a pesar de ello, deben ser interoperables.

3. **Capacidades limitadas en ciertas cosas**: Incluyendo un ancho de banda muy bajo, velocidades muy lentas, bajo poder de transmisión, limitada lógica de recepción, etc. Esta limitación de capacidades en ciertas cosas pueden restringir fuertemente el diseño de la arquitectura subyacente.

4. **Privacidad y seguridad**: Incluso antes de IoT, el internet ya ha tenido varios problemas con respecto a la seguridad-privacidad. Con IoT la cantidad de datos recolectados presenta un riesgo más alto para la privacidad, y como estas "cosas" pueden tener actuaciones, es decir, pueden controlar desde el mundo digital, al mundo físico y la seguridad puede manifestarse desde seguridad de información hasta seguridad física. Por ejemplo, secuestrar un auto conectado lo cual podría causar mucho daño.

5. **Robustez y resistencia**: Saber cuándo la red está bajo ataque, cuándo se dan *bugs* o errores de programación u otros tipos de errores humanos que podrían ocurrir. La red debería seguir funcionando con un grado de limitación, al detectar que algo no desempeña correctamente, para luego reparar y recuperarse rápidamente. Esto puede ocurrir tanto desde un punto de vista de la estructura de la red, como de los protocolos de control.

6. **De los datos a la acción**: Los datos solos no significan que se entiende lo suficiente de ellos como para transformarlos a inteligencia ejecutable, y con la velocidad que los datos fluyen en dispositivos IoT, que toman decisiones y acciones, muchas veces en tiempo real, su gestión puede llegar a ser muy dificultosa. (Chiang, 2015)

Las dificultades descritas corresponden específicamente al proceso de creación de redes, pero cabe recalcar que de manera similar, los dispositivos y servicios tendrán las suyas propias e igual de importantes.

1.12. Protocolos para IoT

Existe una gran cantidad de protocolos de comunicación que pueden aplicarse para el Internet de las Cosas. Algunos de los más empleados son los siguientes (MC Electronics, 2014):

- **Ethernet:** una red de área local (LAN) cableada con alta velocidad.
- **Bluetooth:** Velocidad mucho menor pero con una comunicación inalámbrica y de corto rango. Por ejemplo, el teléfono comunicándose con el automóvil.
- **802.11 WiFi:** Probablemente el protocolo de comunicación inalámbrica más común y fácil de encontrar en edificios, centros comerciales, universidades, etc.

- **Celular: 3G, 4G, LTE (*Long-Term Evolution*):** la red móvil o de celular, también inalámbrica pero con un rango de cobertura mucho mayor y con mejor soporte de movilidad. Para su despliegue debe pagarse por el uso del espectro radioeléctrico, a diferencia de WiFi que no tiene ningún costo por el uso de frecuencias. Algunos de los estándares de IoT dependen en gran medida de 4G y LTE y se han dado discusiones para hacerlos más livianos y permitan las comunicaciones entre las "cosas".
- **802.15.4:** Que corresponde a las redes de área personal (PAN) como ZigBee de la ZigBee Alliance, MiWi de la empresa Microchip, etc.
- **6LowPAN:** Redes con IP versión 6 de área personal de bajo poder y bajo consumo. Para poder desplegar dispositivos que pueden ser usados en la vestimenta o incluso dentro del cuerpo humano.
- **PLC**: Comunicaciones por líneas de poder (*Power Line Communication*).

- **NFC:** Comunicaciones de campo cercano (*Near Field Communication*).
- **RFID:** Identificación por radiofrecuencia (*Radio Frequency Identification*)
- Muchas otras.

Para el prototipo de aplicación de IoT de este trabajo, se utilizarán módulos Ethernet y WiFi, los cuales se explican en el capítulo del hardware.

La figura 1.10 presenta el gran conjunto de protocolos comunes, algunos ya mencionados, para las correspondientes capas de enlace, física, transporte, red, sesión y aplicación. Sobre estas capas se trata más sobre datos (procesamiento, almacenamiento) y los protocolos que se utilizan para ello. Y por último los protocolos empleados para los servicios o para el modelo de negocios y sus aplicaciones (*apps*).

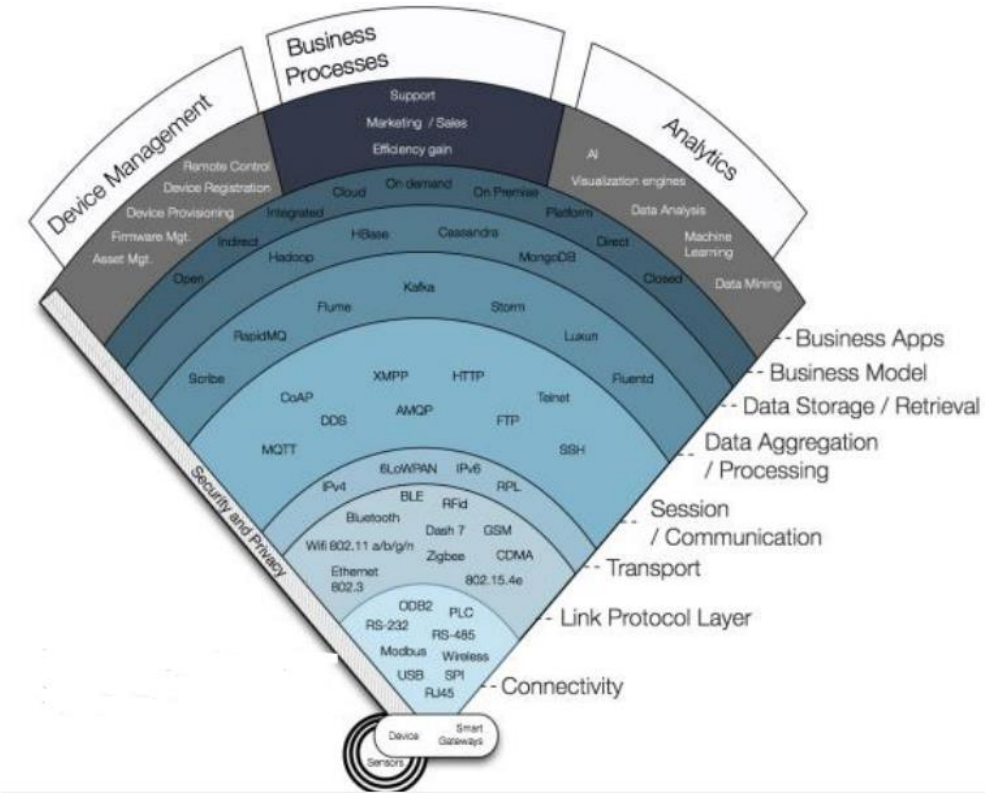


Figura 1.10. Distribución de protocolos para IoT desde dispositivos a procesos de negocio. (Chiang, 2015)

La figura 1.11 muestra muchas de las variantes de protocolos juntas, pudiendo existir entre las "cosas" una comunicación WiFi y luego de pasar por la puerta de enlace o *Gateway*, se accede a una LAN por medio de Ethernet. De igual manera se puede tener un enlace móvil con 3G o 4G y otros protocolos correspondientes.

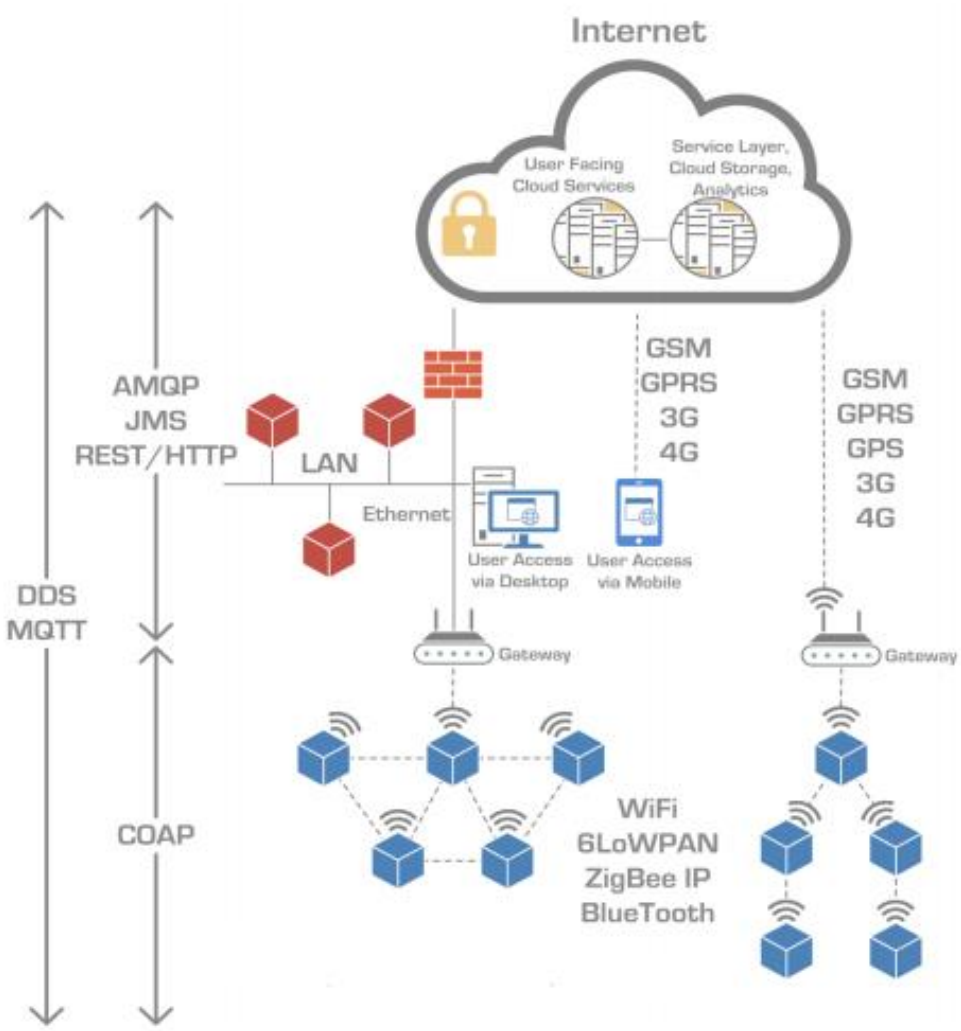


Figura 1.11. Protocolos de Conectividad IoT. (Chiang, 2015)

Conclusiones

En este primer capítulo se presentaron los conceptos, arquitectura, protocolos, necesidades y demás aspectos más importantes sobre IoT, los cuales sirven como base para el desarrollo de aplicaciones de este tipo. Conjuntamente con la información de los capítulos siguientes, se tendrá una idea clara de cómo diferentes tecnologías se complementan entre sí para formar el Internet de las Cosas.

2. CAPÍTULO 2: Computación en la Nube

Introducción

Un aspecto fundamental del Internet de las Cosas (*Internet of Things*) es la conexión cercana con las plataformas de Computación en la Nube. Las aplicaciones IoT están en realidad basadas en Computación en la Nube de manera que sea posible proveer características de almacenamiento con mejor escalabilidad y gran interoperabilidad a través de accesos abiertos e interfaces directas para la comunicación e intercambio de información.

Este capítulo proporciona una visión básica sobre lo que es la Computación en la Nube, cómo está construida y cómo los usuarios pueden construir aplicaciones que se comuniquen con los sistemas *Cloud*. Finalmente, el capítulo revisa algunos de los más populares servicios *Cloud* que permiten a los usuarios manejar información de sensores y construir redes IoT.

2.1. Definición

Computación en la Nube o *Cloud Computing* es la entrega de información como un servicio en lugar de un producto, por lo cual se proveen los recursos, software e información compartidas a los computadores y a otros dispositivos como si fueran un servicio básico (como la red eléctrica) sobre una red (generalmente el internet). (Armbrust, y otros, 2009)

La Computación en la Nube ofrece varios servicios, como los de computación, software, acceso a información y almacenamiento, los cuales no requieren de conocimiento del usuario sobre la ubicación física y la configuración del sistema que entrega los servicios. La figura 2.1 ilustra el concepto de Computación en la Nube y la comunicación entre los servicios (como aplicaciones, almacenamiento de información y recursos de procesamiento) y todo tipo de computación y dispositivos de comunicación.

La Computación de la Nube significa tener cada pieza de información que se necesita para cada aspecto de la vida diaria al alcance y lista para su uso. La

información puede ser móvil, transferible y asequible al instante. La clave para tener un comportamiento portátil e interactivo, es la capacidad para sincronizar la información entre dispositivos, así como para acceder a información compartida. La información compartida es aquella a la que se accede en línea en cualquier lugar, tal como redes sociales, bancos, blogs, salas de prensa, comunidades pagadas, etc. (Hardy, 2014)

Figura 2.1. Una ilustración del concepto de la Computación en la Nube. Todo tipo de dispositivos de cómputo y comunicación pueden interactuar con la Nube y compartir los mismos recursos de información. Los dispositivos y servicios IoT son, de tal manera, una parte de la Nube. (Doukas, 2012)

2.2. Breve historia

El nacimiento y ascenso de la Computación en la Nube está estrictamente relacionado con un tipo de tecnología llamado “Virtualización”. Ésta se origina de la investigación de IBM en los años sesenta cuando la compañía comenzó a desarrollar computadoras de usuarios múltiples de tiempo compartido. La principal meta, en ese entonces, era permitir a varios usuarios compartir los mismos recursos de computación en una sola plataforma al mismo tiempo.

Con los avances de las arquitecturas de CPU, se permitió que muchos más procesos sean ejecutados simultáneamente y que se realizara una mejor gestión de procesos. La virtualización permite a los usuarios crear y trabajar en varios ambientes virtuales en la misma máquina. El software apropiado era también desarrollado para manejar instancias virtuales, compartir el CPU y recursos de memoria de manera apropiada, y hacerse cargo de asuntos como el mantenimiento de sistemas. (Armbrust, y otros, 2009)

2.3. Arquitecturas y Servicios Básicos

Un servicio *cloud* o de la nube tiene tres distintas características que lo diferencias de las aplicaciones tradicionales y servicio de *hosting*. Se lo provee a demanda, es "elástico" (lo que significa que el usuario puede tener tanto o tan poco de un servicio como lo requiera en un momento dado) y finalmente, el servicio es gestionado totalmente por el proveedor. A continuación se describen los componentes principales y los modelos de la Computación en la Nube. (EDUCASE, s.f.)

2.3.1. Componentes de la Computación en la Nube

Un servicio de computación en la nube (como un servicio de almacenamiento en línea) consiste principalmente de tres componentes básicos: la aplicación, la plataforma y la infraestructura. (Doukas, 2012)

La Aplicación de la Nube es la principal interfaz con los usuarios. Esta puede ser una aplicación única (por ejemplo, un servicio de almacenamiento) o incluso un grupo de aplicaciones (como *Google Docs*) ejecutados e interactuados vía un navegador web, un escritorio local o un cliente remoto. Los usuarios no necesitan preocuparse de comprar e instalar ningún software (o incluso el sistema operativo). En su lugar, pagan el servicio en función de su uso.

La Plataforma de la Computación en la Nube (en su mayoría los componentes apropiados de software que se utilizan) maneja cualquier tipo de operación de mantenimiento necesaria (incluyendo la activación e integración de nuevos recursos de hardware a la aplicación del usuario). Esta puede considerarse como la capa intermediadora entre las aplicaciones alojadas en la nube y la infraestructura.

La Infraestructura de la Nube se refiere a la capa de hardware e incluye todos los componentes computacionales esenciales, de almacenamiento y de red que permiten la virtualización de los recursos y que permiten que funcione la plataforma de cómputo y provea los servicios correspondientes a los usuarios.

2.3.2. Modelos y Arquitecturas de la Computación en la Nube

Hay varios tipos de nube disponibles, basados en la disponibilidad de sus recursos y en sus usos. Es decir, hay nubes públicas, comunitarias, híbridas y privadas. (EDUCASE, s.f.)

- **Nube pública:** las nubes públicas describen a la computación de la nube en el sentido tradicional, donde los recursos son ofrecidos dinámicamente al público por internet, vía aplicaciones web / servicios web, desde un proveedor que factura en base de servicios de computación.
- **Nube Comunitaria:** las nubes comunitarias comparten infraestructuras entre varias organizaciones de una comunidad específica con asuntos en común (seguridad, conformidad, jurisdicción, etc.), ya sea manejado internamente o por un tercero y alojado interna o externamente. Los costos se distribuyen entre menos usuarios que en una nube pública (pero más que en una nube privada), por lo que solo se consiguen algunos de los beneficios de la computación de la nube.

- **Nube híbrida:** la nube híbrida es una composición de dos o más nubes (privada, comunitaria o pública) que mantiene entidades únicas pero que están juntas, ofreciendo los beneficios de múltiples modelos de despliegue.
- **Nube privada:** opera su infraestructura solamente por una misma organización, ya sea manejada internamente o por un tercero y se aloja interna o externamente.

En cuanto a las arquitecturas y tipos de los servicios que ofrece un proveedor de la nube, principalmente hay tres tipos de servicios: (Santos, Gummadi, & Rodrigues) (EDUCASE, s.f.)

1. **Infraestructura como un servicio:** los proveedores de servicios se encargan de todo el costo del hardware esencial (servidores, equipamiento de red, almacenamiento y respaldos). Los usuarios solamente tienen que pagar para tener el servicio de computación. Y los usuarios construyen y manejan su propia aplicación de software. La infraestructura de Amazon EC2, es un buen ejemplo de este tipo de servicio.

2. **Plataforma como un servicio-servicio**: en este caso los proveedores, aparte del hardware, también proveen una plataforma específica o un conjunto de soluciones para el usuario (por ejemplo, un servidor de aplicaciones). Esto ayuda a los usuarios a ahorrar inversiones en hardware y software. El motor de aplicaciones de Google (*Google APP Engine*) y *Jelastic* proveen este tipo de servicio. El gran beneficio de este servicio es que los usuarios necesitan enfocarse en el desarrollo y despliegue de su aplicación y no se preocupan de la configuración del sistema operativo y del servidor de la aplicación.

3. **Software como un servicio**: los proveedores de servicio usan su propio hardware e infraestructuras de plataforma para proveer a los usuarios aplicaciones de software específicas, como almacenamiento en línea, e-mail, aplicaciones de documentos, etc. *Google Docs* y *Evernote* son

representantes de aplicaciones de software como servicio basadas en la nube.

2.3.3. Beneficios de la Computación en la Nube

Entonces, ¿Por qué resulta necesaria la computación en la nube? y, ¿Por qué se busca almacenar datos de sensores en infraestructuras basadas en la nube? A continuación se describen algunas de las características y beneficios más importantes de la computación de la nube (Doukas, 2012) (Armbrust, y otros, 2009):

- Eliminación o reducción de costos iniciales por el servicio de alojamiento web o *hosting*: con las plataformas de la computación de la nube, no se necesita instalar uno mismo la infraestructura de hardware apropiada (que incluye servidores, *routers* de red, *firewalls*, etc.) para alojar una aplicación en línea y almacenar datos de sensores.

- Costos administrativos reducidos: no se necesita gastar tiempo ni dinero en la administración (por ejemplo, configurar cosas, cuidar los respaldos, etc.) de los sistemas de software y hardware para la plataforma.

- Mejor utilización de recursos: en caso de que se requiera construir un servicio y proveerlo a otros usuarios; no es necesario dedicarse a construir sistemas (servidor web, servidor de base de datos) para cada servicio. Utilizando la tecnología de la nube, se puede reutilizar mucho del hardware existente para alojar varios servicios, incluso si es que éstos demandan diferentes recursos como sistemas operativos alternativos.

- Escalabilidad demandada: ¿cuánto almacenamiento o poder de procesamiento se necesita? ¿Se podría comenzar con un par de sensores para el monitoreo del hogar y luego invitar a otros usuarios a utilizar la misma plataforma y así expandirse a varias docenas de sensores? No es necesario extender el almacenamiento o los recursos de un servidor web propio; el proveedor del servicio de nube lo hace automáticamente.

- Implementación rápida y fácil: no se requiere conocer a profundidad sobre la configuración y despliegue de una aplicación basada en la web, instalar o configurar un servidor web, el sistema de la base de datos, hacer las conexiones apropiadas ni configurar varias instancias para cumplir ciertos requerimientos de escalabilidad. El usuario solamente se concentra en la aplicación e información que necesita para ser alojada en la nube.

- Calidad de servicio y disponibilidad garantizada: el proveedor de servicio de nube siempre garantiza la disponibilidad de hardware y software. No hay que preocuparse por fallas de hardware, cortes de energía ni conexiones de red del lado del servidor.

- Acceso en cualquier lugar: lo más importante, los recursos de la nube (aplicaciones web, elementos de almacenamiento) pueden ser accedidos desde cualquier tipo de dispositivo computacional que tenga acceso al internet (incluyendo plataformas de microcontrolador, como Arduino).

- Recuperación de desastre/respaldos: no hay que preocuparse de respaldar información o crear mecanismos de recuperación. El proveedor se encarga de todo.

2.4. Comunicación con la Nube utilizando Servicios Web

Como se describió anteriormente, el acceso a servicios en la nube debe ser fácil, directo, abierto, e interoperable, lo que significa que se deben dar los medios de comunicación y las interfaces de programación (APIs) de modo que sean fáciles de implementar en cada plataforma y ambiente de desarrollo.

En la práctica, la manera más abierta e interoperable para proveer acceso a servicios remotos y /o permitir aplicaciones para comunicarse entre ellas, es mediante el uso de servicios web. Existen servicios que se ofrecen en la web, basados en arquitecturas de comunicación específicas, y en estándares de tecnología. Las características clave de los servicios web, consisten en que sean fáciles de construir, desarrollar y reutilizar. De hecho, la principal razón para que alguien decida construir un servicio web, es para exponerlo a otros desarrolladores, para que ellos a la vez, puedan integrarlos en sus propias aplicaciones.

Los servicios web pueden también verse como bloques interoperables para el desarrollo de aplicaciones personalizadas. Un servicio web se identifica generalmente por un URI (Identificador de Recursos Uniforme), como una dirección web. Un servicio web tiene definiciones WSDL (Descripción de Lenguaje de Servicios Web) Éstas son descripciones computarizadas de lo que el servicio web puede hacer, donde se localiza y cómo puede utilizarse (o "consumirse") mediante la aplicación del cliente. Para comunicarse con un servicio web, es necesario usar mensajes SOAP, los cuales se basan en XML, se transportan sobre protocolos de internet como HTTP, SMTP y FTP. (Doukas, 2012)

Los servicios web poseen ciertas ventajas sobre otras tecnologías (Doukas, 2012):

- Los servicios web son independientes de plataformas y de lenguaje ya que ellos utilizan lenguajes XML estándar. Esto significa que el programa del cliente puede estar en C++ y correr en Windows, mientras que el servicio web está programado en java y corriendo en Linux.

- La mayoría de los servicios web utilizan HTTP para transmitir mensajes (así como el servicio de solicitud y respuesta). Esto es una gran ventaja si se quiere construir una aplicación de escala de internet, ya que la mayoría de los *proxies* y *firewalls* de internet no interfieren con el trafico HTTP (a diferencia de *CORBA*, el cual normalmente tiene problemas con *firewalls*).

Algunos términos importantes al hablar de servicios web (Abubakr, 2012) :

- **Procesos de servicio**: esta parte de la arquitectura generalmente involucra más de un servicio web. Por ejemplo, un servicio conocido como "el descubrimiento" pertenece a esta parte de la arquitectura, ya que permite localizar un servicio particular de entre un conjunto de servicios web.

- **Descripción de servicio:** una de las características más importantes de los servicios web es que estos son auto-descriptivos. Esto significa que una vez que se ha localizado el servicio web, se puede pedir que éste se "describa a sí mismo", y como respuesta, indique las operaciones que soporta y cómo se las puede solicitar. Esto se maneja mediante el lenguaje de descripción de servicios web (*WSDL*).

- **Invocación de servicio:** invocar (o consumir) un servicio web involucra pasar mensajes entre el cliente y el servidor. SOAP (Protocolo de Acceso a Objetos Simples) específica cómo se deben diseñar solicitudes al servidor y como el servidor debe diseñar estas respuestas. En teoría, se puede usar otro lenguaje de solicitud de

servicio (tal como XML-RPC, o incluso algún lenguaje XML *ad hoc*) sin embargo, SOAP es la elección más generalizada para servicios web.

- **Transporte:** finalmente, todos estos mensajes deben transmitirse de alguna manera entre el servidor y el cliente. El protocolo de elección para esta parte de la arquitectura es HTTP (*Hipertext Transfer Protocol*), el mismo protocolo que es usado para acceder a páginas web convencionales en internet. También, en este caso, se pueden usar otros protocolos pero HTTP es el más utilizado.

2.4.1. Servicios Web SOAP y RESTful

Actualmente existen dos opciones para el desarrollo de servicios web: el tradicional, basado en estándares, SOAP y el conceptualmente más simple y nuevo, REST.

2.4.1.1. SOAP

Simple Object Application Protocol (Protocolo de Aplicación de Objetos Simples) ha sido uno de los protocolos de implementación iniciales para la provisión de servicios web. Se basa en el intercambio de mensajes XML con un formato especifico que describe los servicios (por ejemplo, métodos de programación) que se proveen (esta parte se llama Lenguaje de Descripción del Servicio Web, o WSDL por sus siglas en inglés) y los datos (por ejemplo, lecturas de sensores) comunicados entre aplicaciones (Mueller, 2013).

La estructura básica de un mensaje SOAP sigue la lógica HTML incluye un encabezado (opcional) y un cuerpo (obligatorio):

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header>
   <!-- Información de encabezado aquí -->
   </env:Header>
          <env:Body>
          <!-- Cuerpo o "carga" aquí -->
          </env:Body>
</env:Envelope>
```

2.4.1.2. REST

Representational State Transfer, (Estado de Transferencia Representativo). A diferencia de SOAP, se puede obtener los contenidos del servicio usando una solicitud HTTP GET, borrar o actualizar, usando las acciones POST, PUT o DELETE respectivamente. Es como si la aplicación que se comunica con servicios web basados en REST utilice lo que HTTP defina para comunicar e intercambiar información.

Es un conjunto de recursos con cuatro aspectos definidos (Mueller, 2013):

- La URI base para los servicios web, tales como http://example.com/resources/
- El tipo de datos de contenido de internet soportados por el servicio web. Comúnmente son JSON, XML o CSV pero puede ser cualquier otro tipo válido de contenido de internet.
- El conjunto de operaciones soportado por el servicio web utilizando métodos HTTP (por ejemplo, POST, GET, PUT o DELETE).
- La API debe manejarse mediante hipertexto.

Un servicio que obtenga los detalles de un usuario llamado Arduino, por ejemplo, debe manejarse utilizando un HTTP GET a http://example.org/users/arduino. Borrar el usuario con HTTP DELETE, y crear un nuevo usuario se haría probablemente con un POST. La necesidad de poner referencias de otras fuentes debería manejarse usando

hipervínculos (*hyperlinks*, el equivalente XML al href de HTTP, el cual es el xlink:href de XLinks) y separar respuestas de solicitudes HTTP.

Los servicios web RESTful están también mucho más cercanas en diseño y filosofía a la Web en sí.

2.5. Computación en la Nube y el Internet de las Cosas

Es claro que la Computación en la Nube provee grandes beneficios para aplicaciones alojadas en la web que también tienen requerimientos computacionales y de almacenamiento especiales. Además, ofrece maneras fáciles de acceder a ellas desde las redes y permite a los usuarios enfocarse en el desarrollo de aplicaciones. Por lo tanto, es bastante razonable que la mayoría de las plataformas IoT existentes dependan de infraestructuras en la nube. Éstas ofrecen la misma experiencia al usuario al estar usando/construyendo o desplegando aplicaciones en la nube. Los usuarios no tienen que preocuparse del mantenimiento de aplicaciones, recuperación, escalabilidad ni de asuntos de red. Ellos solamente necesitan configurar sus sensores para enviar información a través de plataformas IoT (Doukas, 2012).

La siguiente sección presenta los servicios abiertos, basados en la nube, más difundidos para construir redes IoT y sus aplicaciones.

2.5.1. Lista de plataformas disponibles

El número de plataformas disponibles que caracteriza el registro de datos en la nube para dispositivos de sensor está creciendo continuamente. Existe un gran número de ellas que la presentación de cada una podría representar un trabajo totalmente nuevo, pero que valdría la pena explorar. La siguiente lista hace un resumen de algunas de plataformas IoT y en el Capítulo 6 se describirán a mayor detalle dos de ellas.

Nombre	Características	Free | Open Source	URL
ThingSpeak	Visualiza y almacena datos de sensores en línea	Sí | Sí	http://www.thingspeak.com
Nimbits	Registro de datos en la nube	Sí | Sí	http://www.nimbits.com
iDigi	Plataforma en la nube para dispositivos	Sí | No	http://www.idigi.com
SensorCloud	Visualiza y almacena datos de sensores en línea	Sí | No	http://www.sensorcloud.com
Open.Sen.Se	Plataforma para el Internet del Todo	Sí | No	http://open.sen.se
Exosite	Plataforma y Portales de datos basados en la nube y gestión de dispositivos	Sí | No	http://www.exosite.com
EVRYTHNG	Motor de Software y Plataforma de la Nube	Sí | No	http://www.evrythng.net
Paraimpu	Herramientas sociales para las cosas	Sí | No	http://paraimpu.crs4.it
Manybots	Recolecta y gestiona información desde varios dispositivos	Sí | No	http://www.manybots.com
Lelylan	Orientado en automatización del hogar y monitoreo	Sí | No	http://www.lelylan.com

Tabla 2.1. Visión general de las plataformas Cloud IoT. (Doukas, 2012)

Conclusiones

La Computación en la Nube es el componente fundamental de las redes y aplicaciones IoT. El mayor beneficio de este concepto es la gestión y el almacenamiento de información en línea.

El mejor modo de comunicar las aplicaciones propias con infraestructuras de nube, es mediante servicios web y específicamente, servicios web RESTful, los cuales son muy sencillos de programar y livianos de usar en microcontroladores y otros dispositivos.

3. CAPÍTULO 3: Red de Sensores Inalámbricos (WSN)

Introducción:

En este capítulo se hablará sobre la red de sensores inalámbricos o WSN (*Wireless Sensor Network*) donde se analizarán su arquitectura, funciones, modos de ponerla en práctica, cuáles son sus equipos y componentes y las razones por las cuales son una buena opción al momento de crear un sistema de monitoreo; muchas de las aplicaciones que utilizan IoT necesitan de una red WSN.

3.1. Definición

Una WSN está conformada de nodos, Gateway y software, los nodos interactúan con los sensores y mediante un módulo instalado en ella, es posible la configuración del nodo para manejar su comportamiento y monitoreo; lo cual es posible mediante los nodos programables.

Dicho esto, se puede determinar que una WSN está constituida por sistemas autónomos desplegados de forma densa y aleatoria para recolectar información sobre un fenómeno en particular como detección de eventos como incendios, lluvias, inundaciones, etc. Además de mediciones periódicas como agricultura de precisión, aproximación de una función como temperatura y tiempo. (National Instruments, s.f.)

Una red WSN se diseña con el propósito de monitorear un lugar o equipo en específico, mediante una infraestructura de nodos, con la finalidad de control o análisis local de cada una de estas redes. Se puede armar con una infinidad de nodos para su uso dependiendo del tipo de trabajo que se le quiera dar. Los nodos se pueden comunicar entre sí, por cableado o por conexión inalámbrica. En la Figura 3.1. Se puede observar cuáles son los participantes de una Red WSN.

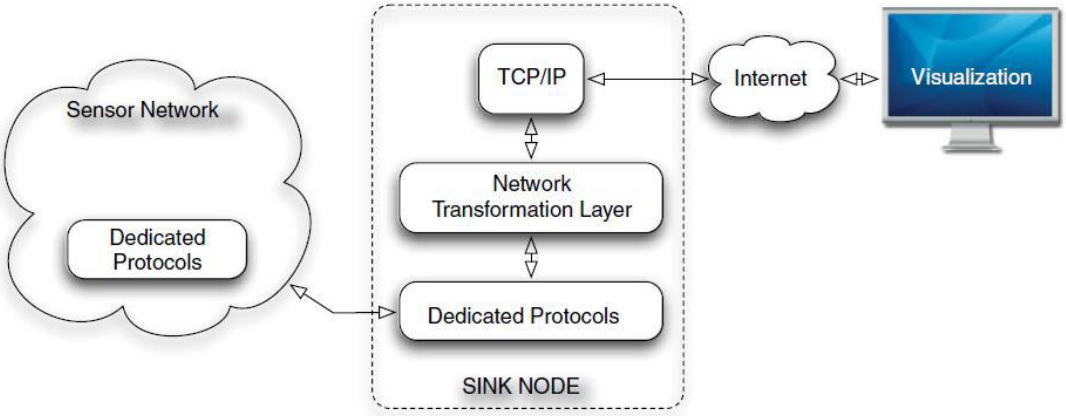


Figura 3.1. Participantes de una WSN. (Samaniego Armijos, 2015)

A una red WSN se la puede desplegar de diferentes formas, la primera es por modo aleatorio donde se puede esparcir los nodos por medio aéreo y así crear la red; el despliegue regular por ubicación planificada y nodos móviles donde se puede complementar el despliegue deficiente o con movimiento pasivo. (National Instruments, 2015)

En la implementación de la red se puede obtener los siguientes servicios:

Medio Ambiente:

- Detección de fuego en bosques.
- Mapa de biocomplejidad del ambiente (monitoreo de árboles).
- Detección de fluidos (lluvias, nivel de aguas en ríos y sensores de clima).
- Agricultura de precisión (monitorear el nivel de pesticidas, nivel de erosión del suelo y contaminación ambiental).

Salud:

- Tele-monitoreo de datos psicológicos humanos.
- Monitorear el comportamiento de personas adultas mayores, en una caída.
- Determinar el comportamiento del corazón, presión sanguínea.
- Localizar y monitorear pacientes y doctores dentro de un hospital.
- Administración de fármacos en un hospital.

- Identificación de alergias mediante sensores, determinando la medicación requerida de los pacientes.

Residencial y Comercial:

- Automatización de residencias y ambientes inteligentes.
- Colocación de sensores en microondas, cocinas, refrigeradoras, para interactuar con las redes externas.
- Museos interactivos
- Control de inventario
- Detección y localización de vehículos
- Detectar y monitorear vehículos robados

El *Gateway* recibe los datos obtenidos de manera inalámbrica para operar independientemente, conectándose al sistema donde el usuario puede obtener la información deseada, procesar y analizar los datos medidos usando software.

3.2. Factores a considerar en el Diseño:

- Tolerancia a fallos
- Escalabilidad
- Limitaciones del hardware
- Topología de red de sensores
- Entorno de operación
- Medio de transición
- Consumo de energía
    - Dentro del consumo de energía es necesario considerar los siguientes factores:
    - Comunicación: Máximo consumo de energía
    - Protocolos: MAC y enrutamiento

- Procesamiento: se hacen paquetes pequeños
- Sensores: Consumos esporádicos gastan menos energía que el consumo constante.

3.3. Arquitectura De Protocolos

- Combina energía y enrutamiento.
- Integra protocolos de *networking* o de creación de redes.
- Comunica eficientemente por medios inalámbricos.
- Promueve el trabajo colaborativo entre nodos.

En la Figura 3.2. Se puede apreciar las capas de la Arquitectura de Protocolos.

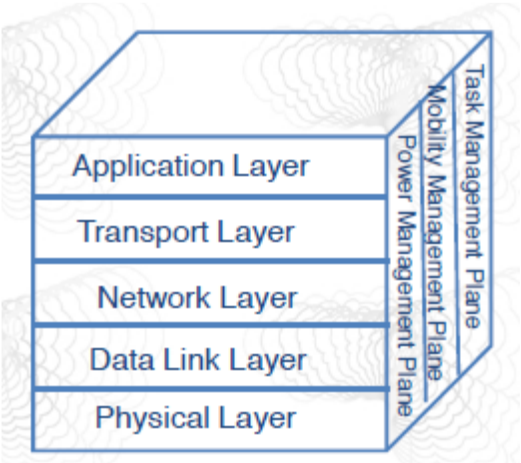


Figura 3.2. Capaz de Arquitectura de protocolos. (Samaniego Armijos, 2015)

Para poder comprender un poco más sobre el sistema *Gateway* y su funcionamiento dentro de una red WSN, el *Gateway* es el encargado de recibir los datos medidos desde los nodos distribuidos y desde los puentes a la red empresarial. En otras palabras, es el coordinador de red encargado de la autenticidad del nodo y almacenamiento de mensajes, aquí se reúne, analiza y presenta los datos de medida. En una red WSN pueden existir múltiples *gateways* donde cada uno se conecta y se comunica en un canal inalámbrico independiente y con la respectiva configuración de software, los cuales son monitoreados para obtener los resultados solicitados.

3.4. Tipos de Gateways

3.4.1. Los *Gateways* Programables

Figura 3.3. Gateway Programable. (National Instruments, 2015)

Administran la red inalámbrica y datos agregados, todos los *gateways* tiene un radio IEEE 802.15.4 de 2.4 GHz y con esta frecuencia el *Gateway* es capaz de comunicarse incluso con 8 nodos finales o caso contrario treinta y seis nodos de medida WSN utilizando una topología tipo malla. . En la figura 3.3 se observa un modelo de este dispositivo.

Todos los *gateways* tienen diferentes tipos de conectividad. Existen, por ejemplo, *gateways* que crean sistemas de WSN en tiempo real, otros que creando un sistema de embebido portátil, operan independientemente de una máquina principal, y por último, sistemas Gateway, que se pueden añadir fácilmente a una red WSN inalámbrica para realizar los monitoreos y controles.

3.4.2. *Gateway* Tipo C:

Orientado a la creación sencilla de sistemas completos de medidas de control (tanto como los sistemas inalámbricos y cableados), este tipo de Gateway soporta hasta 36 nodos de WSN. En la figura 3.4 se observa un modelo de este dispositivo.

Figura 3.4. Gateway Tipo C. (National instruments, 2015)

3.4.3. *Gateway* Ethernet:

Coordina la comunicación entre nodos de medida distribuidos y el controlador principal, el software incluido con este tipo de *Gateway* ayuda a configurar rápidamente la red de sensores y extraer, analizar y presentar los datos medidos usando el entorno de desarrollo gráfico LabVIEW. En la figura 3.5 se observa un modelo de este dispositivo.

Figura 3.5. Gateway Ethernet. (National Instruments, 2015)

3.5. Esquema de una Red WSN

La figura 3.6 muestra un esquema sencillo de una WSN, la cual está conformada por procesadores o placas con comunicación por radio, y que en este contexto se denominan *motes* o motas (M). Además de sensores y hardware para adquisición de datos (S), y la interfaz de red o puerta de enlace, *Gateway*. El conjunto de la mota con los sensores forma un nodo de la WSN.

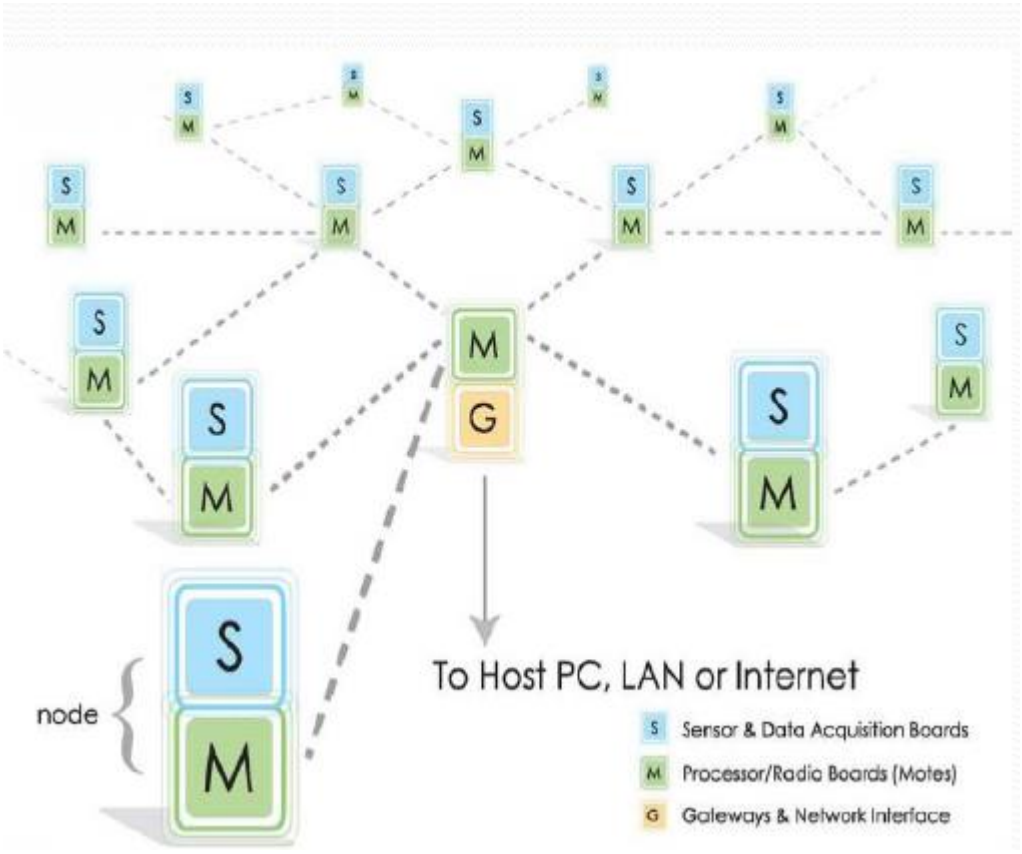


Figura 3.6. Esquema de una WSN. (Samaniego Armijos, 2015)

3.6.Nodos que forman una WSN.

Una mota de un nodo dentro de una WSN, no es más que el hardware que realiza el tratamiento de los datos que son adquiridos por los sensores, los almacena si es necesario, y transmite esa información hacia la mota que posea la puerta de enlace hacia la red o al internet, para que dichos datos sean analizados y tratados. La figura 3.7 enseña la arquitectura básica de una mota.

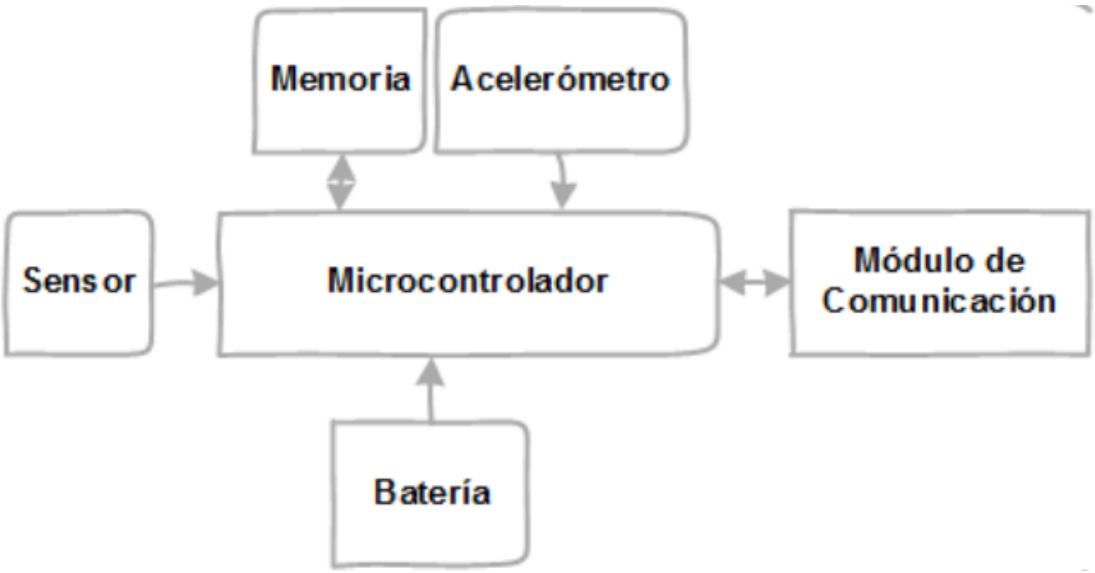


Figura 3.7. Figura de una Mota. (Samaniego Armijos, 2015)

3.7.Ventajas y Beneficios de elegir una Red Inalámbrica

Una Red WSN tiene una estabilidad y fiabilidad de igual calidad que un sistema de red de medida normal a base de cableado, con la diferencia que una red WSN es más flexible, tiene costos más bajos, con la opción incluso para crear la red.

Estos sistemas están diseñados para soportar áreas de difícil acceso y lugares con temperaturas irregulares. Sin embargo, son capaces de trabajar normalmente ante las dificultades ambientales que se presenten.

Son sistemas intuitivos donde no se necesita conocimiento de programación embebida, es decir, son programables en su totalidad y están listos para trabajar en cualquier circunstancia; fácilmente se puede programar el nodo con aplicaciones embebidas y entornos gráficos. Se puede obtener la información de una manera sencilla mediante visualización y análisis en la web, accediendo a los datos desde cualquier lugar, en cualquier plataforma. Habilitada en tiempo real, móvil o laptop. Cualquier equipo que pueda navegar en internet es útil.

La facilidad de no tener limitado el sistema de red de manera inalámbrica, permite tener la mayor flexibilidad, monitorear lugares de difícil alcance, incluso de ser necesario, reubicar los nodos o también añadir nodos a la red conforme la aplicación vaya creciendo.

En consecuencia, al evitarse utilizar cableado de cobre y Ethernet, se obtiene una mejor economía en el proyecto al ahorrar gran cantidad de dinero, ganando tiempo de trabajo, obteniendo condiciones ambientales y condiciones de los equipos, al trabajar con los sistemas inalámbricos al evitar la producción de tiempos muertos y reducir los costos de mantenimiento (National Instruments, 2015)

Conclusiones

Una red WSN brinda muchas ventajas al momento de realizar el monitoreo de una zona, es muy útil para ahorrar muchos recursos y más fácil de mantener comparando con una red de monitoreo a base de cableado. Es capaz de trabajar en ambientes difíciles de acceder, incluso soportar cambios climáticos fuertes, siendo un elemento básico del Internet de las Cosas.

4. CAPÍTULO 4: Internet de las Cosas en la Actualidad, Futuro y Seguridad

Introducción

En este capítulo se trata sobre la situación actual del Internet de las Cosas, sus usos contemporáneos, los avances, el apoyo por parte de empresas en el sector privado, implementaciones, y otros aspectos relacionados.

4.1. Caso Google – Carnegie Mellon

En muchos países del mundo las empresas, institutos o Universidades están haciendo inversiones para lograr avances significativos en este tema, tal caso es la Universidad de Carnegie Mellon, la cual tiene proyectos en marcha como la adquisición de un laboratorio especializado para demostrar la manera que IoT, a través de sensores, equipos y edificios, es capaz de cambiar la vida de las personas.

Figura 4.1. Logo de Google. (Google Inc.)

La empresa Google ha intervenido para invertir en el proyecto de dicha universidad, entregando medio millón de dólares y creando una alianza para futuras investigaciones.

Esta financiación proviene de la organización tecnológica *Open Web of Things*, iniciativa destinada a la creación de una "expedición de investigación e innovación abierta creada para crear avances para facilitar el desarrollo del internet inteligente y seguro de las aplicaciones y servicios de las cosas".

Estas inversiones dan fruto a una nueva plataforma del internet sobre las cosas llamado GLoTTO que tiene como objetivo crear un sistema interoperable de tecnologías de IoT, donde esta plataforma estará especializada en la seguridad privada de los investigadores.

El objetivo es mejorar la relación de humano-humano y humano-computador a través del internet de las cosas, mejorando las aplicaciones de uso y privacidad. A este proyecto, se suma la meta de desarrollar una *appstore* o tienda de aplicaciones del tipo IoT para la comunidad de investigación del campus, permitiendo la creación y acceso de dichas aplicaciones de IoT.

Actualmente ya están creadas algunas aplicaciones como Snap2It, una aplicación que permite conexión con impresoras o proyectores mediante el uso de equipos móviles, con tan solo tomar una foto a los dispositivos. Otro aporte es Impromptu, que puede acceder a las aplicaciones compartidas, por ejemplo, una aplicación de parada de autobús de trasporte público cuando se está en ella. (Hsu, Google Funds University Living Lab for Internet of Things, 2015)

4.2.Caso IBM

Figura 4.2. Logo IBM. (IBM, s.f.)

En abril del 2015, IBM, una de las empresas más grandes del mundo, invirtió 3 mil millones de dólares en IoT. Se centraron en el desarrollo de aplicaciones de *The Weather Company* para ayudar a empresas en campos como seguros, energía, comercio minorista y logística.

Con este acontecimiento, el internet de las cosas toma un gran crecimiento sobre el desarrollo de la tecnología donde los equipos y dispositivos sean capaces de recolectar los datos y compartir todo en línea, así como hay dispositivos y proyectos que ya están en funcionamiento, como por ejemplo *Apple Watch*, el termostato inteligente de *Nest* y carros inteligentes. *The Weather Company* ha vendido los datos de clima y tiempo a empresas que se dedican al transporte y agricultura (Hsu, IBM Bets $3 Billion on the Internet of Things, 2015)

4.3.Caso ARM

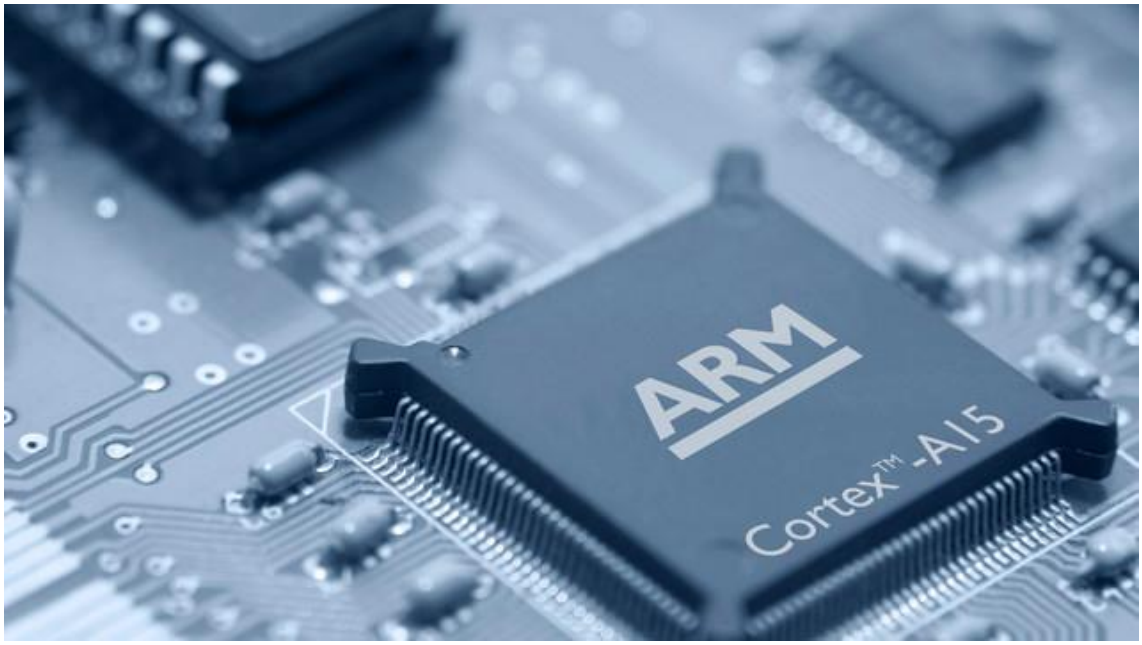


Figura 4.3. Empresa ARM Holdings plc. (ARM HOLDINNGS, s.f.)

ARM Holddings mencionó en 2014 la intención de poner en marcha el proyecto de un nuevo sistema operativo de baja potencia, que conectará dispositivos y aparatos que utilizan procesadores de 32 bits, el sistema operativo se llama MBED OS, está creado y diseñado para resolver los problemas de productividad donde diferentes dispositivos son llamados IoT y que corren en diferentes protocolos.

El objetivo es el siguiente: en lugar de tener grandes equipos, gastando años de diseño de un producto, la compañía pretende reducir esto en pocos meses de trabajo, de modo que permita tomar los componentes de hardware y

consolidar los equipos en un solo software simple. En los últimos años, ARM ha hecho un esfuerzo para desarrollar tecnologías para productos del internet de las cosas. Como resultado a una encuesta, se obtuvo que el 83% de los encuestados creen que el internet de las cosas ayudaría a mejorar su vida. (Pew Research Center, 2014)

Este es el primer sistema operativo desarrollado por ARM. El sistema operativo soporta varios estándares de conectividad como WI-FI, *bluetooth*, y también es compatible con algunos celulares estándares con 3G y LTE.

El sistema fue diseñado para brindar mejor servicio en el tema energético y pensado también en el tiempo de duración de batería, donde el sistema ocupa 256 KB de memoria. Parte del sistema operativo será de código de abierto que se ejecuta en el 36% de los sistemas operativo,s que con el paso de tiempo seguirán creciendo con Android y *Free RTOS* a la cabeza.

La acogida de este sistema operativo fue bastante buena donde fabricantes de hardware y otros desarrolladores se han incluido ya en este proyecto como son ERICSOON, *freescale*, IBM, NXP, y Zebra. (Patel, 2014)

4.4.Caso Bosch

Figura 4.4. Logo de empresa Bosch (Bosch, s.f.)

La empresa Bosch, conocida mundialmente como proveedor de fabricación de subministro de automóviles, anunció que planea contratar a 12.000 nuevos trabajadores en 2015, repartidos en 150 países en un número de 360 empresas y compañías regionales, empresa que ha entrado en el mundo de la tecnología mediante el internet de las cosas.

El 75% de los nuevos empleados son ingenieros, donde el objetivo de este proyecto es incrementar el diseño y el desarrollo de software. Christoph Kübel, director de relaciones industriales de Bosch, mencionó que la conectividad en todos los sectores de negocio, se va expandiendo y la importancia del software también lo hace. (Alexander Hellemans, 2015)

4.5.Caso CyberLightning

Figura 4.5. Logo de CyberLightning ltd. (CyberLightning, 2013)

CyberLightning es una empresa de emprendimiento que ofrece una plataforma para el uso industrial del internet de las cosas, resultando ser muy útil para compañías y proveedores para ayudar a monitorear sus productos a través del interfaz 3D, lo cual hace obtener datos complejos más fácil de manejar y que puede ser esencial cuando se gestionan redes de ciudades inteligentes, dentro de los aspectos de la era industrial del internet de las cosas. Actualmente, la compañía CyberLightning, que nació en el 2010, ha recibido un financiamiento por 4.2 millones de dólares por parte de sus inversionistas. (O'Hear)

4.6. Implementaciones y futuro del internet de las cosas

Todo tipo de innovación e inversión en tecnología nueva implica cierto riesgo y peligro a nivel empresarial, ya que existe incertidumbre al momento de los resultados esperados. Sin bien se dispone de todas las herramientas y la tecnología necesaria para comenzar los procesos, tanto hardware y software, siempre los involucrados están pendientes de los problemas de protección y fiabilidad, y que el proceso permita ofrecer servicios de la forma más eficiente posible.

Uno de los problemas planteados es que el poder de la red sigue muy centralizado, incluso en la era de la nube. Esto permite acceder y trabajar, pero solo es posible cuando no se accede a toneladas de datos y cuando la latencia no es un problema. Este punto no aplica en el internet de las cosas, ya que, por ejemplo, no se puede hacer algo como supervisar el tráfico en cada intersección de un lugar para una ruta más inteligente y evitar atascos. En este caso, la solución podría ser utilizar un paradigma distinto a la nube, el Computación en Niebla o *Fog Computing*, donde se está recopilando los datos para que la cantidad de estos, que puedan enviarse a los servidores centralizados, sea mínima, reduciendo de esta manera la latencia. (Bonomi, Milito, Zhu, & Addepalli, 2012)

Por el momento, la clave está en enfocarse en la inteligencia de los datos, si se pueden distribuir a nivel de aplicación sería más potente que cualquier hardware que se use para resolver el problema, ya que obtener los datos del dispositivo correcto en el momento correcto no consiste únicamente en el hardware y los sensores.

Por otro lado, se tiene la intervención de los datos móviles y el internet de las cosas que, sin duda alguna, los teléfonos inteligentes los recopilan y ofrecen una interfaz de usuario para acceder a ellos y sus aplicaciones. Todo esto siempre estará con el riesgo del incremento excesivo de datos, ya que más del 80% del tráfico de las redes LTE pasa a través de puntos de acceso inalámbrico. Si es que esto sería así, los dispositivos de comunicación tuvieran problemas e inconvenientes en cobertura, fiabilidad, consumo de energía y costos. En

consecuencia, se diría que los dispositivos móviles serían afectados en general en tema de rendimiento, disponibilidad, y otras características importantes. Para esto, el internet de las cosas necesita diferentes soluciones tanto de hardware y de software.

El internet de las cosas generará en un futuro una cantidad masiva de datos, por lo tanto es difícil mantener el ritmo cambiante y constante del crecimiento de datos generados, ya que no es posible supervisarlo todo.

El lugar donde se almacenarán los datos, es un factor muy importante en el internet de las cosas. El internet es el encargado de la distribución de los datos, por lo tanto, se podrá almacenar en un centro de datos esta información, si los datos no están ahí tardarían mucho en llegar y generar una respuesta en tiempo real.

Este tema tiene un proceso evolutivo, donde los componentes que intervienen como microcontroladores, microprocesadores, sensores ambientales, y de otros tipos de corto y largo alcance, son generalmente los más usados. En la actualidad, han crecido con el paso del tiempo, han evolucionado, siendo cada vez más pequeños y menos costosos al producir, donde estas áreas de servicios facilitarán capacidades de comunicación y control remoto vinculados todo con el internet. (Open Mind, s.f.)

Recordando su definición, el internet de las cosas es la red de objetos físicos que se acceden a través del internet, es la facilidad de conectar a personas, cosas, datos y procesos que está transformando la vida.

El internet de las cosas se perfila como la tercera ola del desarrollo del internet. En la década de 1990, se alcanzó un nivel de conexión de mil millones de usuarios, en la década del 2000, incluida la tecnología móvil, dos mil millones de usuarios. El internet de las cosas tiene un potencial de conectar alrededor de 28 mil millones de “cosas” para el año 2020. Gracias a los avances tecnológicos, dispositivos inteligentes como relojes, teléfonos han tenido bastante aceptación en el mercado. Según el reporte de *Global Investment Research*, ha existido un número significativo de cambios tecnológicos, que le

ha dado un crecimiento notorio al internet de las cosas, tales como sensores, que han reducido notablemente sus costos, y ancho de banda a costos más bajos. (Goldman Sachs, 2014)

Los smartphones se han convertido en el acceso personal de internet de las cosas, actuando como un control remoto, concentrador inalámbrico o *hub* para el hogar. En cuanto a *Big Data*, como su nombre lo dice, genera grandes cantidades de datos no estructurados por lo que se utiliza IPv6, la mayoría de los equipos ahora soporta este nuevo protocolo del internet.

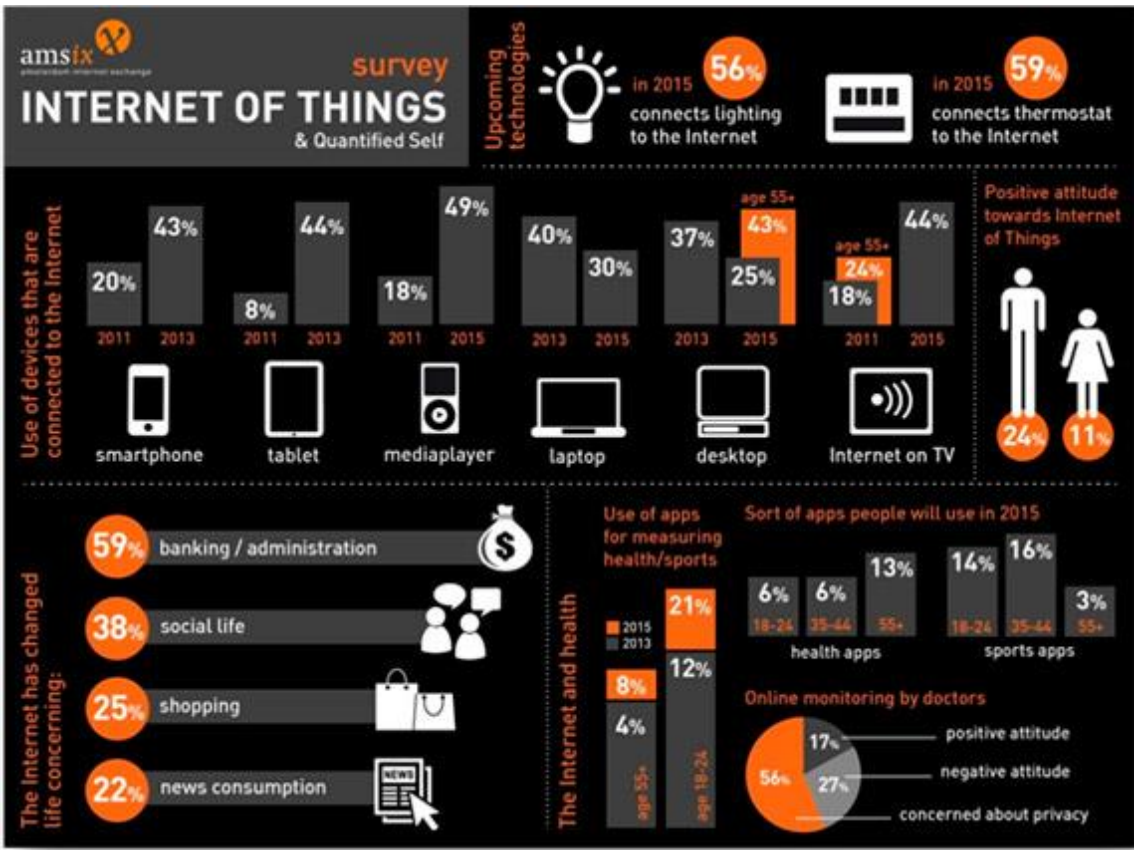


Figura 4.6. Esquema general y demanda del Internet de las Cosas. (Open Mind, 2015)

4.6.1. La Industria de Internet

Las empresas deberán abordar todas las disciplinas de TI para equilibrar el flujo de datos de dispositivos que están conectados a la red. IoT Implementará cambios en la estructura para permitir que sea transparente todo tipo de innovación para las personas. La llamada industria del internet ha logrado un alcance de diez a quince trillones de dólares en dos décadas. En la figura 4.7, se observa un esquema de la interactividad del IoT con el mundo en general.

Las siguientes innovaciones y proyectos a trabajarse, serían las inversiones tecnológicas descritas a continuación.

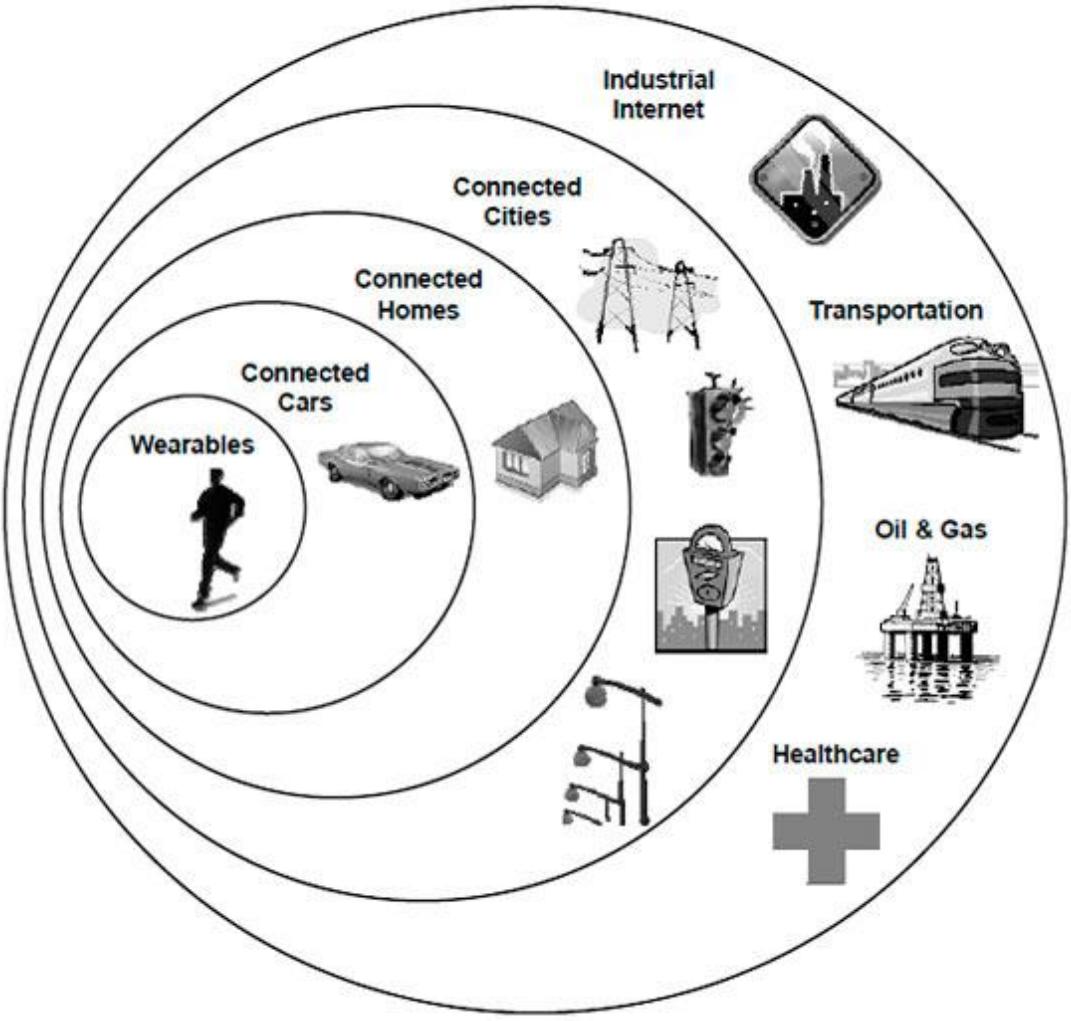


Figura 4.7. Internet de las Cosas y su interactividad con el mundo. (Goldman Sachs, 2014)

Internet de las Cosas y la Nube:

En los siguientes cinco años, más del 90% de los datos serán subidos a plataformas de la nube, reduciendo la complejidad de soporte del internet de las cosas.

Internet de las Cosas y Seguridad:

En los siguientes dos años, el 90% de las redes IT van a tener su seguridad basada en la vulnerabilidad del internet de las cosas.

Internet de las Cosas a través del Tiempo:

Para el 2018, el 40% de información del internet de las cosas será procesada, analizada, y actuará en base en consecuencia de la red.

Internet de las Cosas y la Capacidad de Red:

En tres años, El 50% de las redes IT se centrará en el manejo restringido de dispositivos IoT.

Internet de las Cosas y la Infraestructura no Tradicional:

Para el 2017, el 90% de la base de datos adoptarán nuevos modelos de negocio para administrar una infraestructura no tradicional.

Internet de las Cosas y diversificación Vertical:

Hoy en día, más del 50% de la actividad del Internet de las cosas está centrada en la fabricación, trasporte, ciudad inteligente y aplicaciones de consumo. Pero en cinco años, todas las industrias tendrán roles en base a las iniciativas del internet de las cosas.

Internet de las Cosas y Ciudades Inteligentes:

Trabajar para construir ciudades inteligentes, innovadoras y sostenibles. Un 25% de las inversiones actuales serían para implementar, administrar y realizar negocios basada en el internet de las cosas para el 2018.

Internet de las Cosas y Sistemas Embebidos:

Para el 2018, el 60% de soluciones IT, que originalmente fueron desarrolladas como propietarias, se volverán de *open source* permitiendo un surgimiento de aplicaciones IoT verticales.

Internet de las Cosas y Wearables:

Dentro de cinco años, el 40% de *wearables* se convertirá en una alternativa de mercado masivo de consumo viable para teléfonos inteligentes.

Internet de las Cosas y Millenials:

Para el 2018, el 26% de la población serán *millenials* (personas nacidas a partir del 2000) y tendrá una adopción acelerada de la realidad del internet de las cosas. (Open Mind, 2015)

Google en su conferencia llamada Google I-O para desarrolladores, anunció *Brillo*, un nuevo sistema operativo subyacente del internet de las cosas. Soporta Wi Fi, bluetooth y otros sistemas Android. Adicionalmente, existe en multiplataforma lo que permite a los dispositivos *Brillo*, teléfonos, y el internet comunicarse entre sí. Para el cuarto trimestre del año fiscal 2015, los dispositivos Android, detectarán automáticamente los dispositivos con *Brillo* y otros componentes que serán de ayuda para su interacción. En la figura 4.8 se puede ver un esquema de cómo se relaciona con otros medios. En figura 4.9 se observa la compatibilidad del sistema.

Figura 4.8. *Weave* y su relación con otros medios (Miller, 2015)

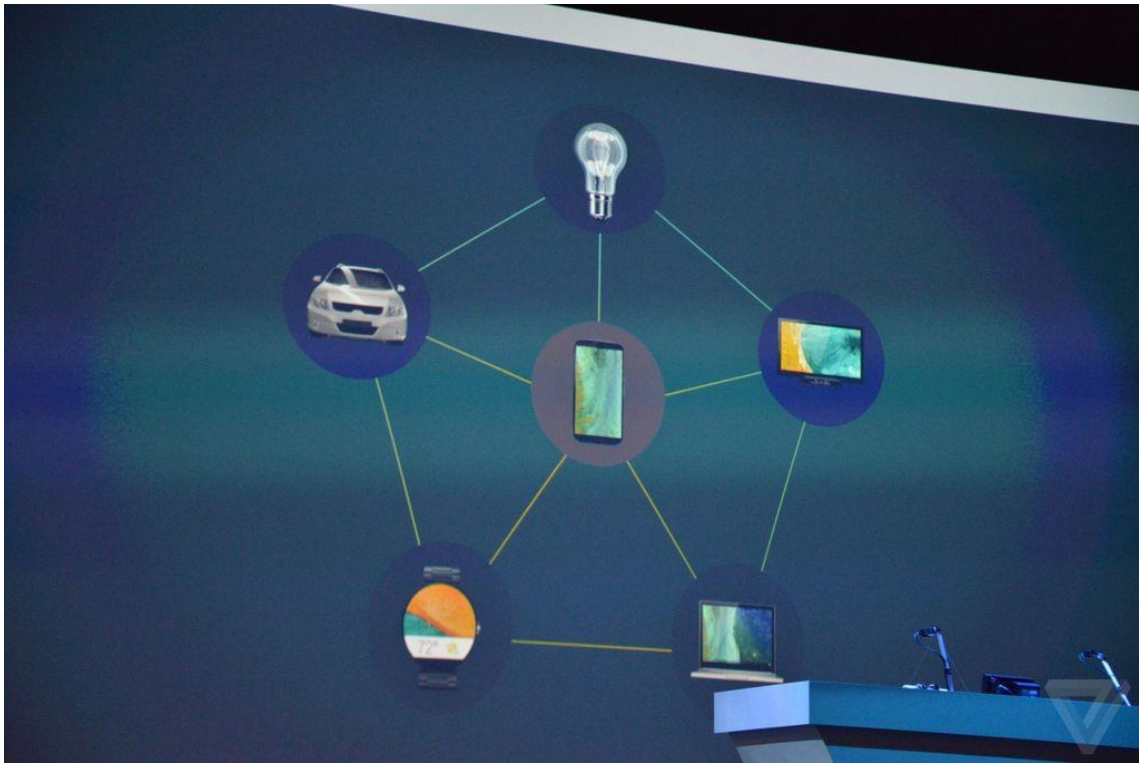


Figura 4.9. Compatibilidad del sistema con otros dispositivos. (Miller, 2015)

El nombre de *Brillo* fue mencionado en un reporte desde *Intercept*, (Efrati & Nellis, 2015) acerca de la desarrolladora de software de Google que puede funcionar con equipos de baja potencia y comunicarse con otros dispositivos cercanos. (Miller, 2015)

4.7.Seguridad para el Internet de las Cosas

Hasta ahora se ha hablado del internet de las cosas como evolución, implementación, y modos de uso donde el ser humano solo obtiene beneficios de ello para mejorar su estilo de vida, interactuando con las cosas de una manera sencilla que agilite la obtención de los datos que una persona pueda necesitar en el día a día, y con este método, obtener ventajas sobra la situación e incluso prevenir contratiempos de trabajo y aprovecharlos al máximo.

Pero a pesar que el Internet de las Cosas da muchos beneficios, también se puede trasformar en un peligro. Todo sistema o programa que participe de este grupo de intercomunicación de hombre y sistema inteligente, no está libre de ser atacado, así como cuando alguien burla las seguridades y puede ingresar al sistema de vigilancia, burla alarmas de sistemas de hogares, o peor aún, manipular los semáforos de la ruta de conducción.

Este es un tema que debe ser tratado con la mayor importancia, el incremento de la demanda de los usos de sistemas inteligentes y del desarrollo y mejoras de ellos, depende que la estabilidad y la confidencialidad que los sistemas puedan otorgar y es la clave del éxito de que en un futuro el internet de las cosas sea el dominante del mundo.

Está comprobado mediante estudios, que las vulnerabilidades existen y con un porcentaje muy alto de inseguridad. Un estudio hecho por *HP research* dice que en cada dispositivo del internet de las cosas, se puede encontrar hasta un promedio de 25 errores, y que en un 70% de los

dispositivos, se encuentra, por lo menos, una falla. (Hewlett-Packard, 2014)

El problema o vulnerabilidad es tan sencillo como que una persona que intente robar una casa ataque al sistema de medidor de luz donde el atacante o *hacker* corte la energía eléctrica de la casa, afectando a todos los sistemas de seguridad eléctricos que funcionan en la misma. En este caso, un hogar es vulnerable a un intento de robo.

En otro ejemplo sencillo, como el caso de las nuevas tecnologías de cerraduras electrónicas que se puedan abrir con tan solo un smartphone.

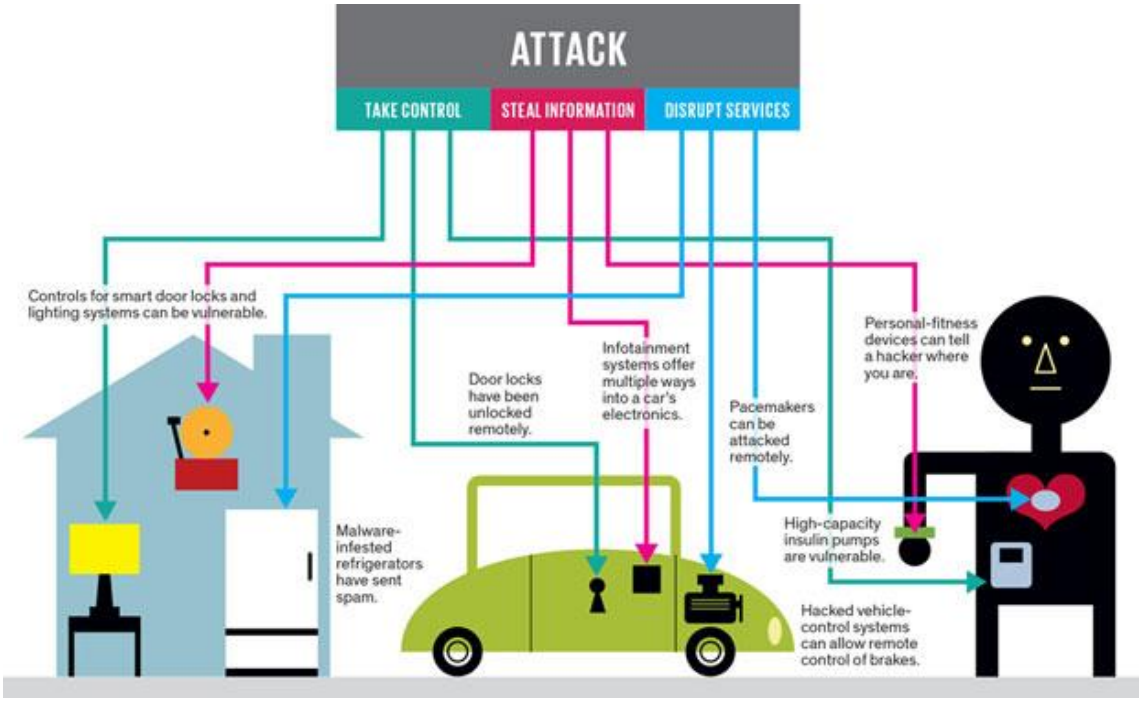


Figura 4.10. Modos y formas de ataque a los sistemas eléctricos. (Grau, 2015)

El mismo problema se da en los autos inteligentes, donde un informático malintencionado pueda atacar el sistema de un vehículo y, por ejemplo, encender y apagar las luces para identificar el auto, encender el motor o apagarlo y tomar el control del vehículo.

Otra forma de estar desprotegido es con la salud. En los hospitales utilizan equipos electrónicos y equipos de monitoreo donde un *hacker*

puede alterar el estado de los equipos y perjudicar gravemente a las personas allí internadas, ya que muchos de estos equipos se conectan a internet para su funcionamiento y trabajo.

Hoy en día, las personas se preocupan mayormente en obtener un nuevo sistema o un producto innovador, cuando no se dan cuenta que lo más importante es la seguridad del sistema que están obteniendo; que no tenga vulnerabilidades porque las consecuencias pueden ser graves. Es preferible un sistema seguro, aunque no sea tan nuevo, a uno de última tecnología con ciertas inseguridades. En la figura 4.10 se muestran las diferentes formas de ataque por parte del ladrón para afectar su hogar u oficina.

La mayoría de los dispositivos del internet de las cosas, no tienen una manera de actualización automática de su software, y situaciones como esta tienen que cambiar. Los fabricantes y productores tienen que encontrar la manera de combatir la piratería informática.

Los piratas informáticos, que normalmente atacan a los sistemas, buscan por lo generar robar información, tomar el control del dispositivo o interrumpir el servicio. Por lo que se desprende que existen muchos objetivos de los atentados a la informática, como por ejemplo espionaje, ingreso a los sistemas para recolección de datos como tarjetas de crédito, e información de una persona. Los medios pueden ser muy diversos como un teléfono, impresora, cámara de video, etc.

Por esto, es necesario el uso de un *firewall* y un IDPS (*Intrusion Detection and Prevention Systems* o sistemas de prevención y detección de intrusos) para impedir estos ataques. Un *firewall* actúa como un guardián que bloquea la obtención de información que está bloqueada. Un IDPS supervisa un sistema informático y la red, bloquea y reporta la actividad sospechosa. Existen otras medidas de protección como utilizar antivirus, que tiene la desventaja de ocupar demasiado espacio y requiere gran potencia para ejecutarse.

En este caso, se podría adoptar un enfoque diferente. En su mayoría, los dispositivos que conforman el internet de las cosas, son sistemas

embebidos que realizan funciones específicas dentro de los sistemas más complejos. Siendo así, se aprovecha la limitación de la función donde podría ser pequeño, rápido y eficiente.

Los sistemas de seguridad deben estar especializados en ataques específicos a los cuales el equipo es vulnerable. Como un sistema flexible para proteger los dispositivos, reforzando las debilidades del sistema, apoyándose con sistemas de motores de gran alcance y bases de datos de firmas de virus que actúan como filtros para detectar amenazas conocidas.

Algunos de los principales actores que trabajan en los sistemas integrados de seguridad, tales como, INTEL, *McAfee*, *Mentor Graphics*, *Wind River*, ya están incorporando nuevas tecnologías de seguridad tanto hardware como software, utilizados para los dispositivos del internet de las cosas.

Hay dos tipos de enfoques para asegurar los sistemas existentes. La una para los productos nuevos, que apoyan a las actualizaciones de software, el fabricante puede construir un *firewall* dentro de las capacidades del producto. Y existen sistemas más antiguos que tienen la capacidad de comunicación pero no son compatibles con actualizaciones de software, como sistemas de monitoreo de hospitales o sistemas de control de fábricas.

Es necesario proteger al internet de las cosas de sus atacantes como por ejemplo utilizando servidores de seguridad, apoyado con un cifrado de datos, con contraseñas fuertes, combinando los caracteres, letras, números y caracteres especiales y de ser posible, utilizar la autenticación basada en certificados, que son los mejores sistemas de seguridad tratados en la actualidad.

Hoy en día, existen estándares de seguridad para el internet de las cosas, *North American Electric Reliability Corp*. es una empresa que tiene un conjunto de directrices para ayudar a los fabricantes para generar productos con más seguridades, como los equipos de salud.

Otra manera de asegurar que el internet de las cosas sea más seguro, es el integrar un administrador de gestión de dispositivos de los productos, donde el software permite la comunicación del sistema de gestión de seguridad de la empresa del servidor. Como por ejemplo, el *McAfee ePolicy Orchestrator*, este tipo de administrador informaría de eventos como intentos de ingresos fallidos y también podría actualizarse el software de seguridad para nuevas amenazas que saldrían de un futuro.

Se debe tener en cuenta el ataque mediante la suplantación de identidad o *Phishing*, ya que es una manera muy fácil de ingresar a los equipos mediante correos electrónicos y mensajes que solicitan datos personales.

Sin embargo, la situación mejora cada vez más. Los fabricantes son conscientes de proteger los productos conectados al internet y están mejorando nuevos métodos de autenticación biométrica para los dispositivos móviles. Están promocionando la autenticación basada en exploración de retina, geometría de la mano, reconocimiento facial, y otros métodos de seguridad difíciles de evadir. (Grau, 2015)

Conclusiones

El Internet de las Cosas demuestra ser una poderosa herramienta al momento de vincularse con los objetos y herramientas de uso de la vida diaria. Siendo así, una nueva manera indispensable de realizar actividades de manera más sencilla y con implementos mejorados donde el ser humano simplifica mucho el estilo de vida, realizando actividades en menor tiempo del que le toma normalmente. Pero no todo es bueno, se tienen muchas vulnerabilidades, que si no se las tratan a tiempo, puede volverse una amenaza para el ser humano y perjudicarlo de manera muy grave por medio de robos y ataques a sistemas, oficinas o lugares de trabajo.

Mientras tanto, la gente ya aprovecha muchas cosas nuevas que existen en la actualidad y explotan los dispositivos inteligentes, que al momento de

utilizar el Internet de las Cosas se terminan convirtiendo en una comodidad y satisfacción para los usuarios.

Muchas industrias están apoyando a este nuevo campo de la ciencia para sacar el mayor provecho al momento de desarrollar nuevas tecnologías de software, aplicaciones para dispositivos inteligentes, y nuevos sistemas de seguridad.

Es fácil darse cuenta de los pasos agigantados a los que va la evolución de la tecnología. Pero, así mismo, se debe estar debidamente capacitado para tomar las debidas precauciones y saber auto protegerse ante cualquier circunstancia de estafa, ataque, *phishing* o cualquier método ya utilizados en la actualidad, para aprovecharse de las personas que no tienen la suficiente capacidad de informarse bien sobre los peligros y la responsabilidad que conlleva tener el Internet de las Cosas en medio de la vida diaria.

5. CAPÍTULO 5: Hardware y Sensores

Introducción

En este capítulo, se presenta el hardware necesario para realizar el prototipo propuesto de Monitoreo Ambiental. Este es el hardware que formaría parte de una red WSN y podría emplearse en distintas aplicaciones IoT.

Se describe la mota o plataforma de microcontrolador a utilizarse, Arduino, con sus *shields* o módulos Ethernet y WiFi. Finalmente, se indican las características técnicas de los sensores que medirán las variables ambientales de temperatura, humedad, concentración de monóxido de carbono e intensidad de sonido.

5.1. Placa Arduino

Para el prototipo de monitoreo ambiental, se pretende aplicar los conceptos que se explicaron sobre las redes inalámbricas de sensores o WSN, en el que cada nodo está compuesto por una mota y hardware de adquisición de datos o los sensores y estos datos son transmitidos hacia una red LAN o hacia el internet a través de otra mota con la interfaz de red correspondiente.

El prototipo consiste básicamente de un solo nodo con conexión a internet, el cual posee un conjunto de sensores que serán explicados más adelante. Este nodo, llega a ser una "cosa" de IoT.

La tarea de mota la desempeña la placa Arduino UNO, mostrada en la figura 5.1. Arduino es una plataforma hardware open-source utilizado para el desarrollo de prototipos de electrónica**.** (Arduino, s.f.)

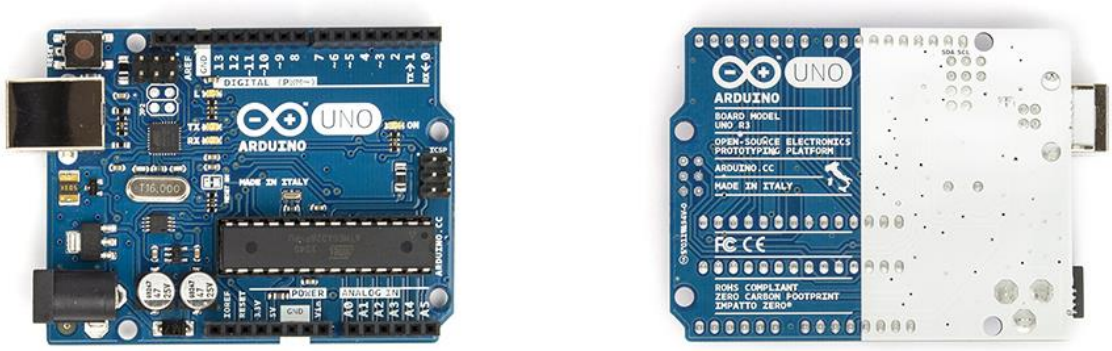


Figura 5.1. Vista frontal y posterior de la placa Arduino R3. (Arduino, s.f.)

Arduino nació en el Instituto de Diseño de Interacción Ivrea como una herramienta fácil para prototipado rápido, dirigido a estudiantes sin mucha experiencia en electrónica y programación. Tan pronto como llegó a una comunidad más amplia, la placa Arduino comenzó a cambiar para adaptarse a las nuevas necesidades y desafíos, expandiéndose más allá de las aplicaciones simples de microcontroladores de 8 bits a productos para aplicaciones del Internet de las Cosas, impresión 3D, *wearables*, y entornos integrados o embebidos. Todas las placas Arduino son open-source, permitiendo a los usuarios su construcción de forma independiente y así adaptarlos a sus necesidades particulares. El software también es de código abierto, que posee un conjunto de instrucciones programables a través del Arduino IDE. (Arduino, s.f.)

5.2.¿Por qué escoger Arduino?

Hay muchos otros microcontroladores y plataformas de microcontroladores disponibles para computación física. Todas estas herramientas toman los detalles complejos de programación y los envuelve en un paquete fácil de usar. Arduino también simplifica el proceso de trabajar con microcontroladores, pero ofrece algunas ventajas para los profesores, estudiantes y aficionados interesados sobre otros sistemas (Arduino, s.f.):

- **Asequible** – las placas Arduino son relativamente baratas en comparación con otras plataformas de microcontroladores. La versión menos costosa del módulo Arduino puede ensamblarse a mano, e incluso los módulos de Arduino pre-ensamblados cuestan menos de $ 50.
- **Multiplataforma** - El software de Arduino se ejecuta en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los sistemas de microcontrolador se limitan a Windows.
- **Entorno de programación simple** - El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para usuarios avanzados para que puedan también aprovecharlo.
- **Código abierto y software extensible** - El software de Arduino está publicado como herramientas de código abierto, disponible para la extensión por programadores experimentados. El lenguaje se puede ampliar a través de bibliotecas de C ++.

5.2.1. Características Técnicas

El Arduino Uno es una placa electrónica basada en el ATmega328. Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 se pueden utilizar como salidas PWM), 6 entradas analógicas, un resonador cerámico de 16 MHz, una conexión USB, un conector de alimentación, un header ICSP, y un botón de *reset*. Contiene todo lo necesario para apoyar el microcontrolador; simplemente hay que conectarlo a un computador con un cable USB o alimentarlo con un adaptador o una batería. La tabla 5.1 muestra el resumen de características técnicas.

Microcontrolador	ATmega328
Voltaje de operación	5V
Voltaje de entrada (recomendado)	7-12V

Voltaje de entrada (límites)	6-20V
Pines de E/S Digitales	14(de los cuales 6 proveen salidas PWM)
Pines de Entrada analógicos	6
Corriente DC por Pin de E/S	40 mA
Corriente DC por Pin de 3.3V	50 mA
Memoria Flash	32 KB
SRAM	2KB
EEPROM	1KB
Velocidad de reloj	16 MHz
Longitud	68.6 mm
Anchura	53.4 mm
Peso	25 g

Tabla 5.1. Características técnicas de la placa Arduino UNO. (Arduino, s.f.)

5.2.1.1. Alimentación

El Arduino Uno puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

La alimentación externa puede venir de un adaptador de CA a CD o una batería. El adaptador se conecta con un enchufe de 2,1 mm de centro-positivo al conector de alimentación de la placa. Las terminales de una batería se pueden insertar en los pines GND y Vin del conector POWER.

La placa puede funcionar con un suministro externo de 6 a 20 voltios. Si se suministra con menos de 7V, sin embargo, el pin de 5V puede suministrar menos de cinco voltios y la placa puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios. (Arduino, s.f.)

Los pines de alimentación son los siguientes:

- **VIN**. El voltaje de entrada a la placa Arduino cuando se utiliza una fuente de alimentación externa (por oposición a los 5 voltios de la conexión USB u otra fuente de alimentación regulada). Se puede suministrar tensión a través de este pin.
- **5V**. Este pin provee una salida regulada de 5V. El tablero puede ser alimentado ya sea desde la toma de alimentación de CD (7 - 12 V), el conector USB (5V), o el pin VIN del tablero (7-12V).
- **3V3**. Un suministro de 3,3 voltios generado por el regulador de la placa. El consumo de corriente máximo es de 50 mA.
- **GND**. Pines de tierra.
- **IOREF**. Este pin de la placa Arduino proporciona la referencia de tensión con la que opera el microcontrolador.

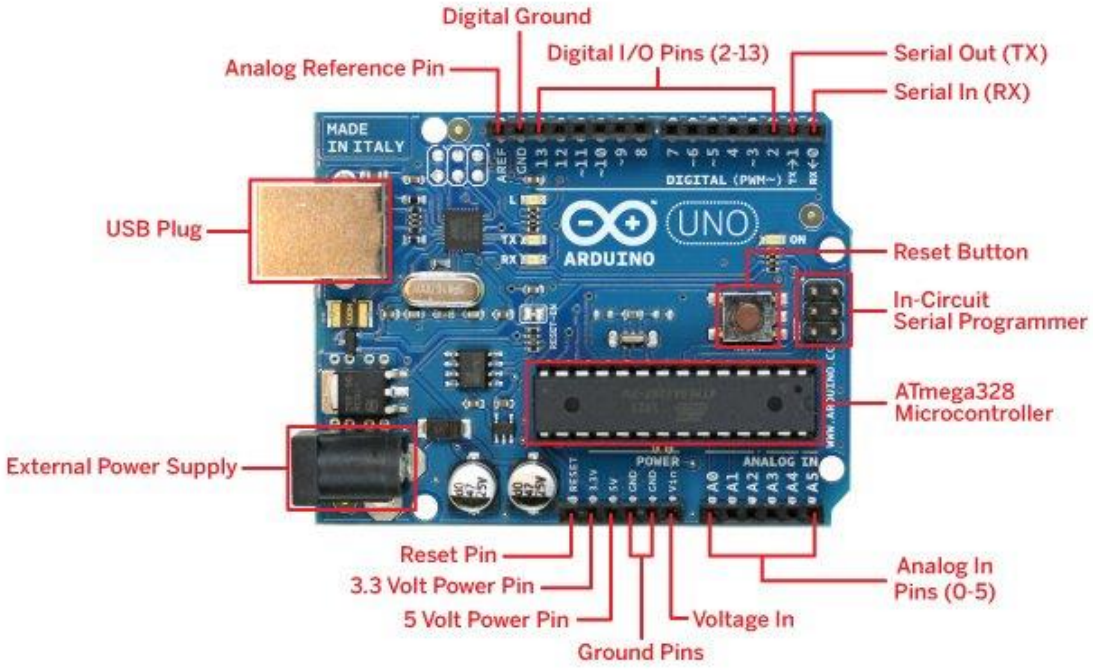


Figura 5.2. Distribución de Pines de la placa Arduino UNO. (ToWebOffice, s.f.)

5.2.1.2.Memoria

El microcontrolador ATmega328 tiene 32 KB de memoria flash (con 0,5 KB utilizado para el gestor de arranque). También tiene 2 KB de SRAM y 1 KB de EEPROM (que puede ser leído y escrito con la librería EEPROM). (Arduino, s.f.)

5.2.1.3.Entrada y Salida

Cada uno de los 14 pines digitales en el Arduino Uno puede ser utilizado como una entrada o salida, utilizando las funciones pinMode (), digitalWrite (), y digitalRead (). Estas operan en 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de *pull-up* (desconectado por defecto) de 20 a 50 kOhm. Además, algunos pines tienen funciones especializadas (Arduino, s.f.):

- **Serial**: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir (TX) datos en serie
- **Interrupciones externas**: 2 y 3. Estos pines pueden configurarse para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor.
- **PWM**: 3, 5, 6, 9, 10, y 11. Proporciona una salida PWM de 8 bits con la función *analogWrite()*.
- **SPI**: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines soportan la comunicación SPI utilizando la librería SPI.
- **LED**: 13. Hay un LED incorporado conectado al pin digital 13. Cuando el pin está en valor *HIGH* o en alto, el LED está encendido, cuando el pin está en bajo, se apaga.

Arduino Uno tiene 6 entradas analógicas, etiquetadas de A0 a A5, cada una de las cuales proporcionan 10 bits de resolución (es decir, 1.024 valores diferentes). Por defecto se miden desde tierra a 5 voltios, aunque es posible cambiar el extremo superior de su rango usando el pin AREF y

la función *analogReference()*. Además, algunos pines tienen funciones especializadas (Arduino, s.f.):

- **TWI**: pin A4 o SDA y A5 o SCL. Soportan comunicación TWI mediante la biblioteca *Wire*.

Hay un par de otros pines en la placa:

- **AREF**. Tensión de referencia para las entradas analógicas. Se utiliza con *analogReference()*.
- **Reset**. Se pone esta línea en BAJO para reiniciar el microcontrolador. Normalmente se utiliza para añadir un botón de reinicio cuando se ve bloqueado el del arduino.

5.3. Shields de Arduino

La placa Arduino UNO, por sí sola, no puede conectarse a una red LAN o al internet. Para ello, necesita de módulos externos (que trabajen en algunos de los protocolos ya mencionados que se utilizan en IoT) que se conectan a la placa, llamados *shields*. A continuación, se describen los *shields* para conexión Ethernet y conexión WiFi.

5.3.1. Arduino Ethernet Shield

El *Shield* de Ethernet permite que una placa Arduino se conecte a internet o a una red mediante un conector RJ-45 estándar. Se basa en el chip ethernet Wiznet W5100. El Wiznet W5100 ofrece una red (IP) y un stack que soportan los protocolos de control de transmisión y de datagrama de usuario, TCP y UDP, respectivamente. Soporta hasta cuatro conexiones de *socket* simultáneas, ya que permite a otros *shields* ser apilados en la parte superior. (Arduino, s.f.)

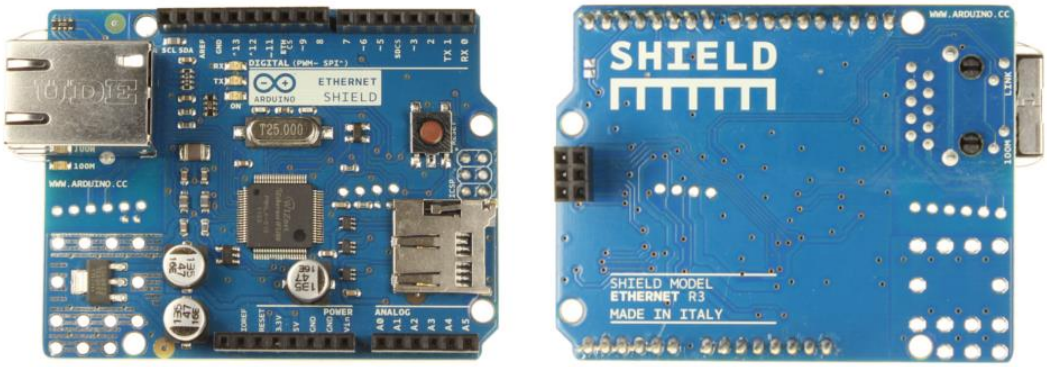


Figura 5.3. Vista frontal y posterior del *shield* de Arduino para Ethernet. (Arduino, s.f.)

El *shield* posee una ranura para tarjetas micro-SD, que se puede utilizar para almacenar archivos para servir a través de la red. Es compatible con el Arduino Uno y Mega (utilizando la librería Ethernet). El lector de tarjetas microSD es accesible a través de la librería SD.

Tiene además un módulo de alimentación a través de Ethernet (PoE), diseñado para extraer energía de un cable Ethernet de par trenzado de categoría 5 convencional. No viene con el módulo PoE integrado, es un componente separado que debe ser añadido a la placa.

El shield contiene un número de LEDs informativos:

- **PWR**: indica que la placa y el escudo están encendidos.
- **LINK**: indica la presencia de un enlace de red y parpadea cuando el shield transmite o recibe datos.
- **FULLD**: indica que la conexión de red es *full duplex*
- **100M**: indica la presencia de una conexión de red de 100 Mb/s (en contraposición a 10 Mb/s)
- **RX**: LED intermitente cuando el *shield* recibe datos.
- **TX**: LED intermitente cuando el *shield* envía datos.
- **COLL**: LED intermitente cuando se detectan colisiones de red.

5.3.2. Arduino WiFi Shield

El *Shield* de WiFi permite que una placa Arduino se conecte a Internet utilizando la especificación inalámbrica 802.11 (Wi-Fi). El microcontrolador de este *shield* , el AT32UC3, proporciona una red (IP) y un stack que soporta TCP y UDP, permite a otros *shields* ser apilados en la parte superior. El *shield* WiFi puede conectarse a redes inalámbricas que operan de acuerdo con las especificaciones 802.11b y 802.11g. (Arduino, s.f.)

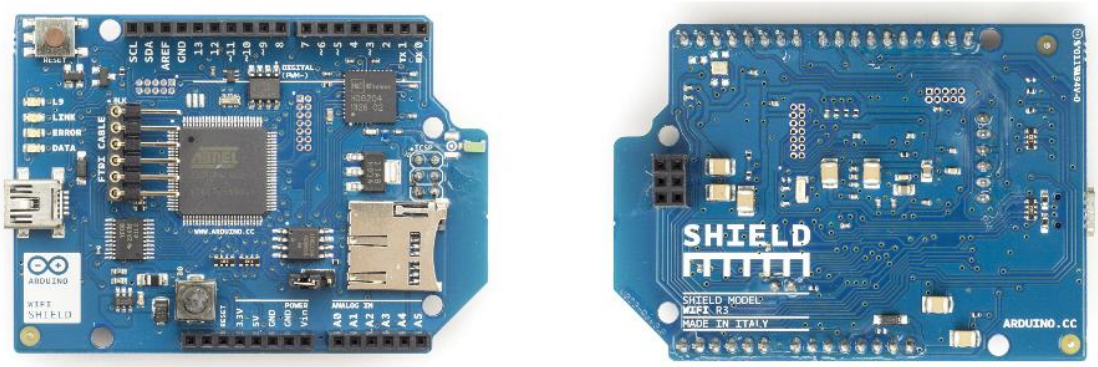


Figura 5.4. Vista frontal y posterior del *shield* de Arduino para WiFi. (Arduino, s.f.)

El *shield* puede conectarse a redes encriptadas que utilizan ya sea encriptación WPA2 (*Wi-Fi Protected Access II*) Personal o encriptación WEP (*Wired Equivalent Privacy*). También puede conectarse a redes abiertas. Una red debe difunde su SSID para que el *shield* sea capaz de conectarse.

El *shield* contiene un número de LEDs informativos:

- **L9 (amarillo):** este está ligado al pin digital 9.
- **LINK (verde):** indica una conexión a una red.
- **ERROR (rojo):** indica cuando hay un error de comunicación.
- **DATOS (azul):** indica los datos que se transmiten/reciben.

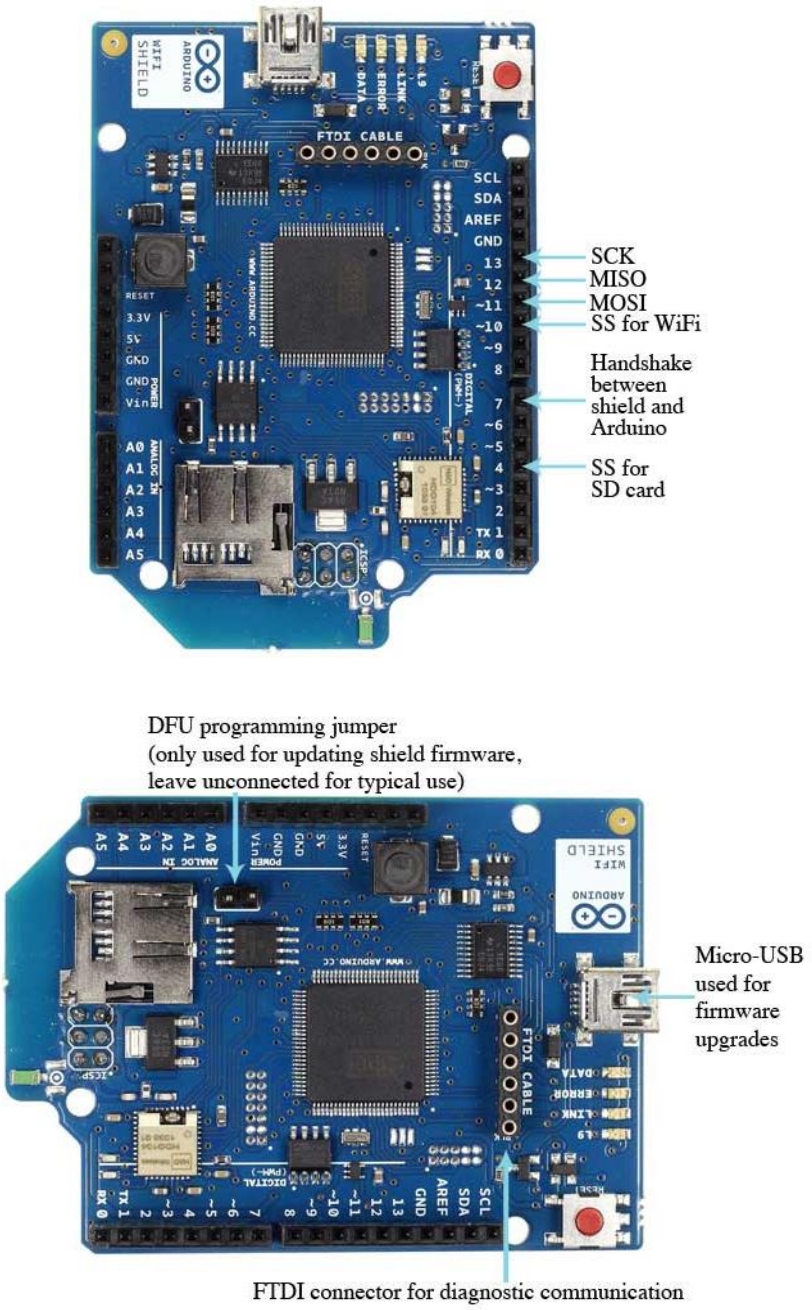


Figura 5.5. Interfaces del *shield* de WiFi. (Arduino, s.f.)

Es importante destacar la importancia de este *shield*, ya que este es el que se utilizaría dentro de una red WSN, justamente por la comunicación inalámbrica que se podría conseguir entre otras placas Arduino (motas) con sus respectivos *shields* wifi.

5.4. Sensores

Para medir las variables ambientales propuestas; ruido, concentración de monóxido de carbono, temperatura y humedad; se requieren de sensores o transductores correspondientes. A continuación se indicarán los sensores empleados en este prototipo de monitoreo ambiental.

5.4.1. Temperatura y Humedad

Se puede medir ambos de estos parámetros con un solo sensor, el sensor DHT11 mostrado en la figura 5.6.

Figura 5.6. Sensor de temperatura y humedad DHT11 de la marca wRobot. (Emartee, s.f.)

De acuerdo a su hoja de datos, este sensor de temperatura y humedad DHT11, dispone a la salida una señal digital calibrada. Su tecnología garantiza la confiabilidad y estabilidad a largo plazo. Posee un microcontrolador de 8-bit. Este sensor incluye un elemento de resistencia y detección de humedad de los dispositivos de medición de temperatura, NTC (*Negative temperature coefficient* o coeficiente de temperatura negativo).

Su descripción indica que cada sensor DHT11 presenta características de calibración precisas para la cámara de calibración de humedad. Los coeficientes de calibración están almacenados en la memoria de programa

OTP (*One-Time Programmable* o Programable una vez), los cuales son utilizados por el proceso interno de detección de señales. El producto viene en un paquete de 4 pines de una sola fila. La tabla 5.2 indica sus especificaciones técnicas. (D-Robotics, 2010)

Voltaje de alimentación	3 – 5.5 V CD
Señal de salida	Señal digital a través de un solo bus
Elemento sensor	Resistor de polímero
Rango de medición	Humedad 20 – 90 %HR; Temperatura 0 – 50 Celsius
Precisión	Humedad +- 5 %HR Temperatura +- 2.0 Celsius
Resolución o sensibilidad	Humedad 1 %HR; Temperatura 0.1 Celsius
Repetibilidad	Humedad +-1 %HR; Temperatura +- 1 Celsius
Histéresis de humedad	+- 1 % HR
Estabilidad a largo plazo	+- 0.5 % HR/año
Período de detección	Promedio: 2s
Intercambiabilidad	Totalmente intercambiable
Dimensiones	Tamaño 12*14.5*5.5mm

Tabla 5.2. Especificaciones técnicas del sensor DHT11.

5.4.2. Concentración de CO

El parámetro a medir es la concentración de monóxido de carbono (CO) en el aire, para ello se utiliza el sensor MQ7 mostrado en la figura 5.7.

Figura 5.7. Sensor de gas MQ7 para CO. (Shenzhen Shanhai Technology Ltd., s.f.)

Los datos técnicos del sensor, presentados en las tablas 5.3, 5.4 y 5.5, indican que este tiene una alta sensibilidad al monóxido de carbono con una larga y estable vida de funcionamiento. Sus aplicaciones más comunes son en el equipamiento para detectar CO en el hogar, industria o automóvil.

Símbolo	**Nombre del parámetro**	**Condición técnica**	**Observación**
Vc	Voltaje del circuito	5 V +- 0.1	CA o CD
VH (H)	Voltaje de calentamiento (high)	5 V +- 0.1	CA o CD
VH (L)	Voltaje de calentamiento (low)	1.4 V +- 0.1	CA o CD
RL	Resistencia de carga	Ajustable	
RH	Resistencia de calentamiento	33 Ω +- 5%	Temperatura de la habitación

TH (H)	Tiempo de calentamiento (high)	60 +- 1 segundos	
TH (L)	Tiempo de calentamiento (low)	90 +- 1 segundos	
PH	Consumo de calor	Cerca de 350 mW	

Tabla 5.3. Condiciones estándar de trabajo. (Hanwei Electronics)

Símbolo	**Nombre del parámetro**	**Condición técnica**	**Observación**
Tao	Temperatura de uso	-20°C – 50°C	
Tas	Temperatura de almacenamiento	-20°C – 50°C	
HR	Humedad relativa	Menor al 95 %HR	
O2	Concentración de oxígeno	21% (condición de soporte) la concentración de oxígeno puede afectar la sensibilidad	El valor mínimo es superior al 2%

Tabla 5.4. Condiciones ambientales. (Hanwei Electronics)

Símbolo	**Nombre del parámetro**	**Condición técnica**	**Observación**
Rs	Resistencia de superficie	2-20k	En 100ppm de monóxido de carbono

a (300/100 ppm)	Tasa de pendiente de concentración	Menor a 0.5	
Condición estándar de trabajo	Temperatura -20°C +- 2°C humedad relativa 65% +-5% RL:10KΩ +- 5%		
	Vc: 5V +- 0.1V VH:5V +- 0.1V VH:1.4+- 0.1V		
Tiempo de precalentamiento	No menor a 48 horas	Rango de detección: 20ppm -2000ppm de monóxido de carbono	

Tabla 5.5. Características de sensibilidad. (Hanwei Electronics)

La figura 5.8 muestra las características de sensibilidad típicas del MQ-7 para varios gases. Con los parámetros:

- Temperatura: 20°C
- Humedad: 65%
- Concentración de O2 del 21%
- RL = 10kΩ
- Ro: resistencia del sensor a 100pm de CO en el aire.
- Rs: resistencia del sensor a varias concentraciones de gases.

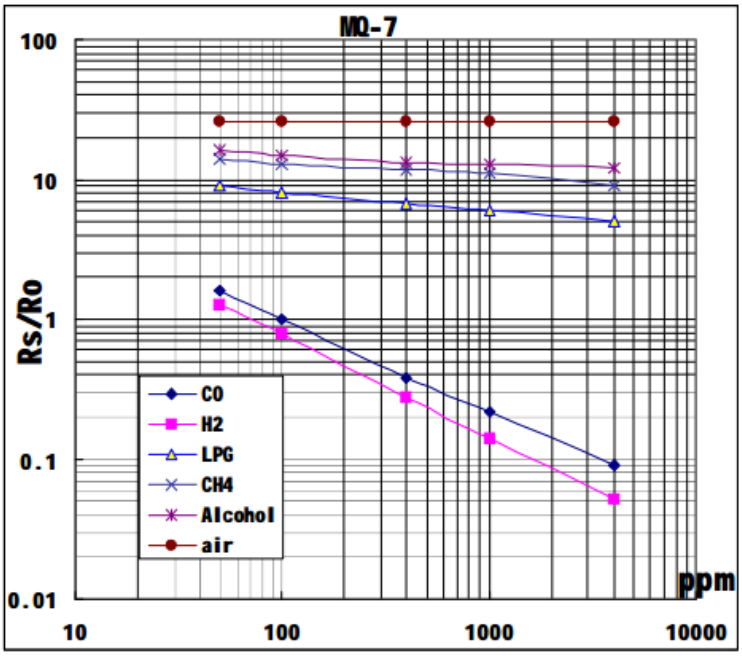


Figura 5.8. Características de sensibilidad del MQ-7. (Hanwei Electronics)

La figura 5.9 muestra la dependencia típica del MQ-7 con la temperatura y la humedad.

- Ro: Resistencia del sensor a 100ppm de CO en el aire a 33 %HR y 20°.
- Rs: Resistencia del sensor a 100ppm de CO a diferentes temperaturas y humedades.

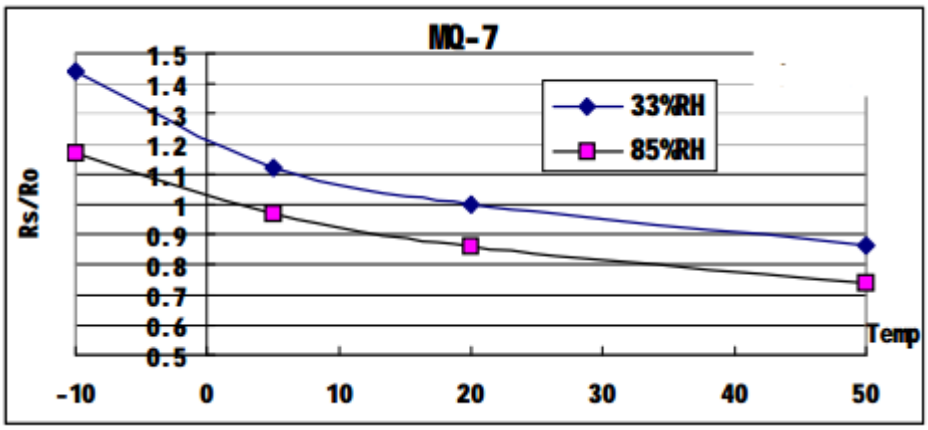


Figura 5.9. Características de dependencia de temperatura y humedad del MQ-7. (Hanwei Electronics)

Principio de operación

La resistencia de superficie del sensor, Rs, se obtiene a través de la señal de voltaje de salida de la resistencia de carga RL. La relación entre ellos se describe con la siguiente ecuación:

$$Rs/RL = (Vc-VRL) / VRL$$

Ajuste de sensibilidad

El valor de resistencia del MQ-7 es diferente para diversos tipos y diversas concentraciones de gases. Por lo tanto, cuando se utilizan estos componentes, El ajuste de sensibilidad es muy necesario. La hoja de datos recomienda que se calibre el detector para 200 ppm de CO en el aire y el

valor de uso de la resistencia de carga (RL) sobre 10 KΩ (5KΩ a 47 KΩ). (Hanwei Electronics)

5.4.3. Ruido (Sonido)

Se pretende realizar la medición de la intensidad de sonido utilizando el sensor básico de la figura 5.10 que cumple este propósito.

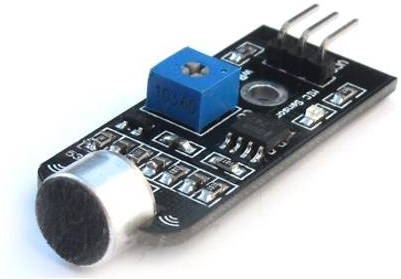


Figura 5.10. Sensor de sonido compatible con Arduino de marca wRobot. (Elechouse)

Este módulo puede detectar sonidos a su alrededor. Siempre que el sonido está por encima de un cierto límite (ajustable por el potenciómetro), existirá una señal de salida. El límite de activación de este módulo se puede ajustar. Para hacerlo, se debe simplemente ajustar la resistencia variable.

La fuente de alimentación es de 5 V, que es compatible con la mayoría de microcontroladores o plataformas como Arduino. (Elechouse)

Para verificar la exactitud de las mediciones, se las compara con un dispositivo más sofisticado, el sonómetro 3M SoundPro, mostrado en la figura 5.11.

Figura 5.11. Sonómetro 3M Quest SoundPro.

Este sonómetro está calibrado a 114 dB y a 3 segundos, en los cuales realiza 15 mediciones y los promedia. El ángulo de recepción del sensor en teoría es de 360° pero en la práctica se acerca más a unos 270°.

Conclusiones

De este capítulo cabe destacar que los sensores descritos, no llegan a ser tan sofisticados ni sensibles como los que se utilizarían en una aplicación de carácter científico y de análisis como el de un verdadero monitoreo ambiental. Para fines académicos y de prototipado, se han seleccionado estos para ejemplificar uno de los servicios que se pueden ofrecer aplicando el concepto del Internet de las Cosas. Por lo que los datos medidos no podrán compararse totalmente con los reales. Estos resultados se analizan más adelante.

6. CAPÍTULO 6: Software y Plataformas

Introducción

En este capítulo, se expone sobre el software de programación de Arduino para el prototipo de monitoreo ambiental y de las plataformas de nube IoT seleccionadas, ThingSpeak y Nimbits, por ser las únicas Open Source y Free Source (ver tabla 2.1) de las que se detallará su funcionamiento.

6.1. Arduino IDE

El entorno de desarrollo integrado de Arduino - o Arduino Software (IDE) - contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino para cargar programas y comunicarse con ellos.

Figura 6.1. Mensaje de inicio del software de Arduino.

Los programas escritos utilizando el Software de Arduino (IDE) se llaman *sketches* o bocetos. Estos *sketches* se escriben en el editor de texto y se guardan con la extensión de archivo *.ino*. El editor tiene funciones para cortar/pegar y para buscar/reemplazar texto. El área de mensajes proporciona

retroalimentación mientras se guarda y exporta el programa, también muestra los errores que pudieron darse en estos procesos. La consola muestra la salida de texto por el software de Arduino (IDE), incluidos mensajes de error completos y otra información. La esquina derecha inferior de la ventana muestra el modelo de placa Arduino configurado y puerto serie al cual está conectado. Los botones de la barra de herramientas permiten verificar y cargar programas, crear, abrir y guardar sketches, y abrir el monitor serial. (Arduino, s.f.)

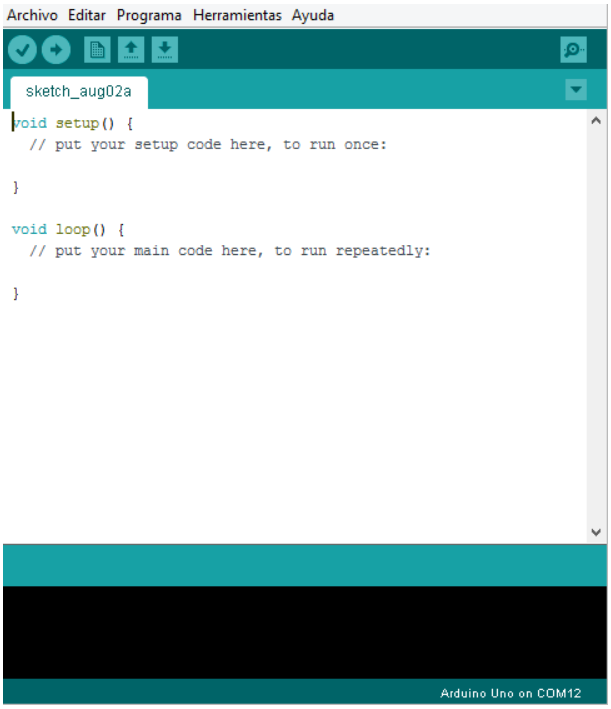


Figura 6.2. Interfaz del IDE, un sketch básico.

6.2. ThingSpeak

ThingSpeak es una plataforma para el "Internet de las cosas" la cual cuenta con una interfaz de programación de aplicaciones o API (*application program interface*) de código abierto para almacenar y recuperar datos de las cosas usando el protocolo de transferencia de hipertexto HTTP (*Hypertext Transfer Protocol*) a través de internet o a través de una red de área local. Con ThingSpeak, se pueden crear aplicaciones de registro de lecturas de sensores,

aplicaciones para seguimiento y localización, y una red social de las cosas con actualizaciones de estado.

Además de almacenar y recuperar datos numéricos y alfanuméricos, la API de ThingSpeak permite el procesamiento de datos numéricos como escala de tiempo, promedios, mediana, sumatorias y redondeo. Cada canal ThingSpeak soporta entradas de datos de hasta 8 campos, además de la información geográfica del dispositivo de transmisión (la mota) como latitud, longitud y elevación. Los canales soportan los formatos JSON, XML y CSV para su integración en aplicaciones.

La aplicación ThingSpeak también cuenta con control de zonas horarias, gestión de la lectura / de claves y gráficas basadas en JavaScript.

El Soporte para ThingSpeak está disponible en la Comunidad del sitio web que cuenta con un blog, foro, documentación y tutoriales. (iobridge ThingSpeak)

Además, cuenta con una aplicación móvil para teléfonos Android para visualizar las gráficas de los datos que recibe en el canal.

Para empezar a utilizar la plataforma ThingSpeak, se sigue la siguiente guía de inicio rápido (ThingSpeak, s.f.):

1. Registrarse a una cuenta ThingSpeak.
2. Crear un nuevo canal para la recepción de los datos.
3. Actualizar el canal vía URL: https://api.thingspeak.com/update?api_key=CLAVE__CANAL&field1=7
4. Ver el *feed* del canal: https://api.thingspeak.com/channels/ID_CANAL/feeds.json

6.2.1. Interfaces de la plataforma

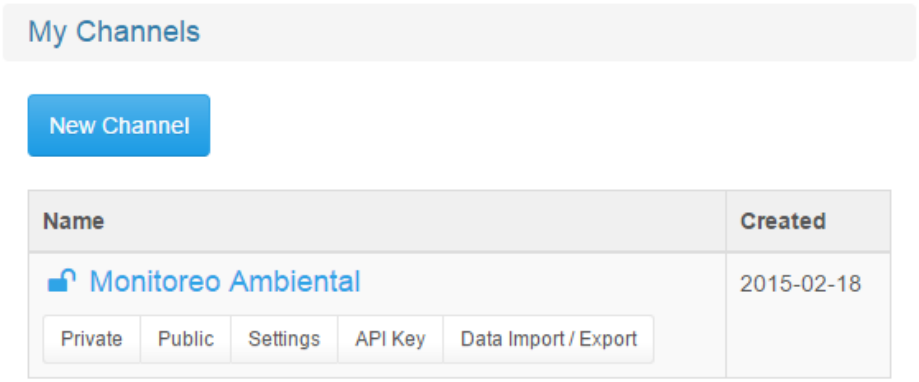


Figura 6.3. Gestión de los canales creados en ThingSpeak.

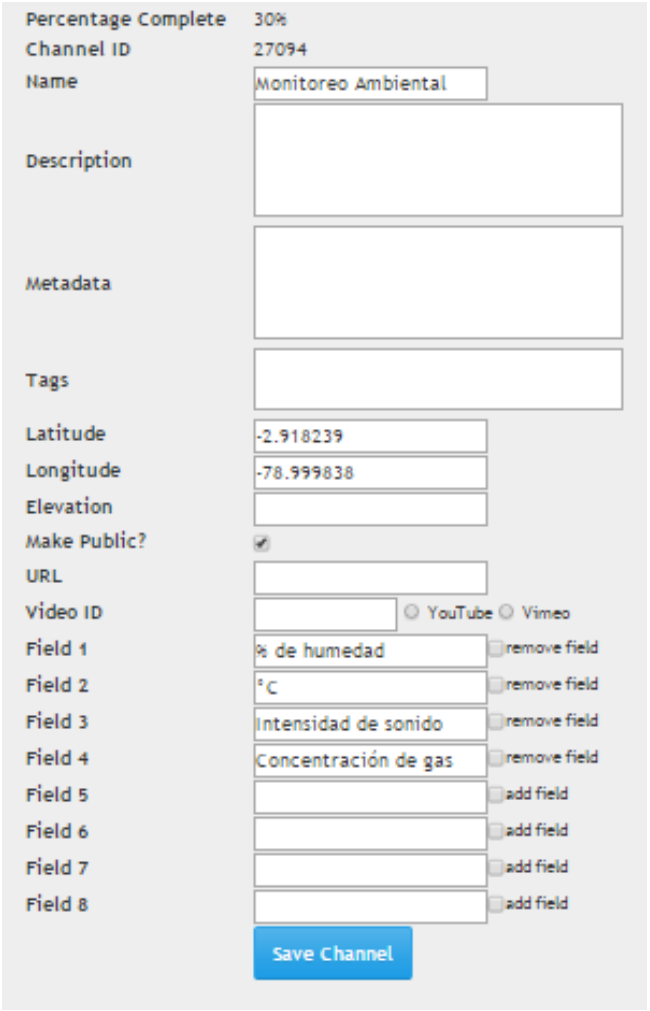


Figura 6.4. Configuración del canal.

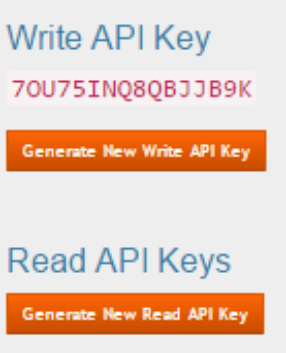


Figura 6.5. Claves o llaves para la lectura y escritura de datos al canal.

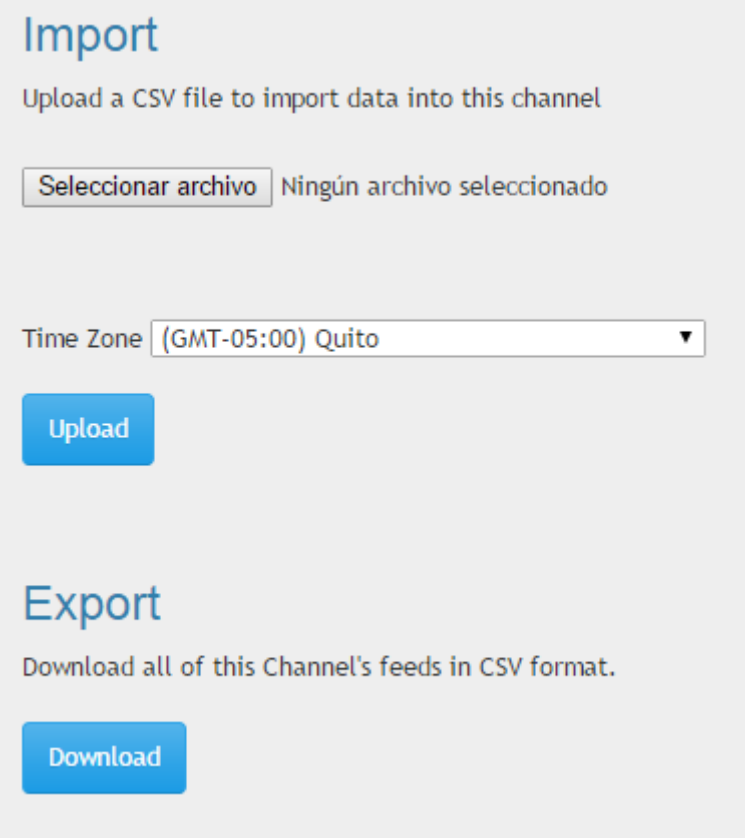


Figura 6.6. Importación y exportación de archivos.

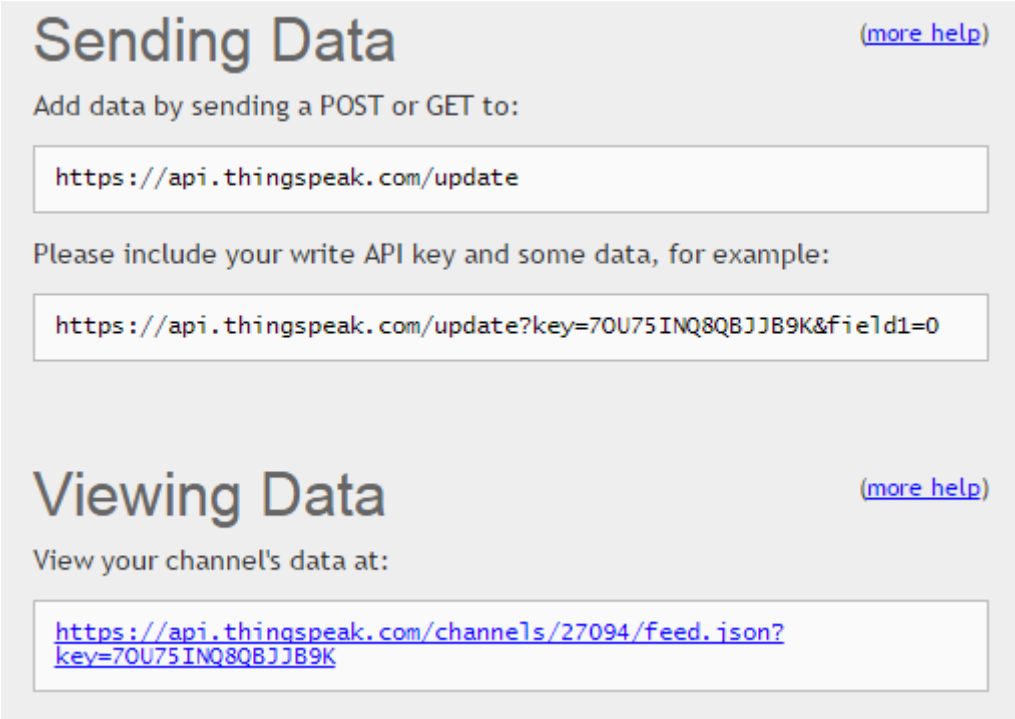


Figura 6.7. Envío y visualización de los datos desde un navegador.

6.2.2. Sumario

Características (ThingSpeak, s.f.)

- API abierta
- Recolección de datos en tiempo real
- Datos de geolocalización
- Procesamiento de datos
- Visualización de datos
- Mensajes del estado del dispositivo
- Plugins

Integración

- Arduino
- Raspberry Pi
- ioBridge / RealTime.io

- Electric Imp
- Aplicaciones móviles / web
- Redes Sociales
- Análisis de datos

Comunidad

- GitHub
- Twitter / Google+
- Blog
- Foro
- Documentaciones
- Tutoriales

6.3. Nimbits

Nimbits es un servicio de registro de datos y Plataforma para conectar sensores y software a la nube. Nimbits aporta para el Internet de las Cosas, proporcionando una plataforma que se puede construir localmente en sistemas embebidos, filtrar ruido, ejecutar reglas y enviar datos importantes hasta la nube.

El servidor de Nimbits registra y procesa datos y ejecuta reglas que se definen en base a la información recibida. Las reglas pueden ser cálculos, estadísticas, alertas de correo electrónico, mensajes XMPP (*Extensible Messaging and Presence Protocol*), notificaciones *push* (en dispositivos móviles) y más.

Es posible descargar el servidor Nimbits para sistemas operativos Linux y construir una infraestructura propia, o por otro lado, se puede registrar a una nube privada y escalarla a cualquier tamaño.

Para este trabajo, se analizan las características de la Nube Pública de Nimbits, la cual es una instancia del servidor de Nimbits con funcionalidades limitadas que corre continuamente y no tiene ningún costo de uso. Nada más se necesita registrar una cuenta local en la plataforma o iniciar sesión con una cuenta de Google.

Nimbits Android está disponible en Google Play. Fue escrito utilizando nimbits.io y expone muchas de las características de un cliente Nimbits, proporcionando gráficos, entrada de datos, alertas y más. Nimbits.io es una librería de Java de código abierto que proporciona una manera fácil de desarrollar soluciones Java, Web y Android que utilizan un servidor Nimbits como plataforma de back-end. (Nimbits, s.f.)

6.3.1. Compatibilidad con Arduino

En el repositorio oficial de Nimbits, se encuentran librerías y *sketches* de Arduino para enviar datos desde una placa Arduino al servidor. Sin embargo, en las últimas versiones, ya no se ha dado un soporte a las librerías de Arduino que trabajan con la nube pública de Nimbits. Se debe optar por registrarse a una nube privada con un nombre propio, por ejemplo sunombre.nimbits.com. Después del pago por instalación de $ 249.95 USD, sólo se paga por los recursos que se consumen, más $ 19.95 USD al mes como una cuota de mantenimiento. (Nimbits, s.f.)

Conclusiones

El desarrollo de la plataforma Nimbits se ha concentrado más en la nube privada y en servidores que corren sobre Linux. Por lo que con su uso no sería posible, por el momento, cumplir con el propósito de este trabajo, y la información que se ha recopilado es puramente informativa ya que no se pueden realizar pruebas de medición de sensores con Arduino por la falta de soporte con este. Determinando así, que las pruebas del prototipo de monitoreo ambiental solo se realizarán con la plataforma ThingSpeak, presentadas en el siguiente capítulo.

7. CAPÍTULO 7: Pruebas y Resultados del Prototipo de Monitoreo Ambiental

Introducción

Después de haber visto los conceptos del Internet de las Cosas, Computación en la Nube, redes WSN y de haber establecido el hardware, sensores y la plataforma en la nube a utilizar, se procede a realizar las pruebas del prototipo propuesto. Comenzando por pruebas de lecturas individuales de los sensores y luego juntarlos para enviar sus datos por medio de Ethernet o WiFi. Finalmente, se analizan los resultados de cada salida de programa realizado.

7.1. Temperatura y Humedad

Para comprobar el funcionamiento del sensor DHT11, se lo conecta a la placa Arduino UNO como se indica en la figura 7.1

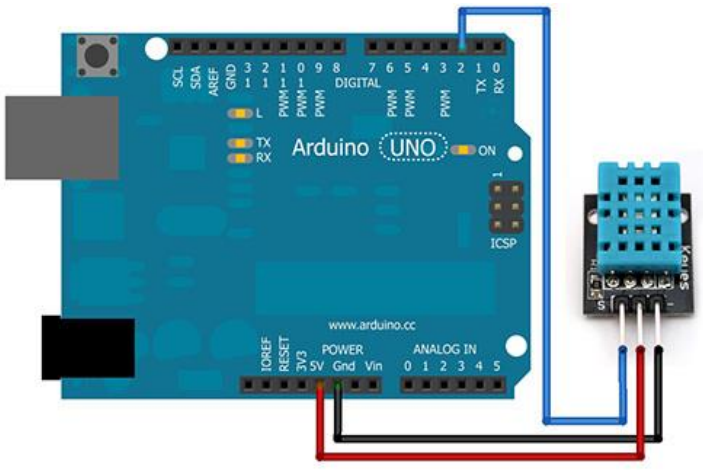


Figura 7.1. Conexión del sensor DHT11 al Arduino Uno. (ChiOSZ robots, s.f.)

7.1.1. Código de Programación

El sketch programado en el Arduino IDE es el siguiente:

```
#define DHTTYPE DHT11   // Versión del sensor.

// Inicializar el sensor DHT para el Arduino normal de 16mhz
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("Prueba DHT11");

  dht.begin();
}

void loop() {
  // Esperar unas segundos entre medidas
  delay(2000);

  // Lectura de temperatura o humedad tarda cerca de 250 ms.

  //Humedad en porcentaje.
  float h = dht.readHumidity();
  //Temperatura en °C
  float t = dht.readTemperature();

  // Comprueba si cualquier lectura falló y sale (a intentarlo de
nuevo).
  if (isnan(h) || isnan(t)) {
    Serial.println("No se pudo leer desde el sensor DHT!");
    return;
  }

  Serial.print("Humedad: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperatura: ");
  Serial.print(t);
  Serial.println(" *C ");
}
```

7.1.2. Salida del Programa

Como fuente de calor, y para fines demostrativos, se utilizó la salida de disipación de calor de una computadora portátil trabajando con su tarjeta de video a casi el 100%. Mediante el monitor serial del IDE de Arduino, se obtiene la salida del programa:

```
Prueba DHT11
Humedad: 50.00 %        Temperatura: 23.00 *C
Humedad: 50.00 %        Temperatura: 23.00 *C
Humedad: 50.00 %        Temperatura: 24.00 *C
Humedad: 50.00 %        Temperatura: 24.00 *C
Humedad: 50.00 %        Temperatura: 24.00 *C
Humedad: 50.00 %        Temperatura: 24.00 *C
Humedad: 50.00 %        Temperatura: 24.00 *C
Humedad: 49.00 %        Temperatura: 25.00 *C
Humedad: 50.00 %        Temperatura: 25.00 *C
Humedad: 49.00 %        Temperatura: 25.00 *C
Humedad: 50.00 %        Temperatura: 25.00 *C
Humedad: 49.00 %        Temperatura: 25.00 *C

                        ...

Humedad: 33.00 %        Temperatura: 38.00 *C
Humedad: 33.00 %        Temperatura: 38.00 *C
Humedad: 32.00 %        Temperatura: 39.00 *C
Humedad: 32.00 %        Temperatura: 39.00 *C
Humedad: 32.00 %        Temperatura: 39.00 *C
Humedad: 32.00 %        Temperatura: 39.00 *C
Humedad: 32.00 %        Temperatura: 39.00 *C
Humedad: 32.00 %        Temperatura: 39.00 *C
Humedad: 31.00 %        Temperatura: 39.00 *C
Humedad: 31.00 %        Temperatura: 39.00 *C
Humedad: 31.00 %        Temperatura: 40.00 *C
Humedad: 31.00 %        Temperatura: 40.00 *C

                        ...

Humedad: 20.00 %        Temperatura: 51.00 *C
Humedad: 20.00 %        Temperatura: 51.00 *C
Humedad: 21.00 %        Temperatura: 51.00 *C
Humedad: 20.00 %        Temperatura: 51.00 *C
No se pudo leer desde el sensor DHT!
No se pudo leer desde el sensor DHT!
```

Listado 1. Respuesta del sensor de humedad y temperatura (a).

En el listado 1, se observa que el sensor empieza a tener problemas al medir los parámetros. Es de esperarse pues corresponde con la especificación técnica la cual indicaba que el rango máximo de medición está cerca de los 50 °C.

A continuación, el sensor es retirado de la fuente de calor y para bajar la temperatura detectada, se la coloca cerca de un contenedor con hielos. Mediante el monitor serial, se obtiene la salida del programa:

```
Prueba DHT11
Humedad: 22.00 %        Temperatura: 45.00 *C
Humedad: 23.00 %        Temperatura: 44.00 *C
Humedad: 23.00 %        Temperatura: 43.00 *C
Humedad: 23.00 %        Temperatura: 43.00 *C
Humedad: 24.00 %        Temperatura: 42.00 *C
Humedad: 24.00 %        Temperatura: 42.00 *C

                        ...

Humedad: 31.00 %        Temperatura: 28.00 *C
Humedad: 31.00 %        Temperatura: 28.00 *C
Humedad: 31.00 %        Temperatura: 27.00 *C
Humedad: 31.00 %        Temperatura: 27.00 *C
Humedad: 31.00 %        Temperatura: 27.00 *C
Humedad: 32.00 %        Temperatura: 26.00 *C
Humedad: 32.00 %        Temperatura: 26.00 *C
Humedad: 32.00 %        Temperatura: 26.00 *C
Humedad: 32.00 %        Temperatura: 26.00 *C
Humedad: 32.00 %        Temperatura: 26.00 *C
Humedad: 32.00 %        Temperatura: 26.00 *C
Humedad: 32.00 %        Temperatura: 26.00 *C
Humedad: 32.00 %        Temperatura: 25.00 *C

                        ...

Humedad: 50.00 %        Temperatura: 4.00 *C
Humedad: 50.00 %        Temperatura: 4.00 *C
Humedad: 49.00 %        Temperatura: 4.00 *C
Humedad: 50.00 %        Temperatura: 4.00 *C
Humedad: 50.00 %        Temperatura: 4.00 *C
```

Listado 2. Respuesta del sensor de humedad y temperatura (b).

Como se observa en el listado 2, la temperatura baja al estar en contacto con el contenedor, sin embargo, la humedad llegó al mismo valor cuando se inició la prueba. Para aumentar el valor de lectura de parámetro, se humedeció con cuidado en sensor. Mediante el monitor serial, se obtiene la salida del programa:

```
Humedad: 66.00 %
Humedad: 67.00 %
Humedad: 67.00 %
Humedad: 68.00 %
Humedad: 68.00 %
Humedad: 68.00 %
Humedad: 69.00 %
Humedad: 69.00 %
Humedad: 69.00 %
Humedad: 70.00 %
Humedad: 70.00 %
Humedad: 70.00 %
Humedad: 71.00 %
Humedad: 71.00 %
Humedad: 71.00 %

...

Humedad: 79.00 %        Temperatura: 14.00 *C
Humedad: 79.00 %        Temperatura: 14.00 *C
Humedad: 79.00 %        Temperatura: 14.00 *C
Humedad: 80.00 %        Temperatura: 14.00 *C
Humedad: 80.00 %        Temperatura: 14.00 *C
Humedad: 80.00 %        Temperatura: 14.00 *C
Humedad: 80.00 %        Temperatura: 14.00 *C
Humedad: 80.00 %        Temperatura: 14.00 *C
```

Listado 3. Respuesta del sensor de humedad y temperatura (c).

7.1.3. Resultados

Las lecturas del sensor de temperatura y humedad, DHT11, subieron y bajaron como se esperó en este experimento. Sin embargo, los valores medidos no corresponden a lo que se asume que son los valores reales. Se detectó una desviación en la temperatura inicial, cercana a +5 °C y en la humedad, de -20% de humedad relativa.

7.2. Concentración de CO

Para comprobar el funcionamiento del sensor MQ-7, se lo conecta a la placa Arduino UNO de manera similar a como se indica en la figura 7.2.

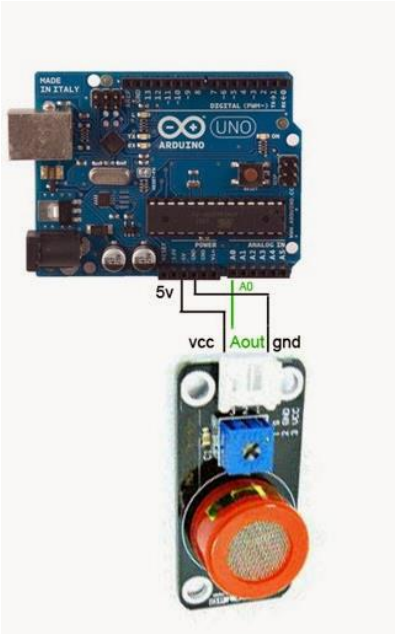


Figura 7.2. Conexión del sensor MQ-7 al Arduino Uno. (DalyBulge, 2015)

7.2.1. Código de programación

El sketch programado en el Arduino IDE es el siguiente:

```
void setup() {

  Serial.begin(9600);
}

void loop() {
  int anRead = analogRead(A2); //A2 Pin al que está conectado la
salida del MQ-7
  int sensorValue = anRead-130; //Se resta del valor del ADC que
equivale a 0 ppm
  if (sensorValue<0){
    sensorValue=0;
  }

  float ppm = sensorValue*(1980/893); //fórmula: valor calibrado
*(rango de medición del sensor / resolución del ADC - 130)

  Serial.print("Valor: ");
  Serial.println(anRead);
  Serial.print("Concentracion de CO: ");
  Serial.print(ppm);
  Serial.println("ppm");
  delay(500);
}
```

7.2.2. Salida del Programa

Se realiza la prueba del sensor al medir el CO de un cigarrillo. Mediante el monitor serial, se obtiene la salida del programa:

```
Valor ADC: 143
Concentracion de CO: 26.00ppm
Valor ADC: 144
Concentracion de CO: 28.00ppm
Valor ADC: 144
Concentracion de CO: 28.00ppm
Valor ADC: 144
Concentracion de CO: 28.00ppm
Valor ADC: 143
Concentracion de CO: 26.00ppm
Valor ADC: 379
Concentracion de CO: 498.00ppm
Valor ADC: 561
Concentracion de CO: 862.00ppm
Valor ADC: 554
Concentracion de CO: 848.00ppm
Valor ADC: 591
Concentracion de CO: 922.00ppm
Valor ADC: 641
Concentracion de CO: 1022.00ppm
Valor ADC: 669
Concentracion de CO: 1078.00ppm
Valor ADC: 669
Concentracion de CO: 1078.00ppm
Valor ADC: 661
Concentracion de CO: 1062.00ppm
Valor ADC: 636
Concentracion de CO: 1012.00ppm
Valor ADC: 600
Concentracion de CO: 940.00ppm
Valor ADC: 607
Concentracion de CO: 954.00ppm
Valor ADC: 594
Concentracion de CO: 928.00ppm
```

Listado 4. Respuesta del sensor de concentración de CO.

7.2.3. Resultados

Con el experimento del cigarrillo, se comprueba que el valor de la lectura incrementa o decrece según el nivel de exposición, pero no tan próximo a valores reales documentados (CO Headquarters, 2001). Al parecer, su aplicación sería más bien orientada a alarmas de detección de fugas de gas o de toxicidad.

Cabe destacar que la aproximación realizada del valor del ADC a ppm se la hizo según la relación entre el rango de sensibilidad del sensor y de la resolución de la placa. Pues las aplicaciones de este sensor normalmente son para detectar cambios en los niveles de concentración de monóxido de carbono, mas no para obtener un valor exacto de ppm, si bien, la hoja de datos del sensor indique una relación entre resistencia y ppm, lo cual no puede plasmarse en el código. Además, el sensor es susceptible, aunque en menor grado, a otros tipos de gases que puedan ser expuestos al mismo.

7.3. Ruido (Sonido)

Para comprobar el funcionamiento del sensor de sonido, se lo conecta a la placa Arduino UNO de manera similar a como se indica en la figura 7.3.

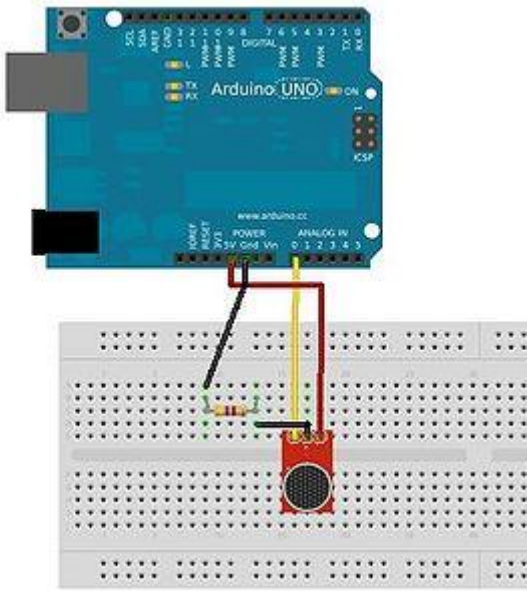


Figura 7.3. Conexión del sensor MQ-7 al Arduino Uno. (YoHa, 2012)

7.3.1. Código de Programación

El sketch programado en el Arduino IDE es el siguiente:

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0); //A0 Pin al que está conectado
la salida del sensor

  float dB = 20*log(sensorValue); //Valor del ADC a dB

  Serial.print("Intensidad de sonido: ");
  Serial.print(dB);
  Serial.println("dB");
  delay(500);
}
```

7.3.2. Salida del Programa

Se realiza la prueba del sensor con para medir la intensidad del sonido al acercar y alejar un parlante aplicado directamente sobre micrófono del sensor. Mediante el monitor serial, se obtiene la salida del programa:

```
Intensidad de sonido: 35.84dB
Intensidad de sonido: 56.66dB
Intensidad de sonido: 55.45dB
Intensidad de sonido: 57.81dB
Intensidad de sonido: 27.73dB
Intensidad de sonido: 57.81dB
Intensidad de sonido: infdB
Intensidad de sonido: 115.43dB
Intensidad de sonido: 108.67dB
Intensidad de sonido: 102.24dB
Intensidad de sonido: 80.15dB
Intensidad de sonido: 102.60dB
Intensidad de sonido: infdB
Intensidad de sonido: 123.64dB
Intensidad de sonido: 124.61dB
Intensidad de sonido: 84.97dB
Intensidad de sonido: infdB
Intensidad de sonido: 124.21dB
Intensidad de sonido: infdB
Intensidad de sonido: 124.84dB
Intensidad de sonido: 125.31dB
Intensidad de sonido: infdB
```

Listado 5. Respuesta del sensor de sonido.

Luego de estas mediciones, se vuelve a realizar mediciones con el sonómetro 3M Quest SoundPro. Midiendo el ruido del ambiente y luego con el calibrador de tono constante de 114 dB de la figura 7.5. Se obtienen los resultados descritos en 7.3.3.

Figura 7.4. Equipamiento del sonómetro 3M Quest SoundPro.

Figura 7.5. Calibrador del sonómetro.

7.3.3. Resultados

Los valores medidos aumentan o decrecen según la intensidad de sonido aplicado. Sin embargo, comparando las lecturas entre el sensor 3M SoundPro y el compatible con Arduino, se determinó que dicho sensor detecta mejor los sonidos de bajas frecuencias, pues como se observa en el listado 5, cuando se lo expone a frecuencias altas, se dispara al infinito el nivel de dB (producto de la medición fuera de rango). Además de que posee poca sensibilidad pues el valor medido posee cerca de 20 a 30 dB menos que el detectado por el SoundPro, un dispositivo mucho más sofisticado y exacto por tener un ángulo de medición cercano a 270°, mientras que el utilizado debe tener la fuente de sonido aplicada directamente sobre su micrófono lo que limita la calidad y exactitud de las lecturas. Se puede concluir que este sensor es de sonido y no de ruido como tal, ya que no se pueden detectar sonidos o ruido a varios rangos de frecuencias. Su empleo sería mejor aplicado en, por ejemplo, encendido/apagado de iluminaciones, con la intensidad del sonido de un aplauso, o en la activación de alarmas de seguridad.

7.4. Envío de los datos al canal de ThingSpeak

Después de conseguir los datos de cada sensor, para mantener el registro de estos, es necesario enviarlos por el internet a una plataforma del Internet de las Cosas. Para ello, se utiliza el shield de Ethernet y el Shield de WiFi para enviar esta información ambiental al canal de ThingSpeak.

7.4.1. Envío mediante Ethernet

Se monta el shield de Ethernet de Arduino sobre la placa Arduino UNO. Para la programación, se emplea la librería del shield correspondiente y se junta el código de los sketches de los sensores probados anteriormente.

7.4.1.1. Código de Programación

El sketch programado en el Arduino IDE es el siguiente:

```
#include <SPI.h>
#include <Ethernet.h> // Librería para comunicarse con el shield de
Ethernet
#include "DHT.h"      // Librería para el sensor de temperatura y
humedad
#define DHTPIN 2      // El pin al que está conectado el sensor
DHT11.
#define DHTTYPE DHT11    //  Versión del sensor de temperatura y
humedad.
DHT dht(DHTPIN, DHTTYPE); // Inicializar el sensor DHT para el
Arduino normal de 16mhz

float Temperatura = 0;
float Humedad = 0;
int sensorValue = 0;
float dB = 0;
float ppm = 0;


// Configuración de la red local
byte mac[] = { 0xD4, 0x28, 0xB2, 0xFF, 0xA0, 0xA1 }; // Debe ser
único en la red local

// Configuración de ThingSpeak
char thingSpeakAddress[] = "api.thingspeak.com";
String writeAPIKey = "7OU75INQ8QBJJB9K";        // llave del canal
de ThingSpeak
const int updateThingSpeakInterval = 1 * 1000;   // Intervalo de
actualización

// Configuración de variables
long lastConnectionTime = 0;
boolean lastConnected = false;
int failedCounter = 0;

// Inicializar Cliente Ethernet
EthernetClient client;
//Parámetros necesarios cuando se trabaja con una IP fija.
// Caso contrario, las siguientes 4 líneas deben ir comentadas.
//IPAddress ip(172,16,7,93);
//IPAddress dn(172,16,1,2);
//IPAddress gateway(172,16,1,3);
//IPAddress subnet(255,255,0,0);

void setup()
{
  Serial.begin(9600);
  dht.begin();
  startEthernet();
}
void loop()
{
    //Llamado a las funciones de lectura de los sensores
    tempHumReading();
    soundReading();
    gasReading();
    char buffer[10];
    String sT= dtostrf(Temperatura, 5, 1, buffer);
    String sH = dtostrf(Humedad, 5, 1, buffer);
    String sdB = dtostrf(dB, 5, 1, buffer);
    String sPpm = dtostrf(ppm, 5, 1, buffer);
  // Imprimir respuesta de actualización en el monitor serial
```

```
  if (client.available())
  {
    char c = client.read();
    Serial.print(c);
  }
  // Desconectar de ThingSpeak
  if (!client.connected() && lastConnected)
  {
    Serial.println("...desconectado");
    Serial.println();
    client.stop();
  }

  // Actualizar ThingSpeak
  if(!client.connected()  &&  (millis()  -  lastConnectionTime  >
updateThingSpeakInterval))
  {
    updateThingSpeak("2="+sT+"&1="+sH+"&3="+sdB+"&4="+sPpm);
  }
  // Comprobar si es necesario reiniciar el shield de Ethernet
  if (failedCounter > 3 ) {startEthernet();}
  lastConnected = client.connected();
}

void updateThingSpeak(String tsData)
{
  if (client.connect(thingSpeakAddress, 80))
  {
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: "+writeAPIKey+"\n");
    client.print("Content-Type:            application/x-www-form-
urlencoded\n");
    client.print("Content-Length: ");
    client.print(tsData.length());
    client.print("\n\n");
    client.print(tsData);
    lastConnectionTime = millis();
    if (client.connected())
    {
      Serial.println("Conectando con ThingSpeak...");
      Serial.println();
      failedCounter = 0;
    }
    else
    {
      failedCounter++;
      Serial.println("Conexion     con     ThingSpeak     fallida
("+String(failedCounter, DEC)+")");
      Serial.println();
    }
  }
  else
  {
    failedCounter++;
    Serial.println("Conexion      con      ThingSpeak      fallida
("+String(failedCounter, DEC)+")");
    Serial.println();
    lastConnectionTime = millis();
  }
}

void startEthernet()
{
  client.stop();
  Serial.println("Conectando Arduino a la red...");
  Serial.println();
```

```
  delay(500);

  // Para conectase a la red y obtener una dirección IP usando DHCP,
quitar los comentarios:
  if (Ethernet.begin(mac) == 0)
  {
    Serial.println("DHCP Fallido, resetear Arduino para intentar de
nuevo");
    Serial.println();
  }
  else
  {
    Serial.println("Arduino conectado a la red usando DHCP");
    Serial.println();
  }

 // Para conectase a la red sin DHCP, quitar los comentarios:
 // Ethernet.begin(mac, ip, dn, gateway, subnet);
 // Serial.println(Ethernet.localIP());
  delay(500);
}

//LECTURAS DE LOS SENSORES----------------------------------------
------------------------------

void tempHumReading() // Función para medir temperatura y humedad
{
  //Humedad en porcentaje.
  float h = dht.readHumidity();
  //Temperatura en °C
  float t = dht.readTemperature();

 // Comprueba si cualquier lectura falló y sale (a intentarlo de
nuevo).
  if (isnan(h) || isnan(t)) {
    Serial.println("No se pudo leer desde el sensor DHT!");
    return;
  }
  Temperatura = t;
  Humedad = h;
}

void soundReading(){ // Función para medir intensidad de sonido
  //Se obtiene un promedio de 5 lecturas
  for (int i = 0; i<5; i++){
    sensorValue += analogRead(A0);//A0 Pin al que está conectado la
salida del sensor de sonido
    delay(10);
  }
  sensorValue /= 5;
  dB = 20*log(sensorValue); //Valor del ADC a dB
}

void gasReading(){ // Función para medir concentración de CO
  int anRead = analogRead(A2); //A2 Pin al que está conectado la
salida del MQ-7
  int sensorValue = anRead-130; //Se resta del valor del ADC que
equivale a 0 ppm
  if (sensorValue<0){
    sensorValue=0;
  }
  ppm = sensorValue*(1980/893); //fórmula: valor calibrado *(rango
de medición del sensor / resolución del ADC - 130)
}
```

7.4.1.2. Salida del Programa

Mediante el monitor serial, se obtiene la salida del programa:

```
Conectando Arduino a la red...

Arduino conectado a la red usando DHCP

Conectando con ThingSpeak...

HTTP/1.1 200 OK
Server: nginx/1.7.5
Date: Mon, 03 Aug 2015 06:39:48 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: close
Vary: Accept-Encoding
Status: 200 OK
X-Frame-Options: ALLOWALL
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, PUT, OPTIONS, DELETE, PATCH
Access-Control-Allow-Headers: origin, content-type, X-Requested-With
Access-Control-Max-Age: 1800
ETag: "35f4a8d465e6e1edc05f3d8ab658c551"
Cache-Control: max-age=0, private, must-revalidate
Set-Cookie: request_method=POST; path=/
X-Request-Id: a4c443de-8efd-4314-b6c3-d9404a2d5bc9

2
78
0

...desconectado

Conectando con ThingSpeak...
```

Listado 6. Respuesta al envío de datos por Ethernet.

7.4.1.3. Resultados

La placa Arduino fue capaz de conectarse al internet mediante el shield de Ethernet y los datos de los sensores fueron exitosamente registrados en el canal de ThingSpeak (cada 40 segundos), como se puede comprobar en las siguientes gráficas consultadas desde dicha plataforma, las mismas que también pueden ser visualizadas desde la aplicación móvil para Android.

Accediendo a la dirección http://api.thingspeak.com/channels/27094/feed.json?key=7OU75INQ8QBJJB9K se pueden observar las entradas de datos en el canal.

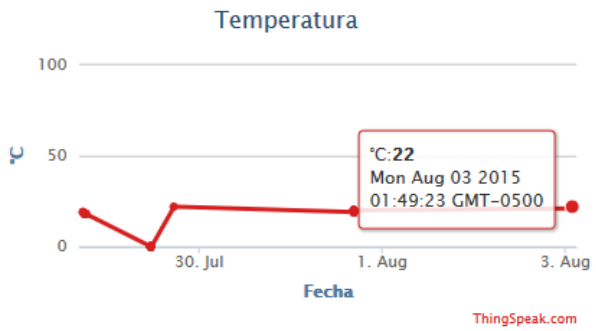


Figura 7.6. Gráfica de Temperatura en ThingSpeak.

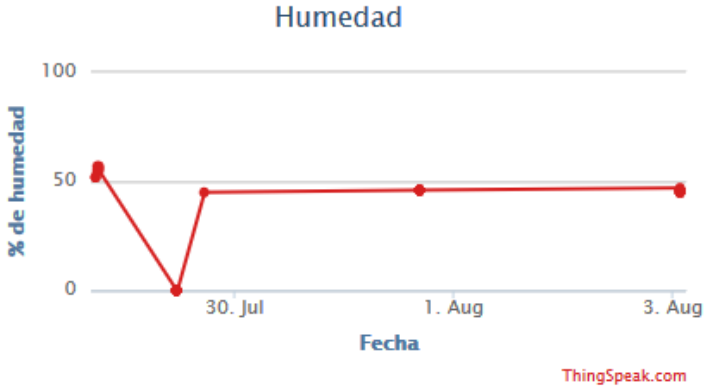


Figura 7.7. Gráfica de % de humedad en ThingSpeak.

Figura 7.8. Gráfica de Intensidad de sonido en ThingSpeak.

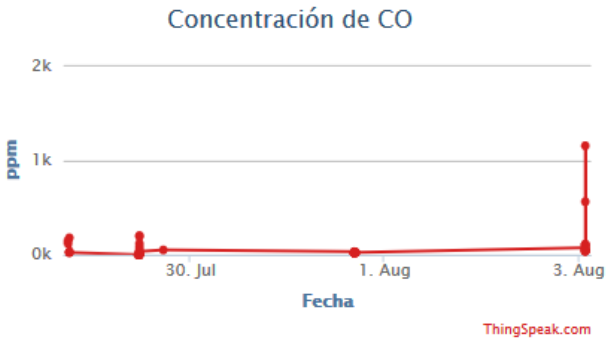


Figura 7.9. Gráfica de concentración de CO en ThingSpeak.

7.4.2. Envío mediante WiFi

Se monta el shield de WiFi de Arduino sobre la placa Arduino UNO, este shield es el que transforma a la mota en un nodo de una red WSN. Para la programación, se emplea la librería del shield correspondiente y se junta el código de los sketches de los sensores probados anteriormente.

7.4.2.1. Código de programación

El sketch programado en el Arduino IDE es el siguiente:

```
#include <WiFi.h>  // Librería para comunicarse con el shield de
WiFi
#include <SPI.h>
#include <SD.h>
#include <avr/wdt.h>
#include "DHT.h"      // Librería para el sensor de temperatura y
humedad
#define DHTPIN 2     // El pin al que está conectado el sensor
DHT11.
#define DHTTYPE DHT11   //  Versión del sensor de temperatura y
humedad.
DHT dht(DHTPIN, DHTTYPE); // Inicializar el sensor DHT para el
Arduino normal de 16mhz

float Temperatura = 0;
float Humedad = 0;
int sensorValue = 0;
float dB = 0;
float ppm = 0;


// Configuración de la red Wifi
char ssid[] = "NombreRed"; //   SSID de la red (nombre)
char pass[] = "ClaveRed"; // clave de la red
```

```
int status = WL_IDLE_STATUS; // el estado del radio WiFi
IPAddress ip; // Dirección IP del shield

// Configuración de ThingSpeak y de variables
char thingSpeakAddress[] = "api.thingspeak.com";
String writeAPIKey = "7OU75INQ8QBJJB9K"; // llave del canal de
ThingSpeak
boolean lastConnected = false;
int failedCounter = 0; // variable to store information about how
many times ThingSpeak connection has been failed
int socketCounter = 0; // variable to store information about how
many times Socket not available error has been happened

File logFile;

char server[] = "SMTPSERVER";

//Inicializar Cliente WiFi
WiFiClient client;
void setup()
{
  Serial.begin(9600);
  Serial.println("Arduino va a reiniciar el shield WiFi en
2s...");
  delay(2000);
  pinMode(A5, OUTPUT);
  digitalWrite(A5, LOW); // Reinicia WiFi shield
  delay(500);
  pinMode(A5, INPUT);
  delay(2000);

  dht.begin();

  // Iniciar Wifi en Arduino
  startWiFi();

  Serial.println(F("Configuracion completa!\n"));
}


void loop()
{
    if (!client.connected() && lastConnected)
    {
      Serial.println(F("...desconectado\n"));
      client.stop();
    }
     //Llamado a las funciones de lectura de los sensores
     tempHumReading();
     soundReading();
     gasReading();

    Serial.print(F("Temperatura: "));
    Serial.println(Temperatura);
    Serial.print(F("% Humedad: "));
    Serial.println(Humedad);
    Serial.print(F("dB: "));
    Serial.println(dB);
    Serial.print(F("ppm: "));
    Serial.println(ppm);

    char buffer[10];
    String stringTemperature = dtostrf(Temperatura, 5, 1, buffer);
    String stringHumidity = dtostrf(Humedad, 5, 1, buffer);
    String stringdB = dtostrf(dB, 5, 1, buffer);
    String stringPpm = dtostrf(ppm, 5, 1, buffer);

    // Actualizar ThingSpeak
```

```
updateThingSpeak("field2="+stringTemperature+"&field1="+stringHumi
dity+"&field3="+stringdB+"&field4="+stringPpm);

    // Revisa si el shield de WiFi debe ser reiniciado
    if (failedCounter >= 3 )
    {
      client.flush();
      client.stop();
      socketCounter++;
      Serial.println(F("Reiniciando conexion WiFi en 5s"));
      delay(5000);
      WiFi.disconnect();
      status = 0;
      failedCounter = 0;
      startWiFi();
    }
    lastConnected = client.connected(); // Guarda el estado de la
conexión
   delay(10000);
}

byte eRcv()
{
  byte respCode;
  byte thisByte;
  int loopCount = 0;

  while(!client.available())
  {
    delay(1);
    loopCount++;
    // si no se recibe nada en 10s, timeout
    if(loopCount > 10000)
    {
      client.stop();
      Serial.println(F("\r\nTimeout"));
      return 0;
    }
  }
  respCode = client.peek();
  while(client.available())
  {
    thisByte = client.read();
    Serial.write(thisByte);
  }

  if(respCode >= '4')
  {
    efail();
    return 0;
  }

  return 1;
}


void efail()
{
  byte thisByte = 0;
  int loopCount = 0;

  client.println(F("SALIR"));

  while(!client.available()) {
    delay(1);
    loopCount++;
```

```
    // // si no se recibe nada en 10s, timeout
    if(loopCount > 10000) {
      client.stop();
      Serial.println(F("\r\nTimeout"));
      return;
    }
  }

  while(client.available())
  {
    thisByte = client.read();
    Serial.write(thisByte);
  }
  client.stop();
  Serial.println(F("desconectado"));
}


void updateThingSpeak(String tsData)
{
  if (client.connect(thingSpeakAddress, 80))
  {
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: "+writeAPIKey+"\n");
    client.print("Content-Type: application/x-www-form-
urlencoded\n");
    client.print("Content-Length: ");
    client.print(tsData.length());
    client.print("\n\n");

    client.print(tsData);
    Serial.print(F("Los siguientes datos han sido enviados: "));
    Serial.println(tsData);

    if (client.connected())
    {
      Serial.println(F("Datos enviados!\n"));
      socketCounter = 0;
      failedCounter = 0;
    }
    else
    {
      failedCounter++;
      Serial.println("Conexion con ThingSpeak fallida 1
("+String(failedCounter, DEC)+")\n");
    }

  }
  else
  {
    failedCounter++;
    Serial.println("Conexion con ThingSpeak fallida 2
("+String(failedCounter, DEC)+")\n");
  }
}

void startWiFi()
{
  int resetCounter = 0;

  // Intento de conexión a una red WiFi
  while (status != WL_CONNECTED)
  {
    if(resetCounter >= 3 || socketCounter >= 3)
    {
     Serial.println(F("Reiniciando!"));
```

```
     wdt_enable(WDTO_1S);
    }

    Serial.print(F("Intentando conectarse a la red: "));
    Serial.println(ssid);
    // Conexión a red WPA/WPA2
    Serial.println(F("Esperar 10s para conexion"));
    status = WiFi.begin(ssid, pass);

    // esperar 10 segundos para la conexion:
    delay(10000);
   resetCounter++;
  }

  Serial.print(F("Direccion IP local: "));
  ip = WiFi.localIP();
  Serial.println(ip);
  Serial.println(F("Conectado a la red usando DHCP"));

}
//LECTURAS DE LOS SENSORES----------------------------------------
------------------------------

void tempHumReading() // Función para medir temperatura y humedad
{
  //Humedad en porcentaje.
  float h = dht.readHumidity();
  //Temperatura en °C
  float t = dht.readTemperature();

 // Comprueba si cualquier lectura falló y sale (a intentarlo de
nuevo).
  if (isnan(h) || isnan(t)) {
    Serial.println("No se pudo leer desde el sensor DHT!");
    return;
  }
  Temperatura = t;
  Humedad = h;
}

void soundReading(){ // Función para medir intensidad de sonido
  //Se obtiene un promedio de 5 lecturas
  for (int i = 0; i<5; i++){
    sensorValue += analogRead(A0);//A0 Pin al que está conectado
la salida del sensor de sonido
    delay(10);
  }
  sensorValue /= 5;
  dB = 20*log(sensorValue); //Valor del ADC a dB
}

void gasReading(){ // Función para medir concentración de CO
  int anRead = analogRead(A2); //A2 Pin al que está conectado la
salida del MQ-7
  int sensorValue = anRead-130; //Se resta del valor del ADC que
equivale a 0 ppm
  if (sensorValue<0){
    sensorValue=0;
  }
  ppm = sensorValue*(1980/893); //fórmula: valor calibrado *(rango
de medición del sensor / resolución del ADC - 130)
}
```

7.4.2.2. Salida del Programa

Mediante el monitor serial, se obtiene la salida del programa:

```
Arduino va a reiniciar el shield WiFi en 2s...
Intentando conectarse a la red: MK8
Esperar 10s para conexion
Direccion IP local: 192.168.10.45
Conectado a la red usando DHCP
Configuracion completa!

Temperatura: 23.00
% Humedad: 46.00
dB: 54.16
ppm: 0.00
Los siguientes datos han sido enviados: field2= 23.0&field1= 46.0&field3= 54.2&field4=  0.0
Datos enviados!

...desconectado
```

Listado 7. Respuesta al envío de datos por WiFi.

7.4.2.3. Resultados

La placa Arduino fue capaz de conectarse al internet mediante el shield de WiFi y los datos de los sensores fueron exitosamente registrados en el canal de ThingSpeak (cada 20 segundos), como se puede comprobar en las siguientes gráficas consultadas desde dicha plataforma, las mismas que también pueden ser visualizadas desde la aplicación móvil para Android.

Accediendo a la dirección http://api.thingspeak.com/channels/27094/feed.json?key=7OU75INQ8QBJJB9K se pueden observar las entradas de datos en el canal.

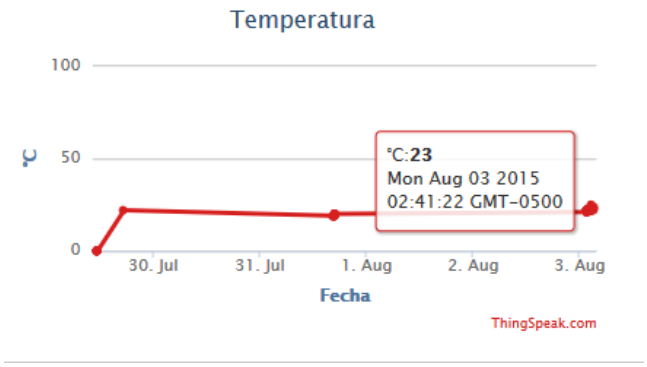


Figura 7.10. Gráfica de Temperatura en ThingSpeak.

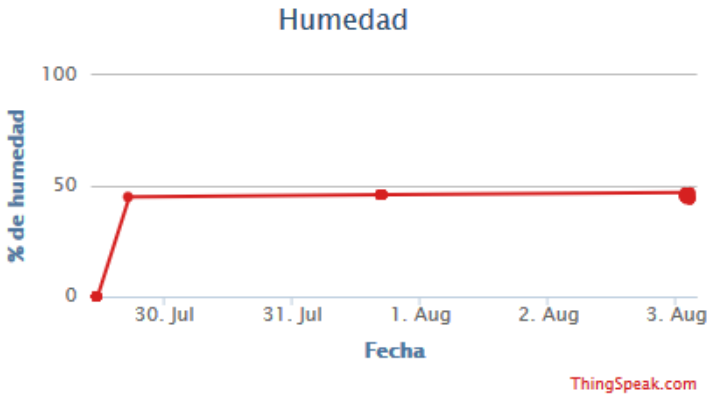


Figura 7.11. Gráfica de % de humedad en ThingSpeak.

Figura 7.12. Gráfica de Intensidad de sonido en ThingSpeak.

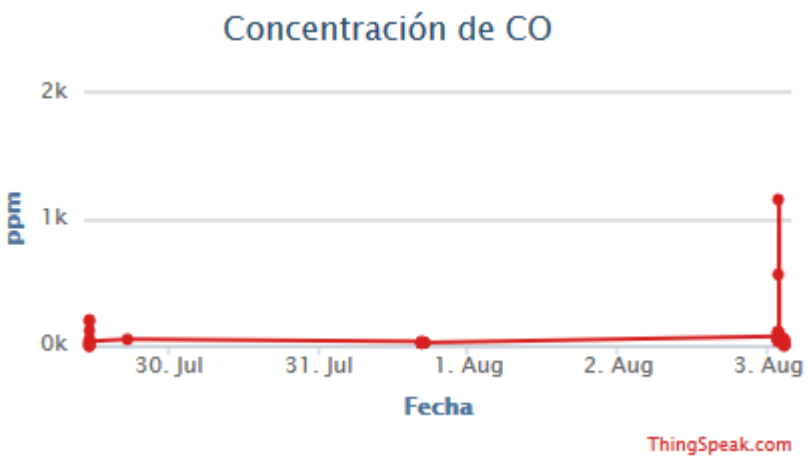


Figura 7.13. Gráfica de concentración de CO en ThingSpeak.

Conclusiones

En este capítulo se probó el funcionamiento de los sensores de temperatura, humedad, sonido y concentración de CO. Si bien los sensores realizan una medición de sus variables ambientales, los valores no corresponden a los que se obtendrían con sensores más sofisticados (ver resultados en 7.1.3., 7.2.3., 7.3.3.) mucho más sensibles a los que se utilizaron en el prototipo realizado.

En cuanto a la conexión internet, los módulos de Arduino permitieron exitosamente la transmisión de datos por Ethernet y WiFi.

8. CAPÍTULO 8. Conclusiones

8.1. Conclusiones Teóricas

En este estudio se ha descrito al Internet de las Cosas, **IoT** por sus siglas en inglés ("*Internet of things*"), como una red global de dispositivos inteligentes que pueden percibir el entorno e interactuar con él, empleando el internet para su comunicación y relación con las personas y con otros sistemas. Para simplificar su entendimiento se ha propuesto la ecuación: *Internet de las Cosas = Objeto físico + Control, sensor y actuadores + internet*.

La IoT permite que los objetos puedan comunicarse con el mundo exterior, a través de un puerto o una antena, operando a cierto valor de frecuencia; percibir su ambiente mediante todo tipo de sensores (movimiento, velocidad, temperatura, humedad); cambiar algo en su ambiente por medio de los actuadores (controladores embebidos); realizar cómputos básicos o complejos (procesadores); y almacenar datos (memoria) para comunicar lo que se ha captado mediante los sensores.

La IoT ha ido ganando importancia con una tendencia creciente, sobre todo en este último lustro, a tal punto que organizaciones como la Unión Internacional de Telecomunicaciones (ITU), la Asociación de Ingenieros Eléctricos y Electrónicos (IEEE), y las agencias normalizadoras ISO/IEC JTC 1 han formado sus propios grupos de trabajo, resoluciones e iniciativas de estándares globales que permitirán a los operadores ofrecer una amplia gama de servicios relacionados con esta plataforma tecnológica.

La necesidad heterogénea de las aplicaciones IoT representa un gran desafío para los investigadores de este campo, puesto que para cada aplicación específica se debe realizar un análisis que establezca el nivel de cada uno de sus requerimientos o condiciones. Los principales desafíos que enfrenta la investigación en este campo, desde el punto de vista de la creación de estas redes, son: su escalabilidad masiva, su alta heterogeneidad e interoperabilidad, las capacidades tecnológicas disponibles limitadas, su alto riesgo de vulnerabilidad de la privacidad y seguridad que requiere una respuesta de diseño que garantice su robustez y resistencia, y la gestión adecuada de una

inmensa cantidad de datos, para convertirlos en información adecuada para la decisión y la acción.

Un aspecto fundamental del Internet de las Cosas es la relación cercana que tiene con las plataformas de Computación en la Nube (*Cloud Computing*). Las aplicaciones IoT están en realidad basadas en Computación en la Nube, de manera que sea posible proveer características de almacenamiento con mejor escalabilidad y gran interoperabilidad a través de accesos abiertos e interfaces directas para la comunicación e intercambio de información. El mejor modo de comunicar las aplicaciones propias con infraestructuras de nube, es mediante servicios web; específicamente, servicios web RESTful, los cuales son muy sencillos y livianos de usar. Por ello resultan ser los más adecuados para microcontroladores y dispositivos.

Una WSN o red de sensores inalámbricos está conformada de nodos, *Gateway* y software. Los nodos interactúan con los sensores y es posible su configuración para manejar su comportamiento y monitoreo mediante programación. Una WSN está constituida por sistemas autónomos desplegados de forma densa y aleatoria para recolectar información sobre un fenómeno en particular como detección y medida de precipitación, temperatura o contaminación, entre otros. Una red WSN se diseña con el propósito de monitorear un lugar o equipo en específico, mediante una infraestructura de nodos, con la finalidad de control o análisis local de cada una de estas redes.

8.2. Conclusiones Metodológicas

El método que se utilizó para este trabajo de titulación consiste en el diseño de un prototipo que integra sensores y sistemas en el ámbito del internet de las cosas. Para el efecto, se aplicó un proceso de diseño basado en la identificación de requisitos, la conceptualización del problema, el análisis, la comparación entre alternativas en el caso de sensores, la selección de soluciones y el diseño y la elaboración del prototipo. La propuesta del dispositivo tecnológico resultante es coherente con las expectativas que se establecen en el Reglamento de Régimen Académico para el nivel de una carrera de grado.

Para la conceptualización del problema fue de particular importancia el consejo que se obtuvo de los miembros del tribunal en la fase de la discusión y aprobación del protocolo del trabajo de titulación. Debe destacarse el acompañamiento que nuestro director nos brindó para decidir los elementos fundamentales que formarían parte de nuestro problema de estudio. Para la fase de comparación de alternativas, una suerte de *benchmarking*, de los sensores que se probaron en el prototipo con respecto a los disponibles para un ámbito profesional, fue de especial ayuda el apoyo del grupo de investigadores de la línea de geomática de la Universidad del Azuay, en el Instituto de Estudios de Régimen Seccional del Ecuador, IERSE.

Una de las maneras más sencillas de conectar los dispositivos al internet es utilizar una tarjeta Arduino, que consiste en un hardware *open-source* utilizado para el desarrollo de prototipos de electrónica que funciona con un software que es también de código abierto. Los protocolos de comunicación más empleados para el Internet de las Cosas son: Ethernet, Bluetooth, 802.11 WiFi, red móvil 3G, 4G, LTE; 802.15.4, 6LowPAN, PLC, NFC, RDIF, entre otros. Para el prototipo de aplicación de IoT de este trabajo, se utilizaron los módulos Ethernet y WiFi.

El prototipo de monitoreo de las variables ambientales de temperatura, humedad, ruido y concentración de monóxido de carbono, aplica los conceptos de las redes de sensores inalámbricos o WSN. Para el efecto, se utiliza una placa de hardware libre Arduino que se comunica por medio de Ethernet y WiFi con el internet, a la plataforma ThingSpeak, para el procesamiento de los datos en la nube. Finalmente, se realiza la gestión de la información que permite mostrar los gráficos del monitoreo en un navegador web.

8.3. Conclusiones Pragmáticas

El Internet de las Cosas demuestra ser un potente medio para vincularse con los objetos y herramientas de uso de la vida diaria. Surge así un modo de realizar ciertas actividades de forma más sencilla y con dispositivos mejorados, donde el ser humano puede simplificarlas y mejorar su estilo de vida,

realizándolas en menor tiempo. Pero no todo es bueno, se tienen muchas vulnerabilidades, que si no se las tratan a tiempo, puede volverse una amenaza para el ser humano y perjudicarlo por medio de robos y ataques a sistemas, oficinas o lugares de trabajo.

Varias industrias están apoyando este nuevo campo de la ciencia para sacar el mayor provecho al momento de desarrollar nuevas tecnologías de software, aplicaciones para dispositivos inteligentes, y nuevos sistemas de seguridad. Es evidente el paso acelerado al que va la evolución de esta tecnología. Sin embargo, se debe estar debidamente capacitado para tomar las debidas precauciones y saber auto protegerse ante cualquier circunstancia de estafa, ataque, *phishing* u otras formas de estafa ya utilizadas en la actualidad.

Una red WSN brinda muchas ventajas al momento de realizar el monitoreo de una zona, es útil para ahorrar recursos y más fácil de mantener comparando con una red de monitoreo a base de cableado. Es capaz de trabajar en ambientes difíciles de acceder, incluso soportar cambios climáticos fuertes, siendo un elemento básico del Internet de las Cosas.

En el prototipo se probó el funcionamiento de los sensores de temperatura, humedad, sonido y concentración de CO. Si bien los sensores realizan una medición de sus variables ambientales, los valores no corresponden a los que se obtendrían con sensores más sofisticados según las pruebas realizadas con un sensor profesional de ruido disponible en el IERSE y la comparación con las tablas de referencia que se encuentran en la literatura especializada. Por ello, para el caso de aplicaciones de carácter profesional, se debería optar por sensores de mayor precisión.

En cuanto a la conexión internet, los módulos de Arduino permitieron exitosamente la transmisión de datos por Ethernet y WiFi. En consecuencia, su uso en entornos profesionales tiene expectativas prometedoras.

Bibliografía

3M. (s.f.). *SoundProUser Manual*. Recuperado el Agosto de 2015, de http://multimedia.3m.com/mws/media/775567O/soundpro-se-dl-series-sound-level-meter-user-manual.pdf

Abubakr, T. (2012). *Cloud app vs. web app: Understanding the differences*. (TechRepublic, Editor) Recuperado el Febrero de 2015, de http://www.techrepublic.com/blog/the-enterprise-cloud/cloud-app-vs-web-app-understanding-the-differences/

Alexander Hellemans. (Marzo de 2015). *Internet of Things Dramatically Increases Demand for IT Specialists at Bosch*. Obtenido de http://spectrum.ieee.org/cars-that-think/transportation/advanced-cars/internet-of-things-dramatically-increases-demand-for-it-specialists-at-bosch

Alexandra Institute. (2011). Inspiring the Internet of Things (IoT Comic Book).

Arduino. (s.f.). *Arduino Board Uno*. Recuperado el Julio de 2015, de https://www.arduino.cc/en/Main/arduinoBoardUno

Arduino. (s.f.). *Arduino Ethernet Shield*. Recuperado el Julio de 2015, de https://www.arduino.cc/en/Main/ArduinoEthernetShield

Arduino. (s.f.). *Arduino Software (IDE)*. Recuperado el Julio de 2015, de https://www.arduino.cc/en/Guide/Environment

Arduino. (s.f.). *Arduino WiFi Shield*. Recuperado el Julio de 2015, de https://www.arduino.cc/en/Main/ArduinoWiFiShield

Arduino. (s.f.). *Introduction to Arduino*. Recuperado el Julio de 2015, de https://www.arduino.cc/en/Guide/Introduction

Arduino. (s.f.). *Sitio Web de Arduino*. Recuperado el 1 de Abril de 2014, de http://www.arduino.cc/

ARM HOLDINNGS. (s.f.). *ARM Holdings Predicts Higher Full-Year Sales on New Clients*. Obtenido de http://www.bloomberg.com/news/articles/2011-10-25/arm-holdings-predicts-higher-full-year-sales-on-new-clients-1-

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., . . . Zaharia, M. (2009). *Electrical Engineering and Computer Sciences. University of California at Berkeley*. Recuperado el Febrero de 2015, de http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf

Beecham Research. (s.f.). *M2M World of Connected Services*. Recuperado el Junio de 2015, de http://beechamtech.com/wp-content/uploads/2013/10/M2M1.jpg

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (Agosto de 2012). *Fog Computing and Its Role in the Internet of Things.* Recuperado el Octubre de 2014, de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.362.9132&rep=rep1&type=pdf

Bosch. (s.f.). *Bosch*. Obtenido de http://www.bosch.us/en/us/startpage_1/country-landingpage.php

Chiang, M. (2015). *Fog Networks and the Internet of Things.* Recuperado el Junio de 2015, de https://www.coursera.org/course/fog

ChiOSZ robots. (s.f.). *DHT11 Digital Temperature Humidity Sensor Module Keyes*. Recuperado el Agosto de 2015, de http://chioszrobots.com/2013/07/15/dht11-digital-temperature-humidity-sensor-module-keyes/

CISCO. (2014). *Connections Counter: The Internet of Everything in Motion*. Recuperado el Julio de 2014, de http://newsroom.cisco.com/feature/1208342/Connections-Counter-The-Internet-of-Everything-in-Motio_2

CO Headquarters. (2001). *CO EXPOSURES AND SCALE OF EFFECTS FROM ZERO TO ONE MILLION PARTS PER MILLION (ppm)*. Recuperado el Agosto de 2015, de http://www.coheadquarters.com/ZerotoMillion1.htm

CyberLightning. (2013). *CyberLightning*. Obtenido de http://www.goodnewsfinland.com/archive/news/finnish-companies-to-participate-in-building-a-new-european-internet-platform/

DalyBulge. (Febrero de 2015). *Ghetto Real-time Carbon Monoxide Data Logger*. Recuperado el Agosto de 2015, de http://dalybulge.blogspot.com/2014/02/ghetto-real-time-carbon-monoxide-data.html

Doukas, C. (2012). *Building Internet of things with the Arduino.* Lexington. Recuperado el 1 de Abril de 2014, de http://www.buildinginternetofthings.com/wp-content/uploads/INTRODUCTION.pdf

D-Robotics. (2010). *DHT11 Humidity & Temperature Sensor.* Recuperado el Julio de 2015, de http://www.micropik.com/PDF/dht11.pdf

EDUCASE. (s.f.). *EDUCAUSE Library CLOUD COMPUTING*. Recuperado el Febrero de 2015, de http://www.educause.edu/library/cloud-computing

Efrati, A., & Nellis, S. (Mayo de 2015). *Google Developing 'Brillo' Software for Internet of Things*. Obtenido de https://www.theinformation.com/Google-Developing-Brillo-Software-for-Internet-of-Things

Elechouse. (s.f.). *Sound Sensor for Arduino.* Recuperado el Julio de 2015, de http://www.elechouse.com/elechouse/index.php?main_page=product_info&cPath=152_162&products_id=2140&zenid=e1lce859qi6pdetivv3nkgpu35

Emartee. (s.f.). *Wrobot DHT11 Analog Temperature & Humidity Sensor*. Recuperado el Julio de 2015, de http://www.emartee.com/product/42184/Wrobot%20DHT11%20Analog%20Temperature%20&%20Humidity%20Sensor

Gartner. (2014). *Gartner Hype Cycle*. Recuperado el Junio de 2015, de http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp

Gartner. (2014). *Gartner's 2014 Hype Cycle for Emerging Technologies Maps the Journey to Digital Business*. Recuperado el Junio de 2015, de http://www.gartner.com/newsroom/id/2819918

Goldman Sachs. (Septiembre de 2014). *The Internet of Things: Making sense of the next mega-trend.* Recuperado el Abril de 2015, de http://www.goldmansachs.com/our-thinking/pages/internet-of-things/iot-report.pdf

Google Trends. (2014). *Interés a lo largo del tiempo para "Internet of Things"*. Recuperado el 2014, de https://www.google.com/trends/explore#q=INTERNET%20OF%20THINGS

Grau, A. (Febrero de 2015). *How to Build a Safer Internet of Things*. Obtenido de http://spectrum.ieee.org/telecom/security/how-to-build-a-safer-internet-of-things

Hanwei Electronics. (s.f.). *TECHNICAL DATA MQ-7 GAS SENSOR.* Recuperado el Julio de 2015, de https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf

Hardy, Q. (Junio de 2014). *The Era of Cloud Computing*. (T. N. Times, Editor) Recuperado el Febrero de 2015, de http://bits.blogs.nytimes.com/2014/06/11/the-era-of-cloud-computing/?_r=2

Hewlett-Packard. (2014). *Internet of Things Research Study.* Obtenido de http://www8.hp.com/h20195/V2/GetPDF.aspx/4AA5-4759ENW.pdf

Hsu, J. (Julio de 2015). *Google Funds University Living Lab for Internet of Things*. Obtenido de http://spectrum.ieee.org/tech-talk/telecom/internet/google-funds-university-living-lab-for-internet-of-things

Hsu, J. (Abril de 2015). *IBM Bets $3 Billion on the Internet of Things*. Obtenido de http://spectrum.ieee.org/tech-talk/telecom/internet/ibm-bets-3-billion-on-the-internet-of-things

IBM. (s.f.). *IBM Event Readiness, un servicio para los momentos de mucha demanda*. Obtenido de http://www.channelbiz.es/2015/06/24/ibm-event-readiness-un-servicio-para-los-momentos-de-mucha-demanda/

IEEE. (s.f.). *Internet of Things*. Recuperado el Julio de 2014, de http://standards.ieee.org/innovate/iot/

iobridge ThingSpeak. (s.f.). *ThingSpeak github repository.* Recuperado el Agosto de 2015, de https://github.com/iobridge/thingspeak/blob/master/README.textile

ISO/IEC. (s.f.). *Resolutions Adopted at the 27th Meeting of ISO/IEC JTC 1*. Recuperado el Julio de 2014, de

http://publicaa.ansi.org/sites/apdl/Documents/News%20and%20Publications/Links%20Within%20Stories/JTC%201%20November%20Resolutions.pdf

ITU-T. (s.f.). *Global Standards for the Internet of Things*. Recuperado el Junio de 2015, de http://www.itu.int/en/ITU-T/techwatch/Pages/internetofthings.aspx

Mattern, F., & Floerkemeier, C. (s.f.). *From the Internet of Computers to the Internet of Things.* Recuperado el 1 de Abril de 2014, de http://www.vs.inf.ethz.ch/publ/papers/Internet-of-things.pdf

MC Electronics. (2014). *Desarrolle Aplicaciones para Internet of Things.* MC Electronics.

McEwen, A., & Cassimally, H. (2014). *Designing the Internet of Things.* Chichester, West Sussex, Reino Unido: Wiley. Obtenido de http://site.ebrary.com/id/10784816

Miller, R. (Mayo de 2015). *Google announces Brillo, an operating system for the Internet of Things*. Obtenido de http://www.theverge.com/2015/5/28/8677119/google-project-brillo-iot-google-io-2015

Mueller, J. (2013). *Understanding SOAP and REST Basics And Differences*. Recuperado el Junio de 2015, de http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/

National Instruments. (s.f.). *¿Qué es una Red Inalámbrica de Sensores?* Recuperado el Julio de 2015, de http://www.ni.com/wsn/whatis/esa/

National instruments. (2015). *Gateway de Red Inalámbrica de Sensores (WSN) de la Serie C*. Obtenido de http://sine.ni.com/nips/cds/view/p/lang/es/nid/210007

National Instruments. (2015). *Gateway Ethernet WSN*. Obtenido de http://sine.ni.com/nips/cds/view/p/lang/es/nid/206919

National Instruments. (2015). *Gateway WSN Programable*. Obtenido de http://sine.ni.com/nips/cds/view/p/lang/es/nid/208440

National Instruments. (2015). *Nodos de Medida de Redes Inalámbricas de Sensores (WSN)*. Obtenido de http://sine.ni.com/nips/cds/view/p/lang/es/nid/206915

National Instruments. (2015). *Nodos de Medida Programables de Redes Inalámbricas de Sensores (WSN)*. Obtenido de http://sine.ni.com/nips/cds/view/p/lang/es/nid/207087

Nimbits. (s.f.). *Nimbits Index Page*. Recuperado el Agosto de 2015, de http://www.nimbits.com/index.jsp

Nimbits. (s.f.). *Register a Private Cloud*. Recuperado el Agosto de 2015, de http://www.nimbits.com/registercloud.jsp

Open Mind. (Abril de 2015). *Internet of Things: Opportunities and Challenges*. Obtenido de https://www.bbvaopenmind.com/en/internet-of-things-opportunities-and-challenges/

Open Mind. (s.f.). *8 Myths and Facts about IOT*. Obtenido de https://www.bbvaopenmind.com/en/8-myths-and-facts-about-iot/

Open Mind. (s.f.). *Internet of Things: Opportunities and Challenges*. Obtenido de https://www.bbvaopenmind.com/en/internet-of-things-opportunities-and-challenges/

Patel, N. V. (Octubre de 2014). *The Internet of Things Gets a New OS*. Obtenido de http://spectrum.ieee.org/tech-talk/computing/embedded-systems/the-internet-of-things-gets-a-new-os

Pew Research Center. (Mayo de 2014). *Report: The Internet of Things Will Thrive by 2025*. Recuperado el Octubre de 2014, de http://www.pewinternet.org/2014/05/14/internet-of-things/

Samaniego Armijos, H. A. (Junio de 2015). Red de Sensores HS. *Seminario Cieela*. Cuenca.

Santos, N., Gummadi, K. P., & Rodrigues, R. (s.f.). *CiteSeerX*. Obtenido de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.167.7563&rep=rep1&type=pdf

Shenzhen Shanhai Technology Ltd. (s.f.). *Wrobot Mini Combustible CO MQ7 Gas Sensor*. Recuperado el Julio de 2015, de http://hkshanhai-shop.sell.curiousexpeditions.org/pz5376990-wrobot-mini-combustible-co-mq7-gas-sensor.html

Tech Crunch. (Enero de 2014). *Google Is Buying Connected Device Company Nest For $3.2B In Cash*. Recuperado el Julio de 2014, de http://techcrunch.com/2014/01/13/google-just-bought-connected-device-company-nest-for-3-2b-in-cash/

ThingSpeak. (s.f.). *Platform Documentation*. Recuperado el Agosto de 2015

ToWebOffice. (s.f.). *Arduino pinout*. Recuperado el Julio de 2015, de http://toweboffice.com/wo/wp-content/uploads/2012/07/arduino_uno_2.jpg

Unión Internacional de Telecomunicaciones. (s.f.). *Internet of Things Global Standards Initiative*. Recuperado el Julio de 2014, de http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx

YoHa. (2012). *Arduino Dinner Hostess: Tutorial*. Recuperado el Agosto de 2015, de http://reboot.yoha.co.uk/index.php?title=Arduino_Dinner_Hostess:_Tutorial&oldid=649#1.3_Inconsistent_values_returned_from_Serial_Monitor

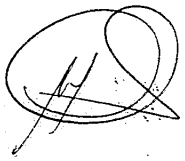
ZDNet. (Enero de 2014). *The Internet of Things might not be what you're hoping for*. Recuperado el Julio de 2014, de http://www.zdnet.com/the-internet-of-things-might-not-be-what-youre-hoping-for-7000025639/

Doctora Jenny Ríos Coello, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay,

C E R T I F I C A:

Que, el H. Consejo de Facultad en sesión realizada el 08 de enero del 201, conoció la petición de los estudiantes **Francisco David Salgado Castillo** con código48375 y **David Santiago Coello Moncayo** con código, que denuncian su trabajo de titulación: (tesis): **"PROTOTIPO DE MONITOREO AMBIENTAL APLICANDO LA INTERNET DE LAS COSAS"** previa a la obtención del Grado de Ingeniero de Sistemas y Telemática. El Consejo de Facultad acoge el informe de la Junta Académica y aprueba la denuncia de tesis. Designa como Director al ingeniero Oswaldo Merchán Manzano y como miembros del Tribunal Examinador a los ingenieros Chester Sellers Walden y Kenneth Palacio Baus. El peticionario tiene un plazo equivalente a dos períodos académicos (semestres) para desarrollar y terminar su trabajo de titulación, a partir de la fecha de la finalización de sus estudios.

Cuenca, enero 9 de 2015



CONVOCATORIA

Por disposición de la Junta Académica de Ingeniería de Sistemas **CONVOCO** a los Miembros del Tribunal Examinador a la sustentación del Protocolo del Trabajo de Titulación denominado: **"PROTOTIPO DE MONITOREO AMBIENTAL APLICANDO LA INTERNET DE LAS COSAS CON ARDUINO Y CLOUD COMPUTING"** presentado por los señores **FRANCISCO DAVID SALGADO CASTILLO (48375)** y **DAVID SANTIAGO COELLO MONCAYO (38093)** previa a la obtención del grado de Ingeniero en Sistemas, para el día **JUEVES 20 DE NOVIEMBRE DE 2014, a las 18H30**

Cuenca, 14 de noviembre de 2014

Dr. Romel Machado Clavijo
Secretario de la Facultad

Ing. Oswaldo Merchán Manzano

Ing. Chester Sellers

Ing. Kenneth Palacios

Comunicado Salgado

Oficio Nro. 129-2014-DIST-UDA

Cuenca, 12 de Noviembre de 2014

Señor Ingeniero
Xavier Ortega Vázquez
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMNISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 12 de Noviembre del 2014, recibió el proyecto de tesis titulado "Prototipo de monitoreo ambiental aplicando la "Internet de las cosas" con Arduino y Cloud Computing", presentada por los estudiantes Francisco Salgado Castillo y David Cuello Moncayo, estudiantes de la Escuela de Ingeniería de Sistemas y Sistemas y revisado por el Ing. Oswaldo Merchán previo a la obtención del título de Ingeniero de Sistemas y Telemática.

La Junta solicita por su digno intermedio notificar al tribunal designado y determinar lugar, fecha y hora de sustentación.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomienda como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Oswaldo Merchán y como miembros del Tribunal al Ing. Chester Sellers y Ing. Kenneth Palacio.

Atentamente,

Ing. Marcos Orellana Cordero
Director Escuela de Ingeniería de Sistemas y Telemática
Universidad del Azuay

Sustentación del Diseño de Tesis (DOCTOR ROMEL MACHADO)

Fecha: 14-11-2014

ESCUELA DE INGENIERIA DE SISTEMAS

Diseños de Tesis
Escuela de Ingeniería de Sistemas

Estudiante: Francisco David Salgado Castillo con código 48375 y David Santiago Coello Moncayo con código 38093.
Tema: "PROTOTIPO DE MONITOREO AMBIENTAL APLICADO LA INTERNET DE LAS COSAS CON ARDUINO Y CLOUD COMPUTING"
Para: La obtención del título de Ingenieros de Sistemas
Director: Ing. Oswaldo Merchán
Tribunal: Ing. Chester Sellers
Tribunal: Ing. Kenneth Palacios

DIA: Jueves

FECHA: 20 - Nov / 2014

HORA: 18½30

ACTA

SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

1.1.1. **Nombre del estudiante:** FRANCISCO DAVID SALGADO CASTILLO y DAVID SANTIAGO COELLO MONCAYO

1.1.2. Código 48375 y 38093

1.1.3. **Director sugerido:** Ing. Oswaldo Merchán Manzano

1.1.4. **1.1.3 Codirector (opcional):** _____

1.1 Tribunal: Ing. Chester Sellers y Kenneth Palacios

1.2 Título propuesto: PROTOTIPO DE MONITOREO AMBIENTAL APLICANDO LA INTERNET DE LAS COSAS CON ARDUINO Y CLOUD COMPUTING

1.3 Resolución:

1.3.1 Aceptado sin modificaciones ✓

1.3.2 Aceptado con las siguientes modificaciones:

_____

_____

_____

1.1.1 Responsable de dar seguimiento a las modificaciones (designado por la Junta Académica de entre los Miembros del Tribunal): Ing. Oswaldo Manzano Merchán

1.1.2 No aceptado

- Justificación:

_____

_____

_____

Tribunal

Ing. Oswaldo Manzano Merchán

Ing. Chester Sellers

Ing. Kenneth Palacios

Sr. Francisco D. Salgado C.

Sr. David S. Coello M.

Secretario de Facultad

Fecha de sustentación: 20/Nov/ 2014

RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN

1.1.1. **1.1Nombre del estudiante:** FRANCISCO DAVID SALGADO CASTILLO y DAVID SANTIAGO COELLO MONCAYO

1.1.2. Código 48375 y 38093

1.1.3. **Director sugerido:** Ing. Oswaldo Merchán Manzano

1.1.4. 1.3 **Codirector (opcional):**

1.1 1.4. **Título propuesto:** PROTOTIPO DE MONITOREO AMBIENTAL APLICANDO LA INTERNET DE LAS COSAS CON ARDUINO Y CLOUD COMPUTING

1.2 Revisores (tribunal): Ing. Chester Sellers y Kenneth Palacios

1.3 Recomendaciones generales de la revisión:

	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
Línea de investigación				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?	✓			
Título Propuesto				
2. ¿Es informativo?				
3. ¿Es conciso?	✓			
Estado del arte				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?	✓			
5. ¿Describe la teoría en la que se enmarca el trabajo	✓			
6. ¿Describe los trabajos relacionados más relevantes?	✓			
7. ¿Utiliza citas bibliográficas?	✓			
Problemática y/o pregunta de investigación				
8. ¿Presenta una descripción precisa y clara?	✓			
9. ¿Tiene relevancia profesional y social?	✓			
Hipótesis (opcional)				
10. ¿Se expresa de forma clara?	✓			
11. ¿Es factible de verificación?	✓			
Objetivo general				
12. ¿Concuerda con el problema formulado?	✓			
13. ¿Se encuentra redactado en tiempo verbal infinitivo?	✓			
Objetivos específicos				
14. ¿Concuerdan con el objetivo general?	✓			
15. ¿Son comprobables cualitativa o cuantitativamente?	✓			
Metodología				
16. ¿Se encuentran disponibles	✓			

los datos y materiales mencionados?				
17. ¿Las actividades se presentan siguiendo una secuencia lógica?	✓			
18. ¿Las actividades permitirán la consecución de los objetivos específicos planteados?	✓			
19. ¿Los datos, materiales y actividades mencionadas son adecuados para resolver el problema formulado?	✓			
Resultados esperados				
20. ¿Son relevantes para resolver o contribuir con el problema formulado?	✓			
21. ¿Concuerdan con los objetivos específicos?	✓			
22. ¿Se detalla la forma de presentación de los resultados?	✓			
23. ¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionadas?	✓			
Supuestos y riesgos				
24. ¿Se mencionan los supuestos y riesgos más relevantes?	✓			
25. ¿Es conveniente llevar a cabo el trabajo dado los supuestos y riesgos mencionados?	✓			
Presupuesto				
26. ¿El presupuesto es razonable?	✓			
27. ¿Se consideran los rubros más relevantes?	✓			
Cronograma				
28. ¿Los plazos para las actividades son realistas?	✓			
Referencias				
29. ¿Se siguen las recomendaciones de normas internacionales para citar?	✓			
Expresión escrita				
30. ¿La redacción es clara y fácilmente comprensible?	✓			
31. ¿El texto se encuentra libre de faltas ortográficas?	✓			

(*) Breve justificación, explicación o recomendación.

- Opcional cuando cumple totalmente,
- Obligatorio cuando cumple parcialmente y NO cumple.

.....

.....

.....

.....

Ing. Oswaldo Merchán Manzano

Ing. Chester Sellers

Ing. Kenneth Palacios

UNIVERSIDAD DEL AZUAY

DOCTORA JENNY RIOS COELLO SECRETARIA DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACION DE LA UNIVERSIDAD DEL AZUAY.

CERTIFICA:

Que, el señor **Francisco David Salgado Castillo,** con código 48375 alumno de la Escuela de Ingeniería de Sistemas y Telemática, tiene aprobado más del 80% de su plan de estudios.

Cuenca, Noviembre 6 del 2014

UNIVERSIDAD DEL AZUAY
FACULTAD DE ADMINISTRACION
SECRETARIA

No. Derecho 068545
rgp.-

UNIVERSIDAD DEL
AZUAY

DOCTORA JENNY RIOS COELLO SECRETARIA DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACION DE LA UNIVERSIDAD DEL AZUAY.

CERTIFICA:

Que, el señor **David Santiago Coello Moncayo,** con código 38093 alumno de la Escuela de Ingeniería de Sistemas y Telemática, tiene aprobado más del 80% de su plan de estudios.

Cuenca, Noviembre 6 del 2014

UNIVERSIDAD DEL AZUAY
FACULTAD DE ADMINISTRACION
SECRETARIA

No. Derecho 068544
rgp.-

Edicion autorizada de 20.000 ejemplares
Del 623.501 al 678.500
No. 0658896

UNIVERSIDAD DEL AZUAY

Cuenca, noviembre 12 de 2014

Ingeniero

Xavier Ortega Vásquez

Decano de la Facultad de Ciencias de la Administración

Ciudad

Señor Decano:

Nosotros, Francisco David Salgado Castillo y David Santiago Coello Moncayo, estudiantes de la Escuela de Ingeniería de Sistemas y Telemática con los códigos 48375 y 38093, respectivamente, presentamos a usted la denuncia del trabajo de titulación denominado "Prototipo de monitoreo ambiental aplicando la Internet de las cosas con Arduino y Cloud Computing". Adjuntamos también el informe favorable del Prof. Ing. Oswaldo Merchán Manzano, M.Sc., a quien proponemos como Director.

Solicitamos comedidamente se sirva autorizar el trámite correspondiente con el fin de que podamos cumplir con los requisitos reglamentarios para la graduación.

Atentamente,

Francisco Salgado Castillo

David Coello Moncayo

UNIVERSIDAD DEL AZUAY

Cuenca, noviembre 11 de 2013

Ingeniero

Xavier Ortega Vásquez

Decano de la Facultad de Ciencias de la Administración

Presente

De mi consideración:

Por la presente, me permito informarle que he revisado el diseño de tesis presentado por las estudiantes **Francisco David Salgado Castillo** y **David Santiago Coello Moncayo** con el tema *"Prototipo de monitoreo ambiental aplicando la Internet de las cosas con Arduino y Cloud Computing"* como requisito previo para la obtención del título de Ingeniero de Sistemas y Telemática.

Al respecto, el diseño de tesis presenta una estructura teórica, metodológica y técnica coherente, para la obtención de un prototipo funcional, cuyo objetivo es medir variables ambientales y procesar esta información en el internet. Por lo expuesto, emito informe favorable y recomiendo su aprobación.

Atentamente

Oswaldo Merchán Manzano

Oficio Nro. 129-2014-DIST-UDA

Cuenca, 12 de Noviembre de 2014

Señor Ingeniero
Xavier Ortega Vázquez
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMNISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 12 de Noviembre del 2014, recibió el proyecto de tesis titulado "Prototipo de monitoreo ambiental aplicando la "Internet de las cosas" con Arduino y Cloud Computing", presentada por los estudiantes Francisco Salgado Castillo y David Cuello Moncayo, estudiantes de la Escuela de Ingeniería de Sistemas y Sistemas y revisado por el Ing. Oswaldo Merchán previo a la obtención del título de Ingeniero de Sistemas y Telemática.

La Junta solicita por su digno intermedio notificar al tribunal designado y determinar lugar, fecha y hora de sustentación.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomienda como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Oswaldo Merchán y como miembros del Tribunal al Ing. Chester Sellers y Ing. Kenneth Palacio.

Atentamente,

Ing. Marcos Orellana Cordero
Director Escuela de Ingeniería de Sistemas y Telemática
Universidad del Azuay

UNIVERSIDAD DEL AZUAY

Universidad del Azuay

Ingeniería de Sistemas y Telemática

Propuesta de trabajo de titulación

1. Datos generales

1.1 Nombres de los estudiantes: Salgado Castillo, Francisco David

Coello Moncayo, David Santiago

1.1.1 Código: ua048375, ua038093

1.1.2 Contacto:

Teléfono convencional: 2883255

Teléfono celular: 099-953-4219, 0998699950

Correo electrónico: f.d.salgado@ieee.org, dcmiport@hotmail.com

1.2 Director sugerido: Merchán Manzano, Oswaldo. Ingeniero, MBA.

1.2.1 Contacto: omerchan@uazuay.edu.ec

1.3 Co-director sugerido: No aplica.

1.4 Asesor metodológico: No aplica.

1.5 Tribunal designado:

1.6 Aprobación:

1.7 Línea de Investigación de la carrera: "Infraestructura de datos espaciales".

1.7.1 Código UNESCO: 1299.02

1.7.2 Tipo de trabajo: Investigación formativa con prototipo.

1.8 Área de estudio: Computación en la nube.

1.9 Título propuesto:

"Prototipo de monitoreo ambiental aplicando la "Internet de las cosas" con Arduino y Cloud computing"

1.10 Subtítulo: No aplica.

1.11 Estado del proyecto: El trabajo es nuevo e integrador.

2. Contenido

UNIVERSIDAD DEL AZUAY

2.1 Motivación de la investigación:

Se trata de un concepto no tan reciente pero que sin embargo se ha vuelto un tema investigativo importante en la actualidad, y su desarrollo se integra tanto en el desarrollo de software, redes y electrónica.

2.2 Problemática:

El internet de las cosas pretende, básicamente, hacer que las cosas hagan más de lo que se les había propuesto hacer. Estas "cosas" pueden ser cualquier objeto del mundo real, pero son, comúnmente, dispositivos como termostatos, sensores de gas, sensores infrarrojos, etc. Y al lograr que estas cosas se comuniquen con el internet, con aplicaciones web, e incluso entre ellos por medio de su ambiente, se podrían crear servicios que actúen en el mundo real/físico sin intervención (directa) humana. Las posibles aplicaciones son: gestión de desperdicios, planificación urbana, sensores ambientales, herramientas de interacción social, respuesta a emergencias, compras inteligentes, automatización en el hogar, etc.

2.3 Pregunta de investigación: ¿Cómo monitorear variables ambientales a distancia y en tiempo real?

2.4 Resumen: El trabajo de titulación propuesto pretende realizar un circuito prototipo para el monitoreo de variables ambientales como ruido y calidad del aire, desde un abordaje enfocado al concepto de la "Internet de las Cosas" que enuncia que objetos del mundo real, mediante componentes electrónicos, pueden conectarse a la internet y enviar datos de lo que sucede a su alrededor. El prototipo será realizado con la tarjeta de hardware libre Arduino que se comunicará con el internet a las plataformas Nimbits y Thingspeak para el procesamiento de los datos en la nube y posteriormente mostrar los gráficos de monitoreo correspondientes en un navegador web.

2.5 Estado del Arte y marco teórico:

Internet de las cosas, IoT por sus siglas en inglés ("Internet of things"), ha recibido varias denominaciones pero se puede describir como una red global de dispositivos inteligentes que pueden "sentir" e interactuar con su ambiente empleando el internet para su comunicación e interacción con usuarios y otros sistemas. [1]

El uso de la palabra "Internet" en el término de IoT hace referencia a una metáfora: en la misma manera que las personas utilizamos la web hoy en día, en un futuro las cosas también se comunicaran entre ellas, usarán servicios, proveerán datos y generarán valor agregado. El término "Internet of Things" fue popularizado por el Auto-ID Center en el Instituto Tecnológico de Massachusetts (MIT), el cual en 1999 comenzó a diseñar y propagar una compañía con una infraestructura de identificación por radio-frecuencia (RFID). En 2002, su cofundador y antiguo jefe Kevin Ashton fue citado en la revista Forbes diciendo, "Necesitamos una internet de las cosas, una manera estandarizada en que las computadoras puedan entender al mundo real". [2] Este artículo fue titulado "The Internet of Things" y fue el primer uso documentado del término en un sentido literal.

Las funciones de un dispositivo que pertenezca al IoT pueden ser resumidas en las siguientes: [1]

- Colectar y transmitir datos: el dispositivo puede sentir el ambiente (nuestro hogar, nuestro cuerpo) y colectar información relacionada con él (temperatura e iluminación) y transmitirla a diferentes dispositivos (celular, computadora) a al internet.
- Accionar dispositivos basados en triggers: Se puede programar que accione otros dispositivos (apagar las luces o apagar la calefacción) basándose en condiciones establecidas por el usuario.
- Recibir Información: de la red a la que pertenecen o través del internet.
- Asistencia en comunicación: los dispositivos pueden asistir en la comunicación entre otros nodos de la misma red.

Una de las maneras más sencillas de conectar estos dispositivos a la internet es utilizando una tarjeta Arduino, un hardware open-source utilizado para el desarrollo de prototipos de electrónica que incluye software, así mismo, open Source [3]. Con la Arduino se pueden crear estos objetos interactivos o distintos ambientes para permitir la comunicación entre ellos.

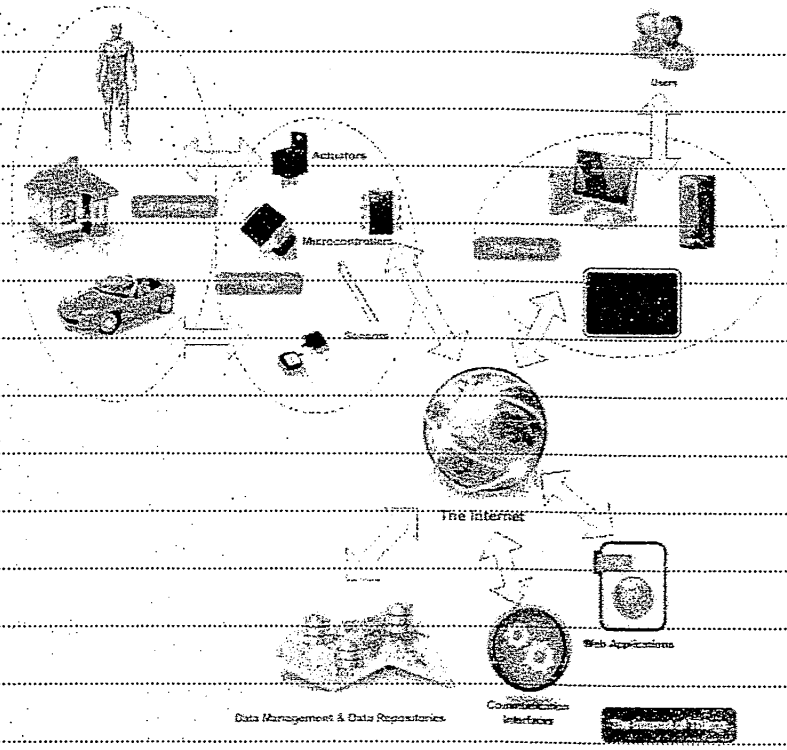


Ilustración del "Internet de las Cosas". Constando las cosas como sensores, actuadores que interactúan con el ambiente y el Internet permitiendo a usuarios gestionarlos y a sus datos sobre varias interfaces.

2.6 Hipótesis: No aplica.

2.7 Objetivo general:

UNIVERSIDAD DEL AZUAY

Desarrollar un prototipo de servicio mediante IoT con sensores conectados al internet por medio de una placa Arduino, y procesar esta información en la nube.

2.8 Objetivos específicos:

- Desarrollar una indagación exploratoria sobre el concepto de la Internet de las Cosas y Computación en la Nube.
- Adquirir y emplear dispositivos y componentes electrónicos a utilizarse en el montaje del circuito.
- Emplear plataformas existentes en la nube para la gestión de aplicaciones de la Internet de las Cosas.
- Medir, mediante el circuito montado, los niveles ambientales de temperatura, CO (calidad del aire) y ruido. Y enviarlos a la nube para ser procesados.
- Recuperar, y presentar la información procesada sobre los datos ambientales desde un navegador web.

2.9 Metodología:

La recopilación y envío de los datos obtenidos de los sensores ambientales serán realizados por la tarjeta Arduino, programada para comunicarse con las plataformas a utilizarse para la gestión de los datos en la nube. Las cuales son: Nimbits y Thingspeak. Las pruebas de funcionamiento se las realizarán en un ambiente que cuente al menos con conexión a internet (wifi o ethernet), enviando los datos monitoreados, a dichas plataformas, en tiempo real.

2.10 Alcances y resultados esperados:

Desarrollar un prototipo de monitoreo ambiental en el cual los sensores de niveles de ruido y calidad del aire, puedan comunicarse con la placa central (Arduino) y enviar datos del ambiente al internet para poder procesarlos y luego visualizar esta información, que a futuro podría utilizarse para desarrollar servicios o aplicaciones que ayuden a la toma de decisiones y tengan un impacto real en el mundo físico.

2.11 Supuestos y riesgos:

- No encontrar los componentes electrónicos necesarios en el mercado local, como los módulos de comunicación inalámbrica, sensores, etc.
- El daño repentino de la tarjeta Arduino o de los demás sensores electrónicos.

2.12 Presupuesto:

Rubro-Denominación	Costo USD (detalle)	Justificación

UNIVERSIDAD DEL AZUAY

Arduino UNO R3 board	22	Microcontrolador principal, realiza toda la gestión de los datos
Arduino ethernet shield	45	Conexión a internet (cableada)
Arduino wifi shield	45	Conexión a internet (inalámbrica)
componentes electrónicos variados	30	Para construir los sensores

2.13 Financiamiento: Particular a cargo de los estudiantes.

2.14 Esquema tentativo:

El internet de las cosas

- Definición
- Historia
- Aplicaciones

Computación en la Nube

Desarrollo del Circuito con Arduino

- Hardware
- Software

Procesamiento de los datos en Nimbits/ThingSpeak

- Pruebas
- Resultados

Conclusiones

2.15 Cronograma:

Objetivo Específico	Actividad	Resultado esperado	Tiempo (semanas)
Desarrollar una indagación exploratoria sobre el concepto de la Internet de las Cosas y Computación en la Nube.	Investigación sobre el Internet de las Cosas y Computación en la Nube	Conocimiento necesario para el marco teórico del trabajo de graduación.	3
Adquirir y emplear dispositivos y componentes	Búsqueda del hardware a utilizarse.	Listado de componentes electrónicos a	2

electrónicos a utilizarse en el montaje del circuito.		comprar y realizar presupuesto final.	
	Adquisición de los dispositivos y componentes que cumplan con las características y especificaciones requeridas.	Hardware necesario para empezar las pruebas de conexión y funcionamiento.	3
Emplear plataformas existentes en la nube para la gestión de aplicaciones de la Internet de las Cosas.	Comparar las plataformas disponibles para el procesamiento de datos en la nube y selección de la más apropiada.	Resultados de pruebas de conexión a la nube y de procesamiento de datos.	3
	Selección y prueba de la plataforma en la nube a utilizarse.	Consolidación de la(s) plataforma(s) a utilizarse en el funcionamiento del prototipo.	2
Medir mediante el circuito montado, los niveles ambientales de temperatura, CO (calidad del aire) y ruido. Y enviarlos a la nube para ser procesados.	Montaje del circuito con los componentes adquiridos.	Prototipo físico de monitoreo ambiental.	3
	Configurar la conexión a la nube del circuito para el envío de datos ambientales	Prototipo lógico de monitoreo ambiental.	4
Recuperar, y presentar la información procesada sobre los datos ambientales desde un navegador web.	Elaborar y presentar gráficos estadísticos y los valores medidos de los niveles de ruido y calidad del aire.	Visualización de datos y mensajes de alerta y/o informativos sobre el ambiente monitoreado.	4

2.16 Referencias:

UNIVERSIDAD DEL AZUAY

C. Doukas, «Building Internet of things with the Arduino,» 2012. [En línea]. Available: http://www.buildinginternetofthings.com/wp-content/uploads/INTRODUCTION.pdf. [Último acceso: 1 Abril 2014].

F. Mattern y C. Floerkemeier, «From the Internet of Computers to the Internet of Things,» [En línea]. Available: http://www.vs.inf.ethz.ch/publ/papers/Internet-of-things.pdf. [Último acceso: 1 Abril 2014].

Arduino, «Sitio Web de Arduino,» [En línea]. Available: http://www.arduino.cc/. [Último acceso: 1 Abril 2014].

2.17 Anexos: No aplica.

2.18 Firma de responsabilidad (estudiante)

Francisco Salgado

David Coello

2.19 Firma de responsabilidad (director sugerido)

Ing. Oswaldo Merchán

2.20 Fecha de Entrega: 27-11-2014

Edición autorizada de 20.000 ejemplares
Del 6581501 al 6601500