industriales
etsii

Escuela Técnica Superior de Ingeniería Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Diseño e implementación de un sistema de alarma IoT basada en tecnologías Open Source.

TRABAJO FIN DE MÁSTER

MÁSTER EN SISTEMAS ELECTRÓNICOS E INSTRUMENTACIÓN

Autor: **Francisco José Martínez Moreno**
Director: Andrés Iborra García

Cartagena,30 de Agosto de 2019

Universidad Politécnica de Cartagena

Índice

Índice de figuras

CAPÍTULO 1.

INTRODUCCIÓN

1. Introducción.

La Internet of Things (IoT) es un concepto que cada vez toma más relevancia y que en los últimos años se escucha con mayor frecuencia ganando más popularidad. Así la IoT mejora objetos que antiguamente se conectaban mediante circuito cerrado, como comunicadores, cámaras, sensores, y demás, y les permite comunicarse globalmente mediante el uso de la red de redes. También cabe destacar la importancia del cómputo en la nube, siendo además una de sus características fundamentales.

Por todo ello, la IoT está revolucionando los negocios tradicionales, dando paso a metodologías más ágiles de trabajo donde priman la adquisición y el análisis de los datos, así como la interconexión de multitud de dispositivos.

En el presente trabajo se va a realizar un estudio de la IoT, donde se relacionará el desarrollo de productos innovadores, de bajo coste y tecnológicamente avanzados que puedan ser candidatos a ser usados para emprender un negocio mediante una start-up.

La línea de trabajo que se planteó en el proyecto previo a la realización de este trabajo fin de Máster estaba orientada únicamente bajo una perspectiva de negocio.

Como resultado de los primeros estudios para la realización del TFM, se decidió ampliar el estudio abordando la problemática de desarrollar un prototipo IoT de bajo coste que permitan validar la viabilidad tecnológica de un modelo de negocio propuesto, en un periodo de tiempo razonable.

Para ello se realiza un estudio sobre el uso de plataformas IoT comerciales y hardware libre.

Considerando la complejidad y extensión del trabajo, se ha limitado el alcance del mismo al diseño conceptual de un prototipo y su viabilidad de implementación. Desde un principio del estudio se aprecian los siguientes problemas:

1. La existencia de multitud de plataformas IoT.
2. Dificultades técnicas en el uso de estas herramientas y la necesidad de seguir un procedimiento eficaz para la consecución de los objetivos planteados.

En base a estos problemas se formuló la siguiente hipótesis: “A pesar de que la utilización de hardware libre y plataformas IoT es un proceso complejo y

estructurado, pueden ser utilizados para el desarrollo de prototipos que premitan validar rápidamente un modelo de negocio IoT.

1.1. Objetivos.

El principal objetivo que se pretende alcanzar con este Trabajo de Fin de Máster es demostrar que, mediante el uso de las plataformas software y dispositivos hardware, se puede realizar un producto mínimo viable (PMV) antes del desarrollo de un plan de negocio, es decir, comprobar que mediante el uso de esta tecnología se puede verificar la viabilidad tecnológica de un producto IoT.

La metodología empleada para la elaboración del presente TFM ha sido básicamente de tipo documental y experimental. En primer lugar, se ha recopilado información de fuentes abiertas (Trabajos Fin de Máster, Libros, revistas, etc…) y de entrevistas realizadas a emprendedores. Posteriormente, se ha hecho uso de la experimentación para la evaluación de las principales herramientas que son utilizadas en el desarrollo de productos IoT.

Como resultado de todos estos trabajos se pantearon las siguientes líneas de investigación.

1. Estudiar las principales plataformas IoT software que existen en la actualidad y sus características.
2. Estudiar y analizar los principales dispositivos hardware IoT.
3. Estudair la validez de la arquitectura planteada para corroborar la hipótesis de partida.

Cada una de estas líneas de investigación han sido desarrolladas en los diferentes capítulos de este TFM.

En el siguiente apartado se puede ver la estructura del documento.

1.2. Estructura del documento

El presente TFM consta de los siguientes capítulos:

- Capítulo 1: Contiene una breve introducción y se describen los objetivos, y estructuración del documento.

- Capítulo 2: Contiene la descripción, características, ventajas y desvenajas de las plataformas software y dispositivos hardware utilizados para el desarrollo de productos IoT. Se definen también las partes de las que se compone una arquitectura de IoT, y, por último, se explican los motivos por el cual se han elegido las plataformas y dispositivos (SW y HW) para el desarrollo de un prototipo funcional.

- Capítulo 3: Contiene una descripción detallada del plan de negocio del dispositivo que se va a desarrollar en este TFM, asimismo, se explica cómo influye la viabilidad técnica dentro de la metodología Lean Startup.

- Capítulo 4: Contiene una descripción funcional del dispositivo IoT. Se explican las principales funcionalidades del dispositivo, las diferentes partes que lo componen, como se cablean y cuál es la arquitectura del mismo.

- Capítulo 5: Contiene una descripción muy detallada de los pasos que se han seguido para la elaboración del producto IoT, explicando cómo se han utilizado las herramientas. Incluye también una comparación con otros sistemas similares, describiendo ventajas y desventajas.

- Capítulo 6: Contiene las conclusiones que se han sacado de este TFM, indicando si se han cumplido los objetivos propuestos.

CAPÍTULO 2.

ESTADO DEL ARTE

2. Internet of Things

La necesidad de los últimos años de estar constantemente conectados y poder cuantificar cualquier cosa, ha dado lugar a lo que hoy día se conoce como la IoT.

Generalmente se define IoT como un concepto basado en la interconexión de cualquier producto con otro cualquiera de su alrededor. El objetivo que se busca con el IoT es hacer que todos los objetos físicos que nos rodean se comuniquen entre sí, y por consiguiente, sean más inteligentes e independientes. Para que este concepto tome forma, es necesario que se realice una gran labor de desarrollo tecnológico.

En esta nueva revolución tecnológica, los objetos cotidianos del día a día, evolucionan gracias a la IoT, pasando a estar interconectados y dotándoles de nuevas funcionalidades, pudiéndose controlar y administrar desde tabletas, ordenadores o teléfonos móviles.

Los recientes avances en el ámbito de la IoT aceleran la aparición de plataformas IoT a gran escala. Con ellas se adquieren, se analizan y se procesan los datos en tiempo real para alimentar al ecosistema de soluciones “inteligentes”, o Smart Solutions. El uso de esas plataformas para la sensorización de los objetos que nos rodean supondrá, en el futuro inmediato, una revolución en la forma de obtener información. Con ese enorme volumen de datos se podrá, entre otras muchas cosas, optimizar la gestión de la industria (Smart Industry y Smart Energy), del mundo médico (Smart Health), del hogar (Smart Home), del mundo agrícola (Smart Farming) o, incluso, de una ciudad entera (Smart City).

Como se ha visto, la IoT ofrece una gran variedad de usos y aplicaciones que engloba, prácticamente, todos los ámbitos que comprende la vida humana.

En los siguientes apartados se describirán las plataformas Hardware y Software que permiten trabajar en la IoT (adquisición de datos, procesamiento de los mismos, interconexión de dispositivos, cloud computing, etc…). Tras enumerar y definir todas estas plataformas y herramientas, se explicará la configuración elegida para probar la viabilidad técnica de un producto mínimo viable, pasando por el desarrollo experimental descrito en el capítulo cuatro de este trabajo de fin de Máster y por qué se ha seleccionado esta tecnología.

2.1. Plataformas IoT SW

2.1.1. ThinkSpeak

2.1.1.1. Definición

ThingSpeak™ es una plataforma softWare que ofrece un servicio de análisis basado en la IoT que permite agregar, analizar y visualizar datos de forma directa a partir de flujos de la nube. ThingSpeak permite llevar a cabo visualizaciones inmediatas de los datos enviados por los dispositivos. Además, se puede ejecutar código MATLAB para el análisis y procesamiento de datos online a medida que se reciben. ThingSpeak se utiliza frecuentemente para diseñar prototipos y pruebas de concepto de los sistemas IoT que requieren un análisis de grandes volúmenes de datos (BBVAOpen4U, 2016).

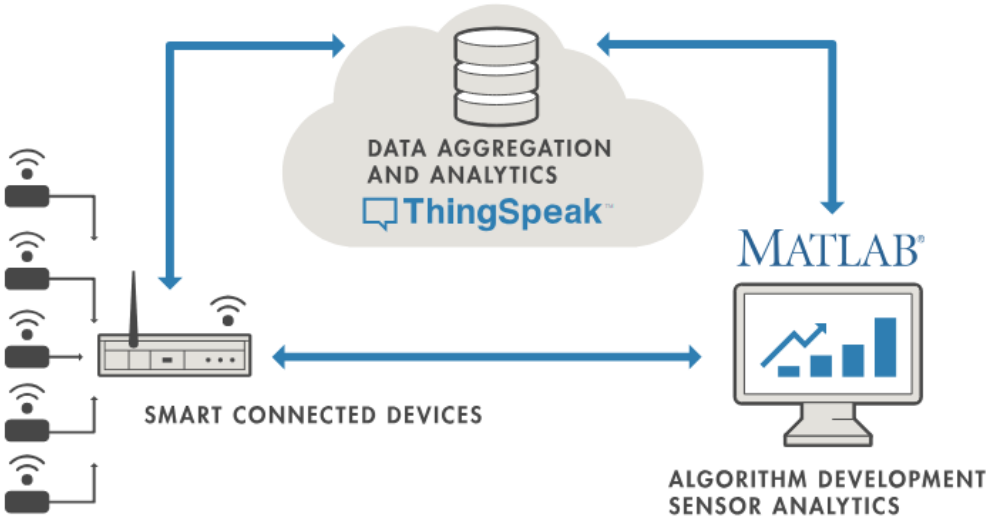


Figura 1: ThingSpeak. (Fuente: https://bbvaopen4u.com/es/actualidad/apis-para-el-internet-de-las-cosas-thingspeak-pachube-y-fitbit).

2.1.1.2. Características

Algunas de las propiedades más relevantes de ThingSpeak son las siguientes:

- Fácil configuración de los dispositivos que envían datos a ThingSpeak.
- Visualización en tiempo real de los datos adquiridos de los diferentes sensores.

- Usar la potencia de MATLAB para procesar los datos.
- Prototipado y construcción de sistemas IoT sin necesidad de servidores o desarrollo software.
- Actuar automáticamente sobre los datos, permitiendo también la comunicación con servicios como Twilio o Twitter.

2.1.2. Altair SmartCore (anteriormente Carriots)

2.1.2.1. Definición

Altair SmartCore forma parte del paquete de Altair SmartWorks. Se trata de una solución de arquitectura abierta. Es una plataforma de nube nativa, que ofrece un conjunto integrado de servicios y características para ayudar a conectar de manera sencilla las cosas con el mundo digital. Está disponible como Paas (Platform as a service). Altair SmartCore permite ayudar a ejecutar los proyectos IoT más rápidamente en un entorno fácil de usar, fiable y altamente escalable (Altair, 2019).

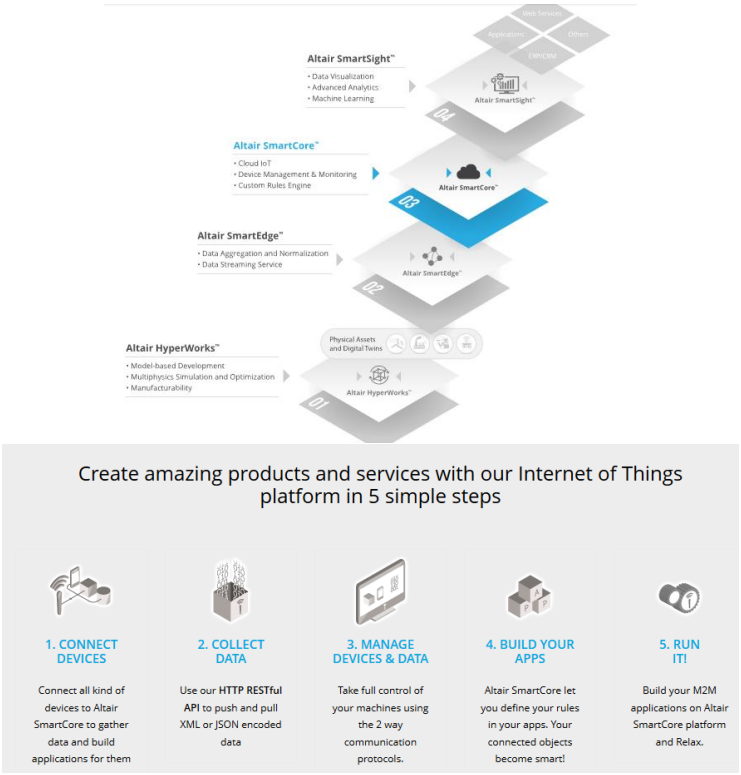


Figura 2: Altair Smartcore. (Fuente:https://partners.sigfox.com/products/altair-smartcore).

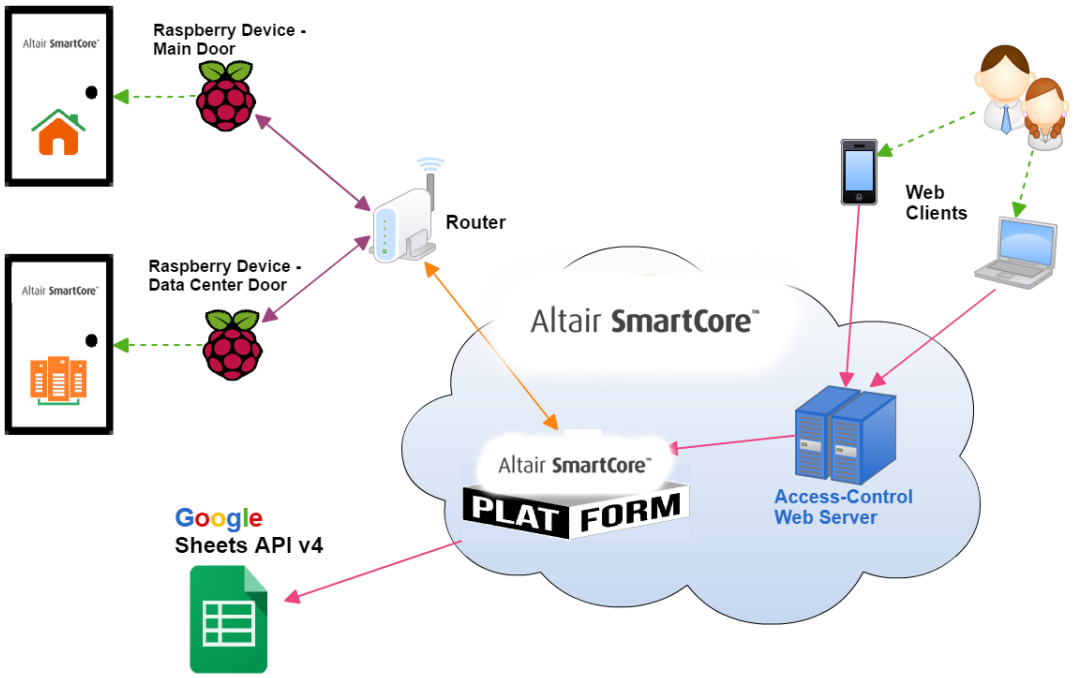


Figura 3: Arquitectura Altair Smartcore.
(Fuente:https://partners.sigfox.com/products/altair-smartcore).

2.1.2.2. Características

- Soporta multitud de dispositivos hardware usados comúnmente para el desarrollo de proyectos basados en IoT, permitiendo la integración rápida de datos.
- Comunicación entre un amplio número de sistemas conectados.
- Alta escalabilidad.
- Elevada rapidez y entorno seguro para el desarrollo de proyectos basados en la IoT gracias a la implementación de protocolos de seguridad.

2.1.3. Electric Imp

2.1.3.1. Definición

Electric Imp es una plataforma software con una arquitectura única de punto a punto, que está completamente integrada por el hardware, los dispositivos y el software de la nube, comunicaciones, APIs, servicios gestionados en la nube y seguridad continua para una oferta completa y lista para dar solución (Electric Imp, 2019).

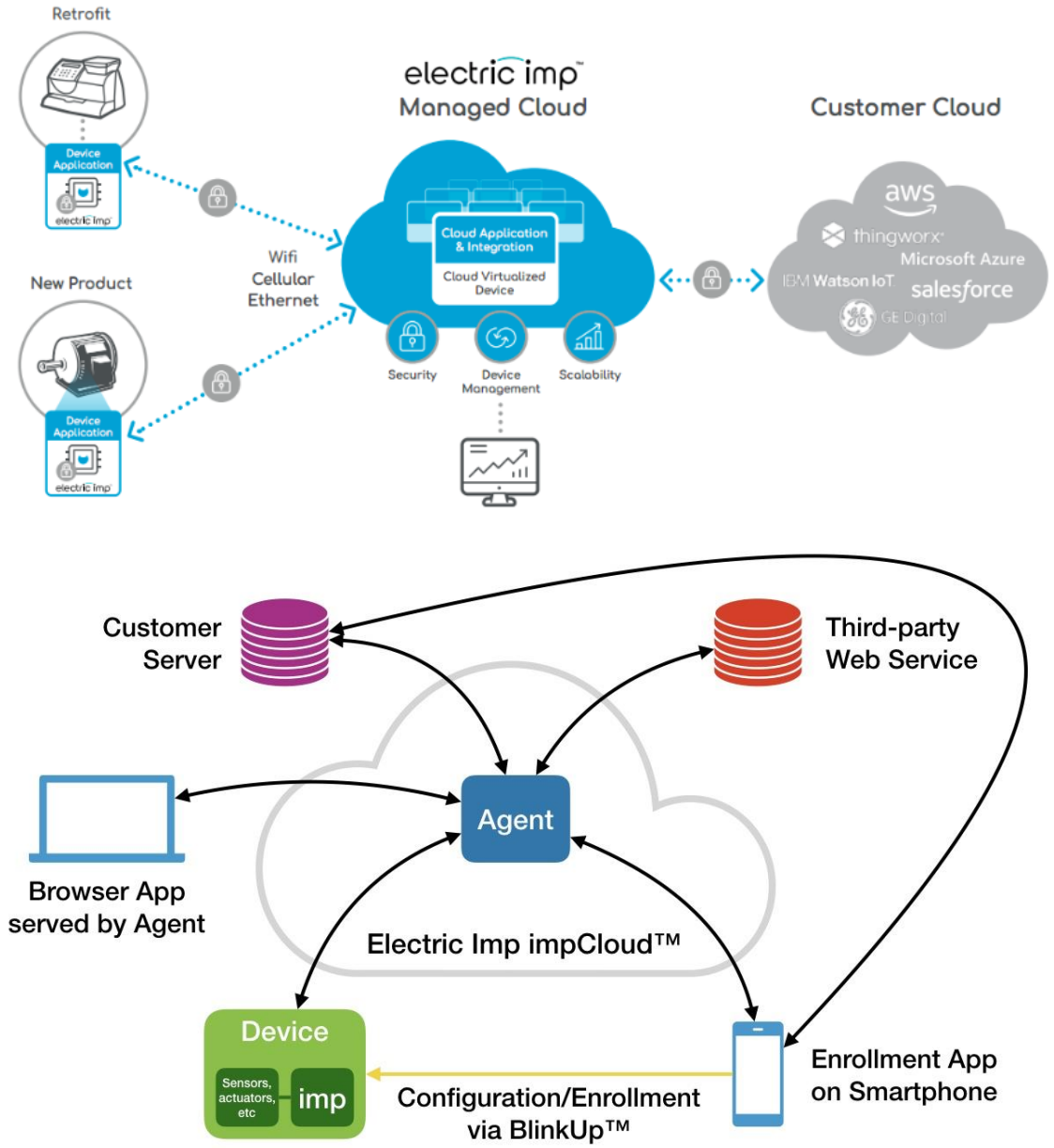


Figura 4: Electric Imp. (Fuente: https://www.electricimp.com/platform/how-it-works/).

2.1.3.2. Características

- Proporcionar herramientas de funcionalidad avanzada, informes y comandos para gestionar los dispositivos.
- Usado para desarrollo de prototipos debido a su gran variedad de dispositivos.
- Supervisar el estado de todos los dispositivos conectados a la red.
- Comprobar el estado de conectividad y el uso del ancho de banda de los dispositivos.

- Gestión de los dispositivos desde la nube.

2.1.4. Spark Works

2.1.4.1. Definición

Spark Works es una plataforma software que recoge los eventos generados por los dispositivos IoT conectados a la red, asimismo, la plataforma Spark Works envía comandos a los dispositivos IoT (SensorFlare, 2019).

Esta plataforma también se conecta con la nube con el fin de realizar análisis y procesamiento de datos (cloud computing). En la siguiente figura se puede observar el flujo de datos.

Además, ofrece la plataforma gratuita SensorFlare.

SensorFlare permite implementar fácil y rápidamente la aplicación, ofreciendo una alta escalabilidad en términos de usuarios, en número de dispositivos conectados y el volumen de datos procesados. Además, proporciona una elegante interfaz web, aplicaciones móviles para Android/ iOS y plantillas de código para integrar los dispositivos IoT con la plataforma IoT Spark Works. Ayuda a controlar todos los dispositivos inteligentes, incluso si son producidos por diferentes fabricantes (SensorFlare, 2019).

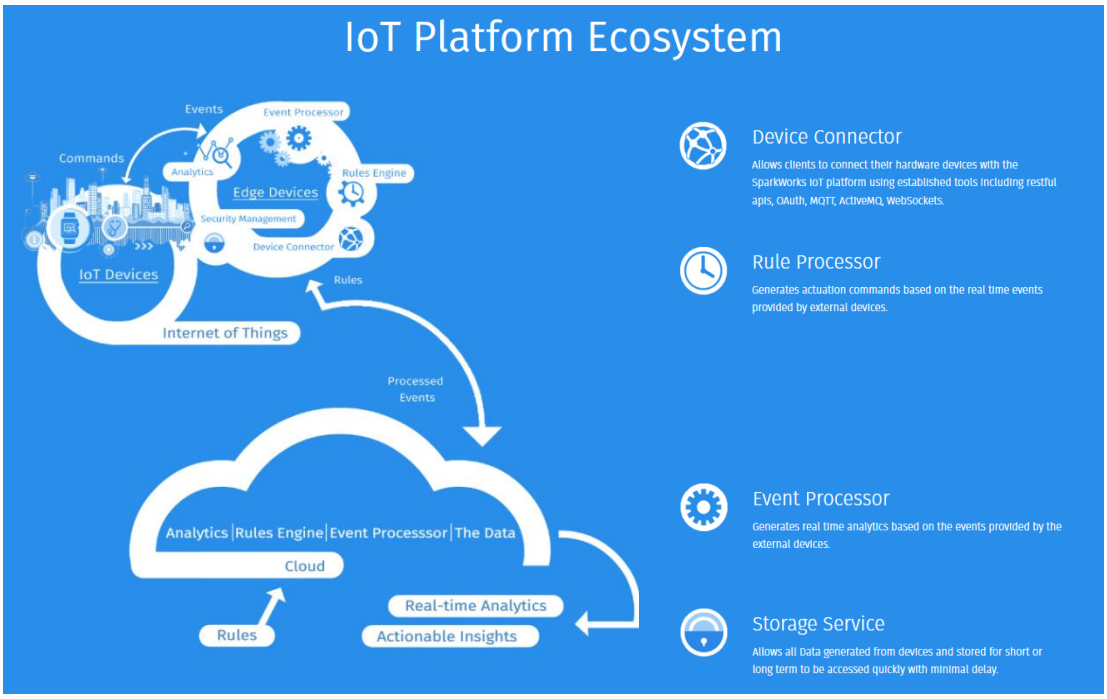


Figura 5: Spark Works. (Fuente: https://www.electricimp.com/platform/how-it-works/).

2.1.4.2. Características

- Rápido acceso a los datos de los dispositivos IoT, valores de los sensores y analítica de datos.
- Métodos sencillos para combinar vistas de dispositivos, etiquetar y filtrar dispositivos y acceder a datos históricos.
- Acceso abierto desde cualquier lugar a través a un gran conjunto de proyectos de código abierto.
- Ofrece plantillas de código para conectar los dispositivos IoT y configurar tareas de análisis de datos.
- Fácil monitorización de datos de los sensores conectados a la red.

2.1.5. Blaulabs

2.1.5.1. Definición

Blaulabs es una plataforma de IoT basada en SaaS (software as a Service). Permite recopilar y analizar datos de consumo de agua, energía y gas que son recogidas por equipos e instrumentación asociada (sensores de presión, temperatura, nivel, válvulas, bombas, hornos, etc…). Ofrece bases de datos, monitorización y análisis en tiempo real. Tiene como propósito acelerar el desarrollo de proyectos relacionados con Smart Manufacturing, Smart Infrastructure, Smart Energy, Smart Grid y Smart Cities.

2.1.5.2. Características

- Proporciona una plataforma que ofrece herramientas de análisis de datos de los sensores conectados a la plataforma.
- Adquiere, almacena y procesa cualquier tipo de dato a partir de cualquiera de los dispositivos conectados.
- Ofrece módulos listos para usarse para crear soluciones IoT para ciudades inteligentes (Smart City).
- Proporciona eficiencia e innovación a las infraestructuras y soluciones inteligentes listas para usarse para el control de la energía.

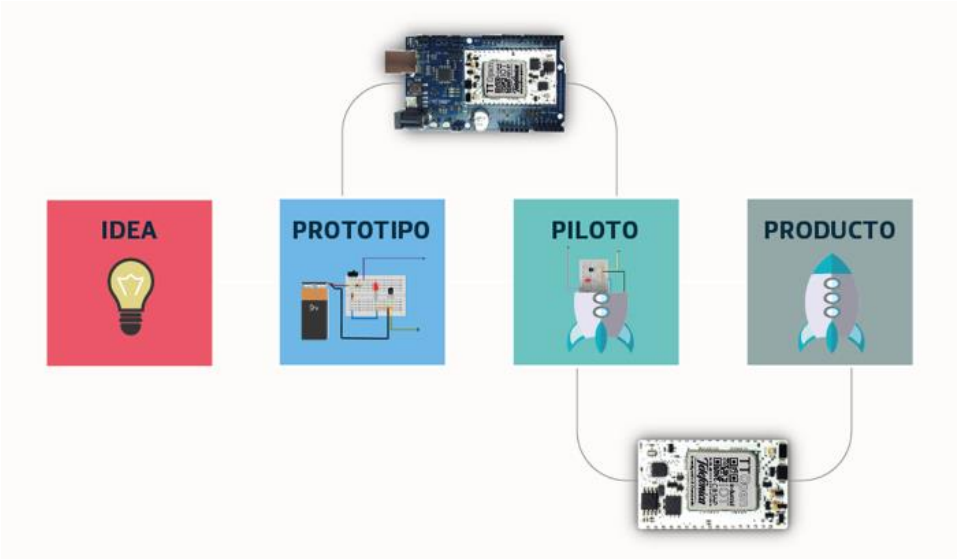
- Ayuda a mejorar la productividad de producción y a reducir los niveles de emisión de CO2.
- Permite reducir el consumo de energía y controlar los costes energéticos.
- Mejora la fiabilidad, la eficiencia y ahorra costes de las redes inteligentes (Smart Grids).

2.1.6. Thinking things

2.1.6.1. Definición

Thinking Things es una solución modular de punto a punto que combina módulos Plug and Play para la construcción de dispositivos inteligentes y conectados de forma personalizada en el área de IoT. Facilita la integración de Thinking Things en diferentes plataformas. Además, permite la construcción de soluciones a medida por los desarrolladores (Telefonica, 2019).

Hay distintos kits (estándar) preparados para utilizarse. El kit ambiente que permite el seguimiento y la supervisión de un dispositivo al incluir un GPS y módulo de comunicación. El kit de presencia ofrece la monitorización sencilla de estancias ofreciendo una solución Plug and Play. El kit jardín proporciona el control de dispositivos inteligentes al incluir diferentes sensores ambientales y un módulo de comunicación.



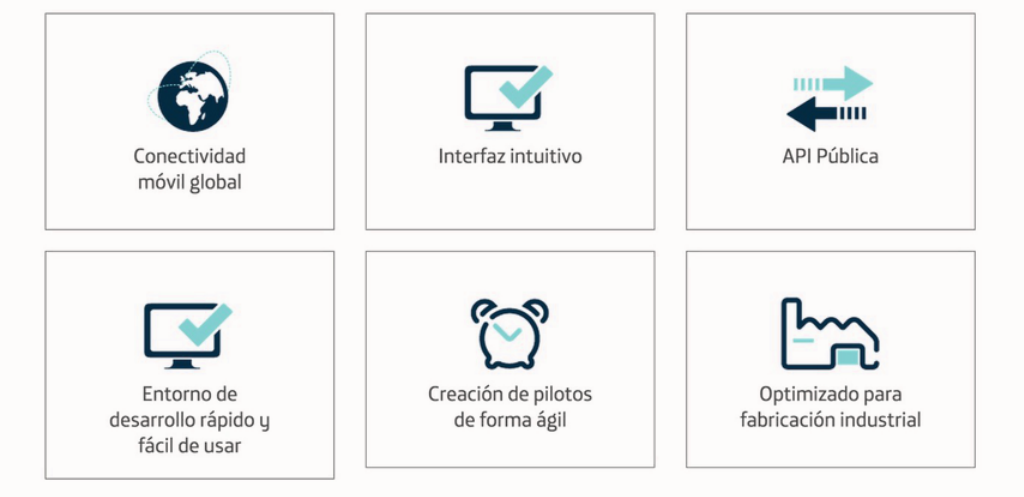


Figura 6: Thinking Things. (Fuente: https://blogthinkbig.com/thinking-things-el-iot-modular-mas-intuitivo).

2.1.6.2. Características

- Creación personalizada de dispositivos conectados mediante la combinación de los módulos Plug and Play sin necesidad de infraestructura.
- Conexión desde cualquier lugar del mundo con conectividad móvil.
- Control y supervisión remota sobre la plataforma.
- Programación de secuencias automáticas mediante interfaces sencillas de usar por web y móvil.
- Alertas vía SMS, correo electrónico, etc., fácilmente programables.
- Fácil integración entre la plataforma y dispositivos Hardware existentes, ofreciendo así soluciones a medida.

2.1.7. Zatar

2.1.7.1. Definición

Zatar es una plataforma software que combina la gestión de la configuración, el transporte y el almacenamiento de datos. Esto permite crear soluciones eficaces de manera muy rápida que pueden ser controladas con un amplio conjunto de herramientas. Además, ayuda a mejorar la eficiencia y optimizar las experiencias de los usuarios (Zatar IoT Platform, 2014).

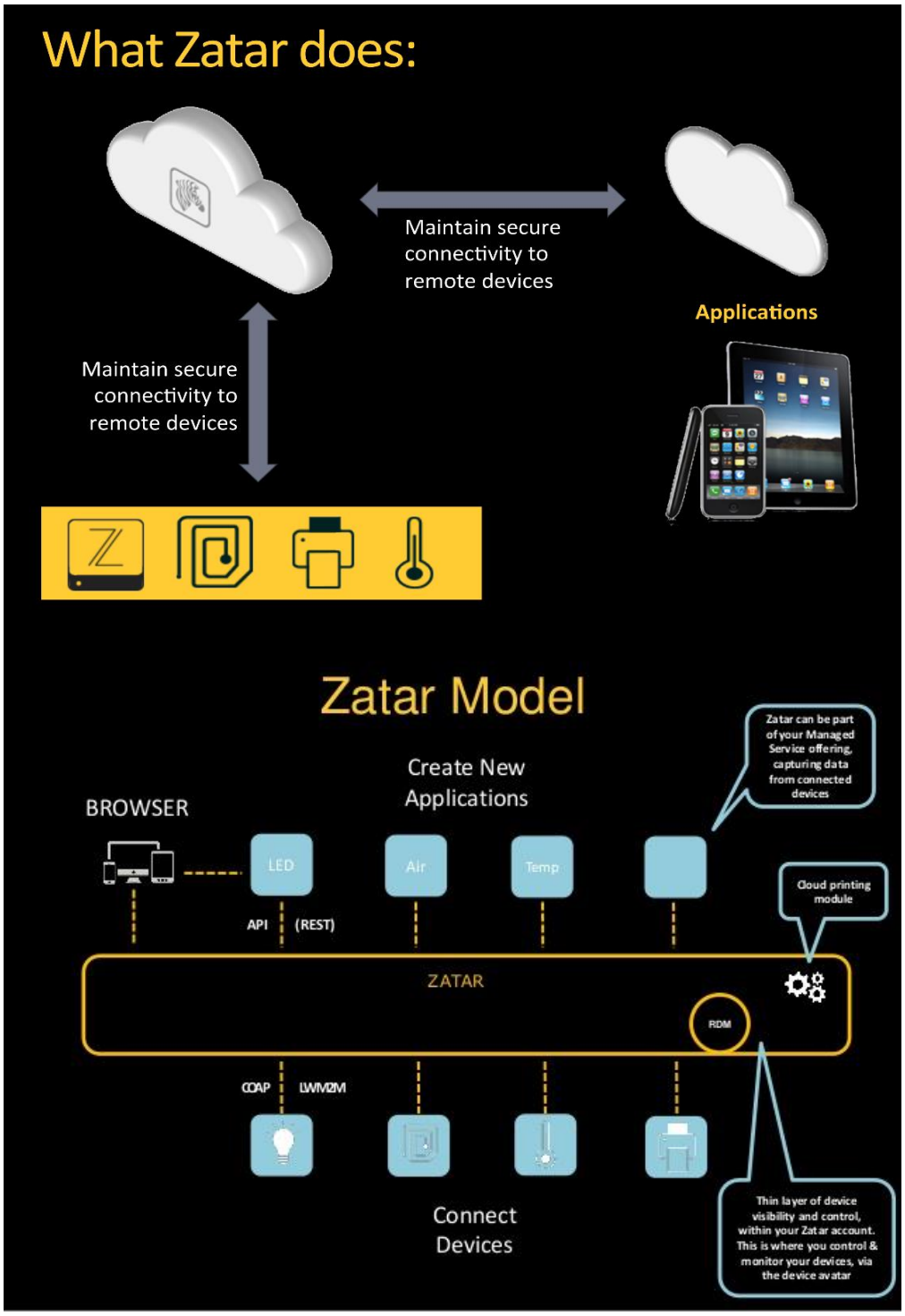


Figura 7: Zatar. (Fuente: http://me.westcon.com/content/support/app-zone/zatar).

2.1.7.2. Características

- Desarrollo de aplicaciones que proporcionen información en tiempo real sobre las aplicaciones.

- Creación de experiencias de usuario sofisticadas con la recopilación de forma segura de los datos a partir de los sensores inteligentes.
- Vigilancia y supervisión del rendimiento de la aplicación de forma remota.
- Control de la actividad de los dispositivos y de los datos analíticos proporcionados combinando los datos en tiempo real de los dispositivos con datos históricos.
- Diseñar dispositivos de IO más inteligentes para que proporcionen datos útiles.
- Aprovechamiento de la potencia de los estándares (CoAP, LWM2M) para así no perder la comunicación si la energía disminuyese o el ancho de banda estuviera limitado.

2.1.8. Amazon Web Services IoT (AWS IoT)

2.1.8.1. Definición

AWS IoT ofrece una comunicación segura bidireccional entre los dispositivos conectados a internet, como sensores, actuadores, microcontroladores integrados o dispositivos inteligentes, y la nube de AWS. Esto le permite almacenar y analizar los datos adquiridos a través los distintos sensores. Además, puede crear aplicaciones para permitir a sus usuarios el control de los dispositivos desde sus teléfonos o tablets (AWS IoT, 2019).

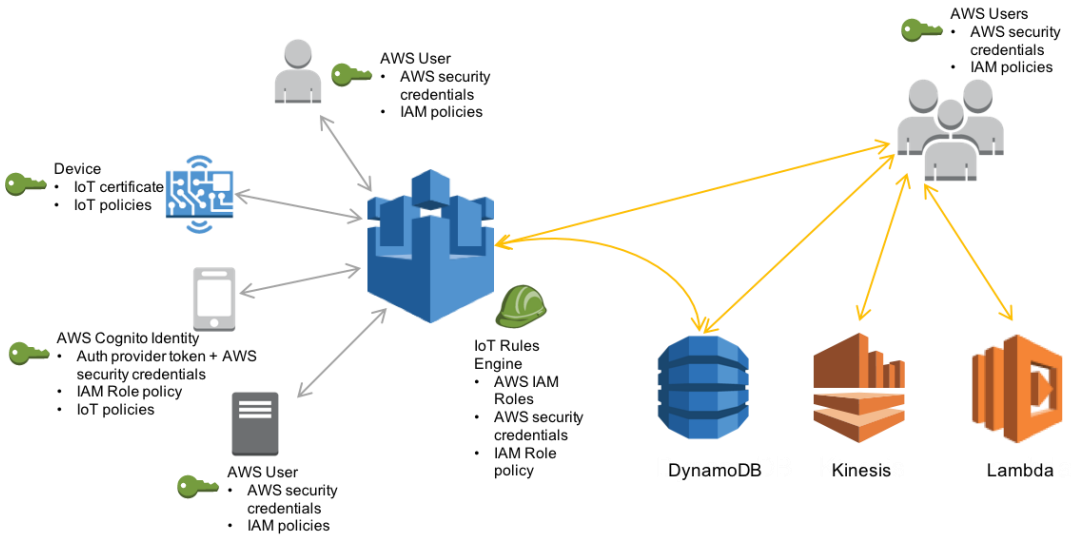


Figura 8: Amazon Web Service. (Fuente: https://docs.aws.amazon.com/es_es/iot/latest/developerguide/iot-security-identity.html).

2.1.8.2. Características

- El gateway para dispositivos permite una comunicación segura y eficaz con AWS IoT.
- Mecanismo seguro para que los dispositivos y las aplicaciones de AWS IoT publiquen y reciban mensajes entre sí.
- Proporciona funciones de procesamiento de mensajes y de integración con otros servicios de AWS.
- Ofrece un servicio de seguridad e identidad para proteger las credenciales de los datos.
- Organización de recursos asociados a cada dispositivo en la nube de AWS.
- Posibilidad de asociar certificados e ID de cliente MQTT a cada dispositivo para responder eficazmente a los problemas que se presenten.
- Organización de dispositivos en grupos creando una jerarquía donde todos los cambios se aplicarán a todos.
- El servicio de Jobs permite describir un conjunto de operaciones remotas que se ejecutan o envían a los dispositivos conectados a AWS IoT.

2.1.9. Azure IoT Hub

2.1.9.1. Definición

Microsoft Azure es una plataforma software en expansión de servicios en la nube. Ayuda a cubrir las necesidades comerciales. Proporciona la capacidad de crear, desarrollar, administrar e implementar aplicaciones en una red global con diferentes herramientas (Microsoft Azure, 2019).

Las aplicaciones de Azure son fácilmente escalables al aumentar la demanda de los usuarios. También, ofrece la posibilidad de diseñar aplicaciones de alta disponibilidad incluyendo la conmutación por error. La administración de los servicios puede hacerse a través de Azure Portal o mediante programación a partir de las API y las plantillas específicas del servicio.

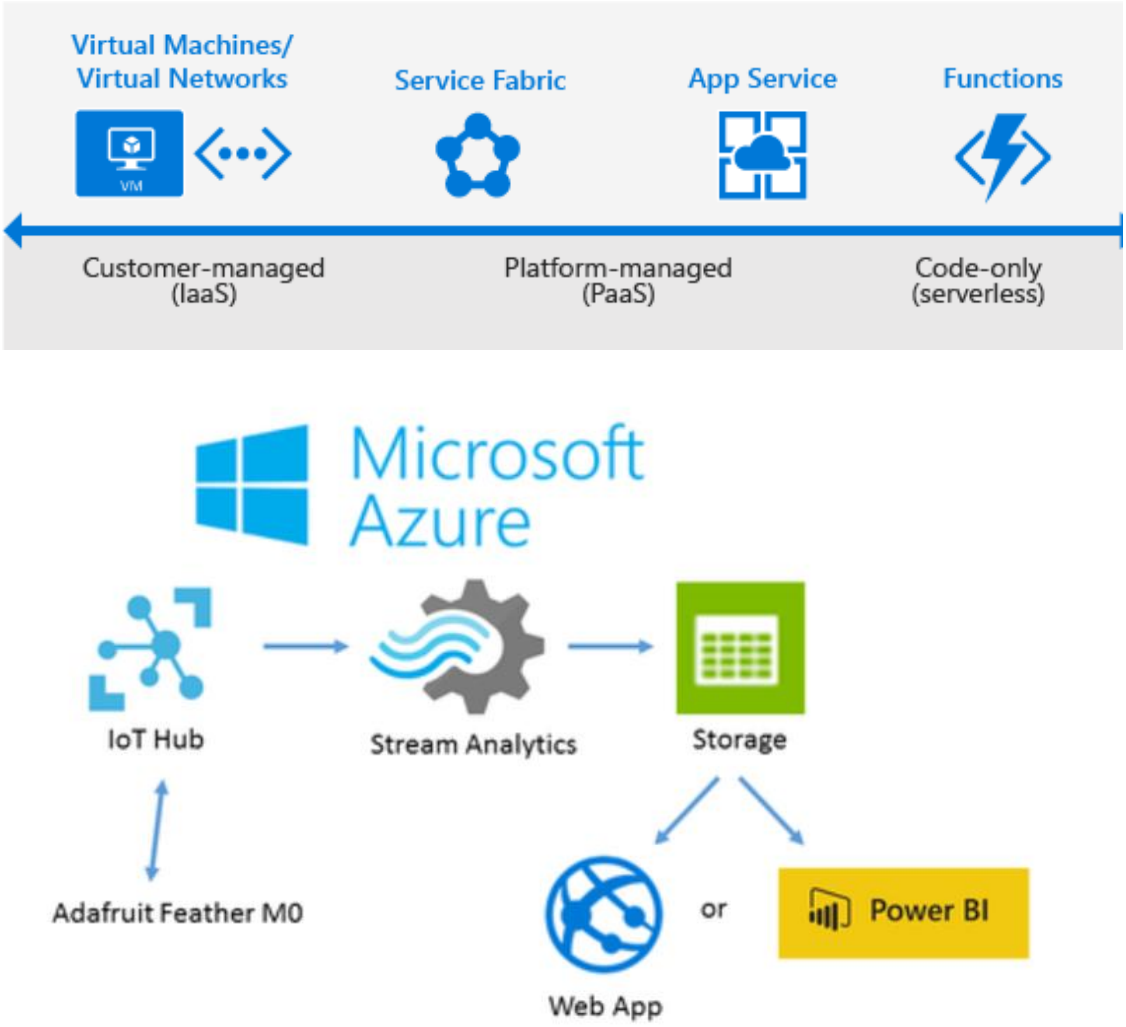


Figura 9: Microsoft Azure. (Fuente: https://azure.microsoft.com/es-es/services/iot-hub/).

2.1.9.2. Características

- Permite reducir los ciclos de comercialización al entregar características con mayor rapidez.
- La nube proporciona la capacidad de desarrollar e implementar soluciones desde cualquier lugar.
- Los servicios de inteligencia artificial y datos solidos permiten la creación de aplicaciones inteligentes, permitiendo el desarrollo de aplicaciones web, back-ends de aplicaciones móviles y aplicaciones de API.
- Provee requisitos de seguridad y privacidad.
- Proporciona la infraestructura como servicio IaaS (Infraestructure as a Service) para ofrecer al usuario un control total sobre el hospedaje de las

aplicaciones, implementando la aplicación a máquinas virtuales Windows o Linux.

- Ofrece una amplia variedad de servicios para obtener un desarrollo completamente personalizado.

2.1.10. Oracle Internet of Things

2.1.10.1. Definición

Oracle Internet of Things es una plataforma que innova, moderniza y compite en el mundo virtual. La plataforma ofrece servicios integrados y completos en la nube que capacita a los usuarios a crear, organizar y gestionar un amplio volumen de datos sin dificultades, tanto en la nube como on-premises. Utiliza la infraestructura como servicio (IaaS) para mejorar la transformación digital y aumentar la participación de otros usuarios por medio de interacciones sociales y aplicaciones móviles (Oracle, 2019).

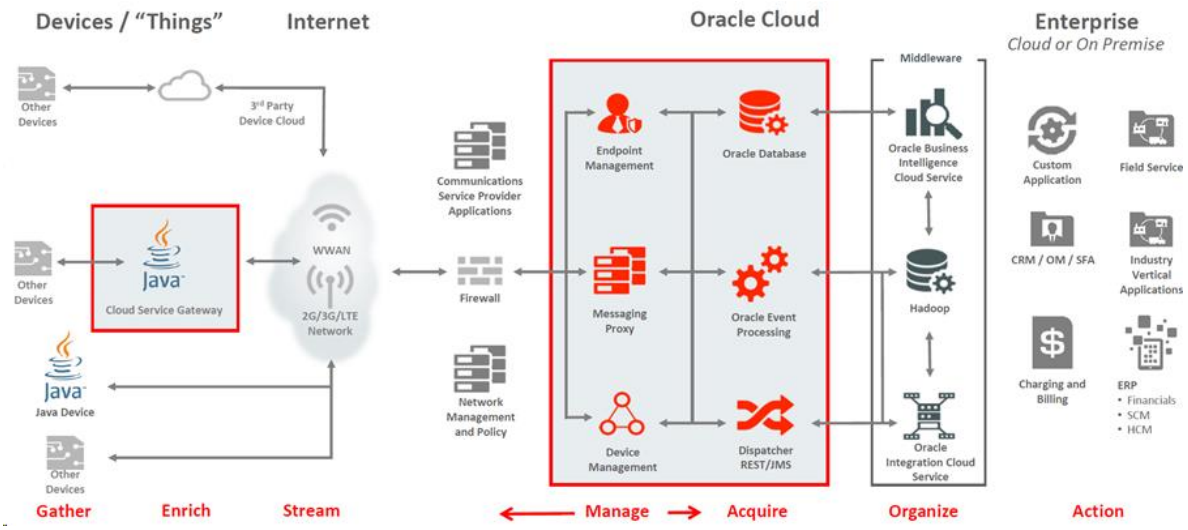


Figura 10: Oracle Internet of Things. (Fuente: https://www.oracle.com/es/solutions/internet-of-things/).

2.1.10.2. Características

- Estandarizar la integración de los dispositivos virtualmente, basado en API con aplicaciones y dispositivos IoT de Oracle y de terceros.
- Mensajería de alta velocidad permitiendo una comunicación bidireccional, segura y fiables entre la nube y los dispositivos.
- Ofrece la gestión de los puntos finales de los dispositivos, los datos y las identidades.
- Análisis de flujos de datos entrantes con agregación, filtrado y correlación de eventos en tiempo real.
- Análisis de big data para consultar y visualizar un gran volumen de datos con soporte integrado de la plataforma.
- Envío de datos y eventos de IoT a aplicaciones y flujos de procesos desde aplicaciones empresariales y móviles con o sin conectividad.
- Combina los servicios de la nube pública con los on-premises.
- Ofrece alto rendimiento para aplicaciones tradicionales y bases de datos de alta disponibilidad aprovechando el alto ancho de banda y la red de gran escala.

2.1.11. Watson IoT

2.1.11.1. Definición

Watson IoT es una plataforma que te permite conectar sensores y dispositivos a través de la nube. Proporciona la capacidad de crear y utilizar la plataforma de forma personalizada adecuándose a la necesidad del usuario, ofreciendo un recurso adaptable, escalable y abierto. La plataforma permite al usuario construir, lanzar y gestionar las aplicaciones IoT y obtener soluciones rápidas y seguras. Además, permite adquirir y analizar datos de forma segura en tiempo real. También se puede gestionar el ciclo de vida de los datos para optimizar la utilización del almacenamiento y reducir los costes, manteniendo al mismo tiempo la flexibilidad (Watson Internet of Things, 2019).

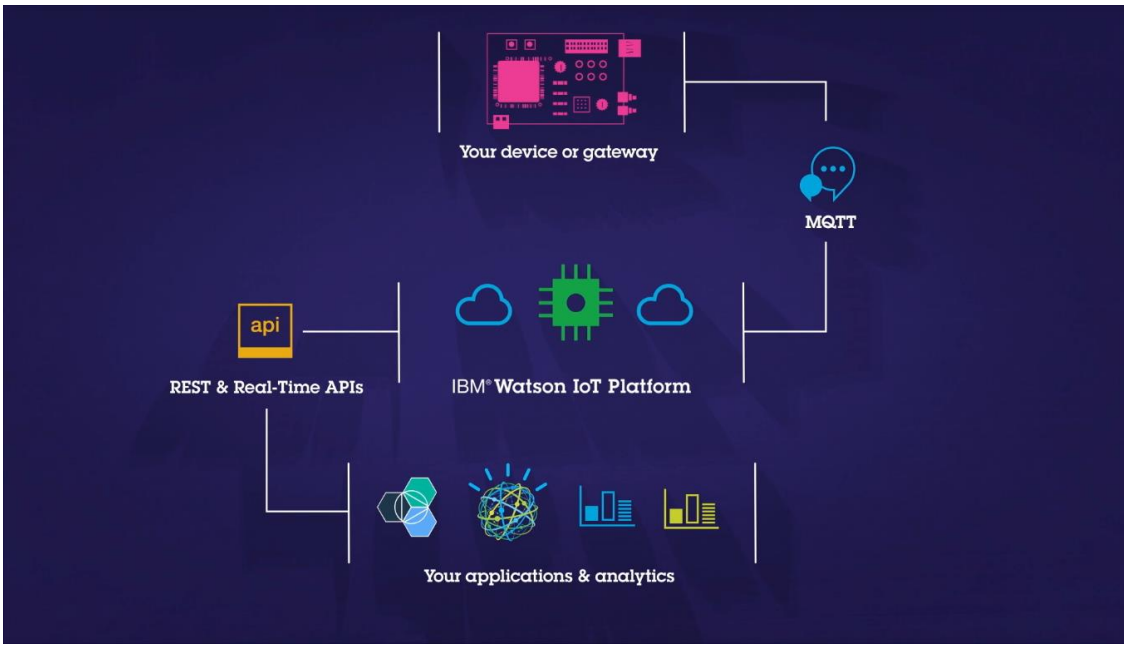


Figura 11: Watson IoT. (Fuente: https://developer.ibm.com).

2.1.11.2. Características

- Conexión, gestión y protección de los dispositivos de forma rápida y segura.
- Escalar rápidamente debido al alcance global de la plataforma.
- Adquisición, procesamiento y almacenamiento de los datos procedentes de los sensores para transformarlos en datos valiosos.
- Fácil adquisición y visualización de la información a través de los análisis basados en inteligencia artificial.

- Reducir el coste operacional haciendo que los dispositivos funcionen de forma más eficiente, aumentando la productividad.
- Comunicación bidireccional que aumenta la innovación de los productos.

2.1.12. Xively – Google Cloud IoT

2.1.12.1. Definición

Xively es una plataforma IoT inteligente que ofrece un conjunto de servicios completamente administrado e integrado para conectar, administrar y almacenar datos a gran escala, así como de forma fácil y segura, mediante una red global de dispositivos. Por tanto, permite a los usuarios controlar sus sistemas desde cualquier lugar y en cualquier momento. Además, permite procesar, analizar y visualizar datos en tiempo real e implementar cambios operacionales, posibilitando crear metodologías más ágiles y rentables. Actualmente, Xively forma parte de la plataforma de nube de Google (Xively, 2019; Google Cloud, 2019).

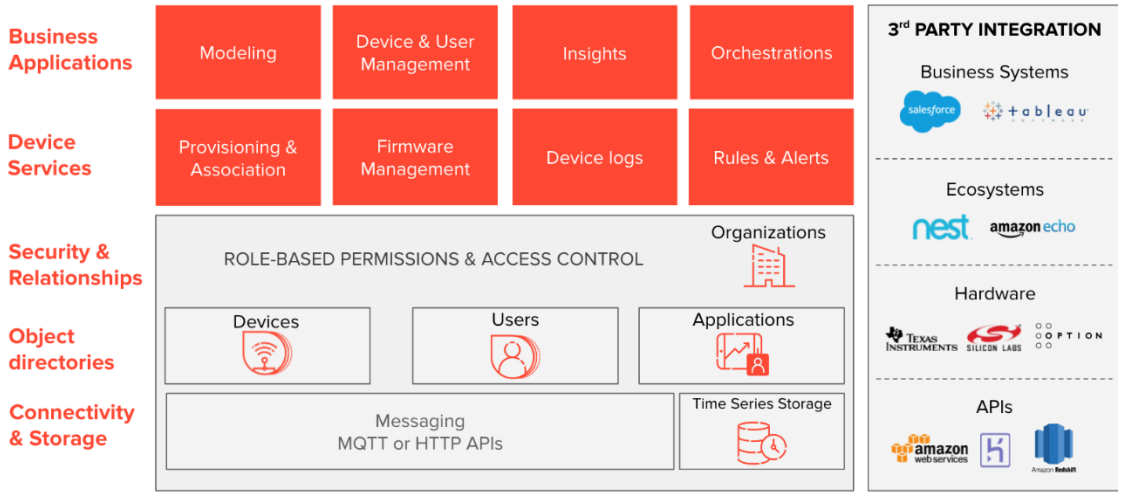


Figura 12: Xively IoT. (Fuente: https://xively.com/).

2.1.12.2. Características

- Soporte de los dispositivos y control de los datos que estos producen en tiempo real.
- Conexión rápida y segura de dispositivos.
- Seguridad máxima a través de certificados firmados por una autoridad certificadora y con la autentificación con claves asimétricas.
- Mejorar la eficacia operativa de los dispositivos con actualizaciones del firmware.
- Plataforma sin servidores evitando costes de mantenimiento de infraestructura.
- Servicio compatible con los dispositivos de distintos fabricantes de hardware.

2.1.13. Samsung Artik

2.1.13.1. Definición

Samsung ARTIK es una plataforma de punto a punto que ofrece que unifica hardware, software, nube, seguridad y un ecosistema de socios. Esto permite que la plataforma esté diseñada para ofrecer rapidez, además de tener seguridad en los dispositivos y protección de los usuarios. Los módulos ARTIK incluyen las redes necesarias para su hardware de red integrado, por lo que permite reducir costes y tiempo (Samsung ARTIK, 2016).

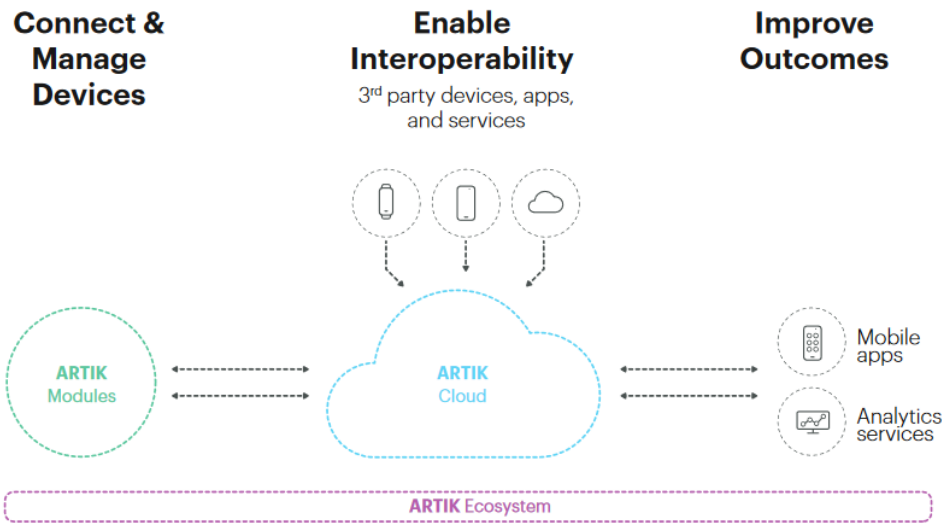


Figura 13: Samsung Artik. (Fuente: https://www.artik.io/blog/).

2.1.13.2. Características

- Rápido despliegue de los dispositivos y gateways utilizando módulos plug&play.
- Soporte de gran número de protocolos de conectividad.
- Conexión eficaz y bidireccional entre los dispositivos y la nube.
- Intercambio seguro de mensajes entre los dispositivos y la nube incluso en dispositivos de bajo consumo como portátiles o sensores.
- Fácil comunicación de los dispositivos con otros servicios de la nube o dispositivos.
- Adquisición de datos en tiempo real
- Eficaz seguridad y privacidad de los dispositivos, aplicaciones e interacciones de los usuarios.
- Monitorización y control del conjunto de dispositivos y ejecución de forma remota.
- Aumento de la productividad de desarrollo con un modo sencillo de usar, APIs abiertas, SDKs y herramientas.
- Escalar o reducir sin pérdidas en periodos de inactividad.

2.1.14. Adafruit IO

2.1.14.1. Definición

Adafruit IO es una plataforma que facilita el análisis de los datos al centrarse en la sencillez de uso y en la conexión de los distintos dispositivos con una programación muy sencilla. Se trata de un servicio en la nube que evita que el usuario tenga que gestionarlo, permitiendo al usuario conectarse a él mediante internet. La plataforma está pensada para almacenar y recuperar datos principalmente. Adafruit IO está construido en base a Ruby on Rails y Node.js (Adafruit, 2019).

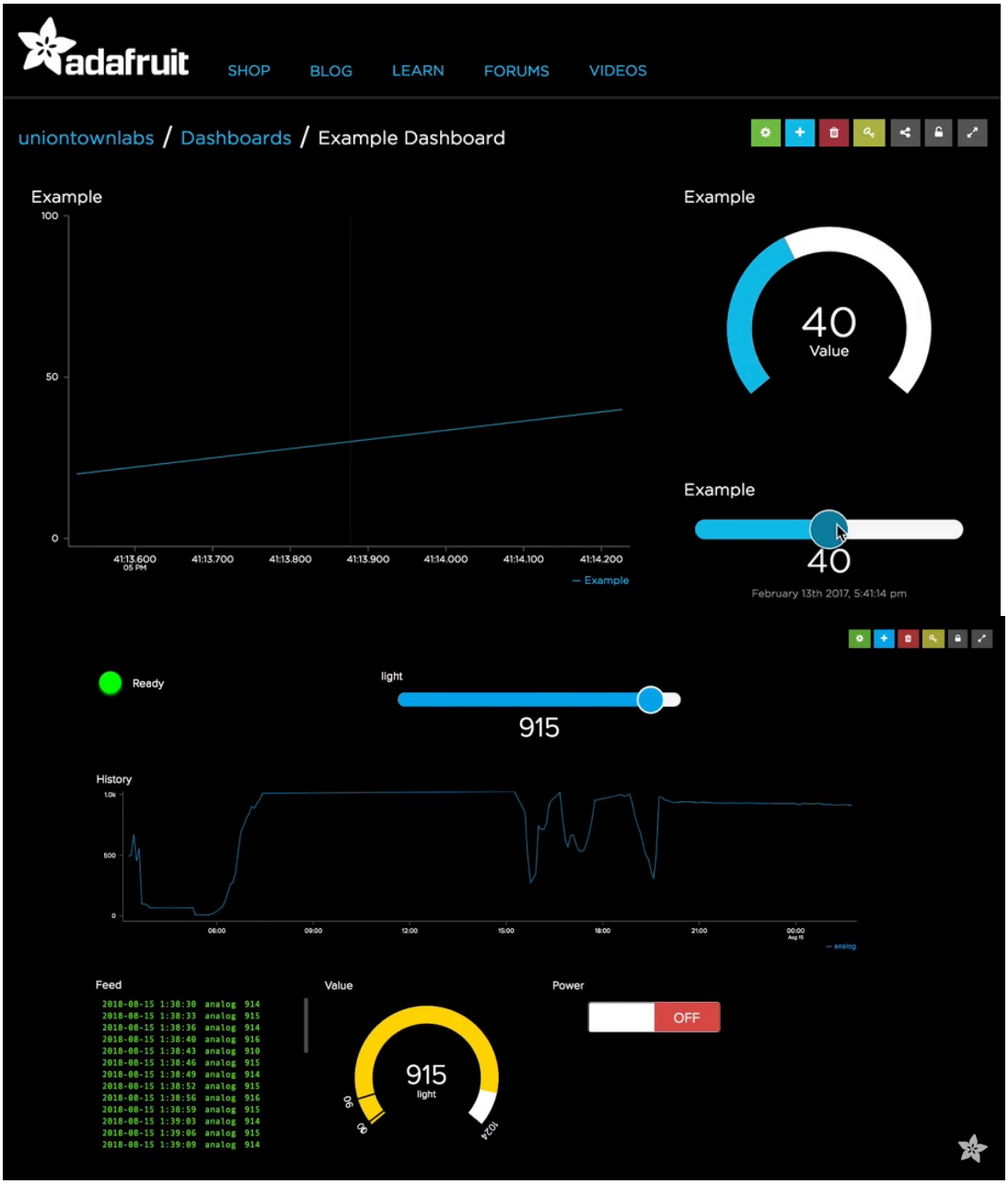


Figura 14: Ada Fruit IO: (Fuente: https://io.adafruit.com/)

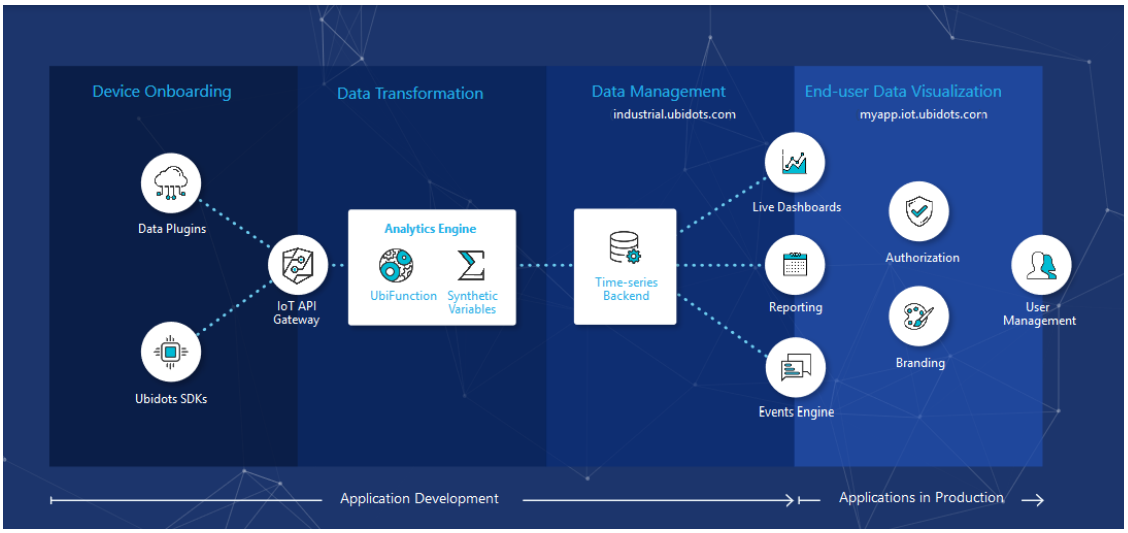
2.1.14.2. Características

- Muestra los datos en tiempo real.
- Conexión del proyecto a la web para un fácil control sobre los actuadores y lectura de datos de los sensores.
- Conexión de los proyectos a servicios web como Twitter, RSS feeds, servicios meteorológicos, etc., y a otros dispositivos con acceso a Internet.
- Servicio gratuito, lo que disminuye costes al usuario.
- Tratamiento y visualización de múltiples fuentes de datos.
- Posibilidad de programación de eventos que envían correos electrónicos o SMS.

2.1.15. Ubidots

2.1.15.1. Definición

Ubidots es una plataforma IoT que presenta una nube basada en dispositivos globales. Es una plataforma asequible, fiable y utilizable en un ecosistema de plataformas IoT. Ubidots está especializada en soluciones de hardware y software conectadas para monitorizar, controlar y automatizar procesos remotamente en el ámbito de la salud, energía, industria, fabricación, servicios públicos y transporte. Además, recientemente creó Ubidots for Education que es una plataforma para estudiantes que les permite construir, desarrollar, probar, aprender y explorar el futuro de las aplicaciones y soluciones conectadas a Internet (Ubidots, 2019).



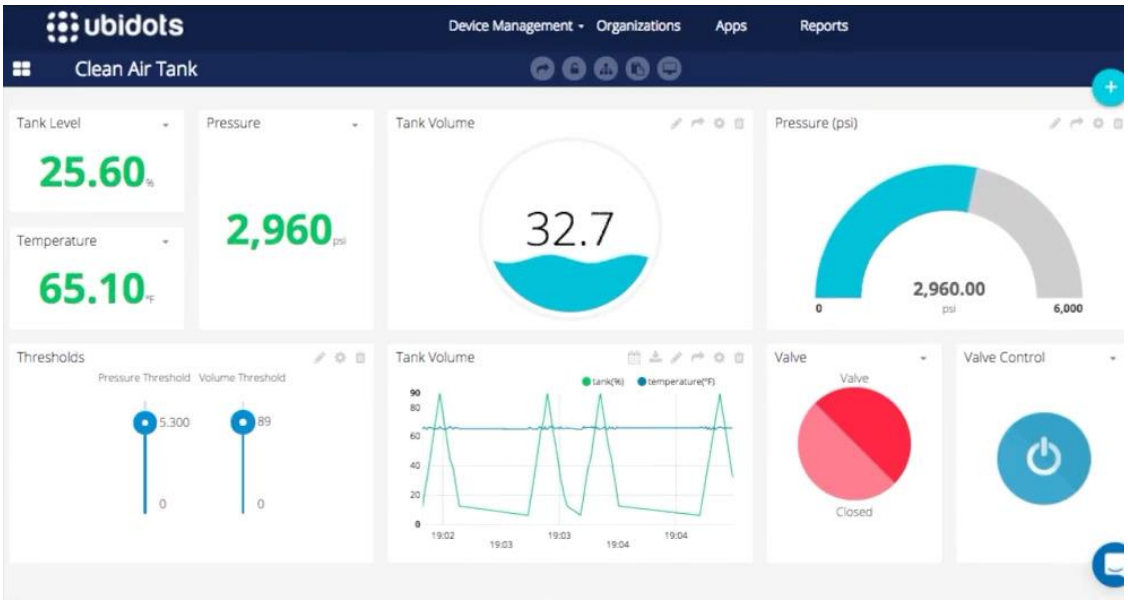


Figura 15: Ubidots. (Fuente: https://ubidots.com/).

2.1.15.2. Características

- Conexión del hardware a la nube con multitud de bibliotecas, SDKs y tutoriales.
- Configuración automática de variables, propiedades y de apariencia de los dispositivos para replicar el proceso en nuevos dispositivos.
- Personalización de la API.
- Mejora de la supervisión y el análisis de datos de las aplicaciones con integraciones API.
- Transformación de datos nativos en información mediante variables sintéticas.
- Creación de cuadros de mando en tiempo real para el análisis de datos y control de los dispositivos.
- Facilita compartir datos con enlaces públicos o integrando cuadros de mando o widgets en aplicaciones web privadas y móviles.
- Los comandos "kill switch" o "restart" pueden activarse cuando el hardware ha estado inactivo durante demasiado tiempo.
- Asignación de permisos y restricciones a cualquier usuario que interactúe con cuadros de mando, dispositivos y/o eventos.

2.1.16. My Devices Cayenne

2.1.16.1. Definición

Cayenne es una plataforma que destaca por ser el primer constructor de proyectos de IO de arrastrar y soltar (drag and drop) que permite a los usuarios crear rápidamente prototipos y compartir sus proyectos de dispositivos conectados. Cayenne está diseñado para ayudar a los usuarios a crear prototipos basados en la IoT y llevarlos a la producción. Además, las aplicaciones móviles Cayenne permiten supervisar y controlar remotamente los proyectos de IO desde las aplicaciones de teléfonos inteligentes Android o iOS y navegadores populares. También, mediante widgets personalizables facilita visualizar datos, establecer reglas, programar eventos con Cayenne Online Dashboard (Cayenne, 2019).

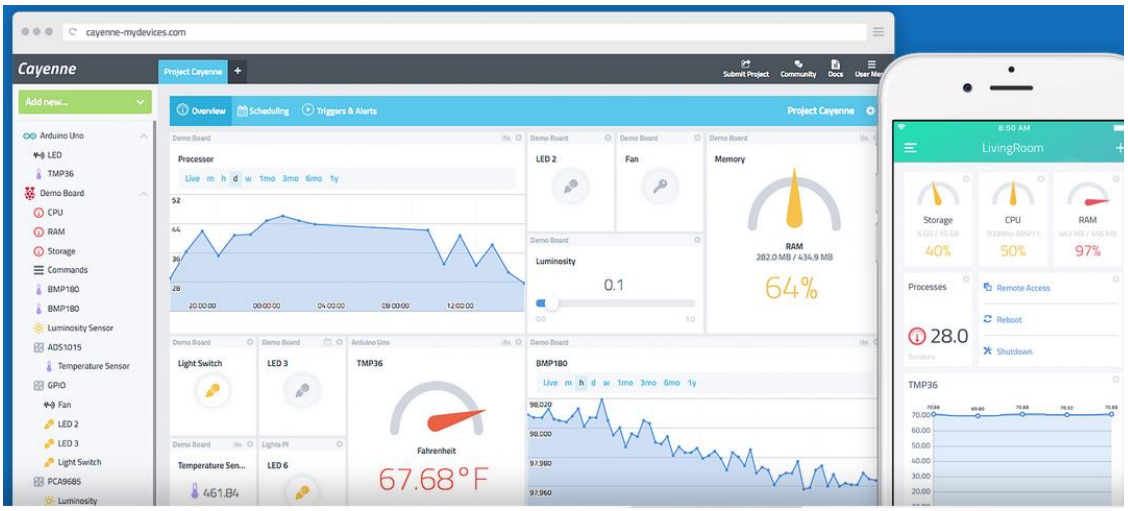


Figura 16: MyDevices Cayenne. (Fuente: https://mydevices.com/cayenne/docs_stage/intro/).

2.1.16.2. Características

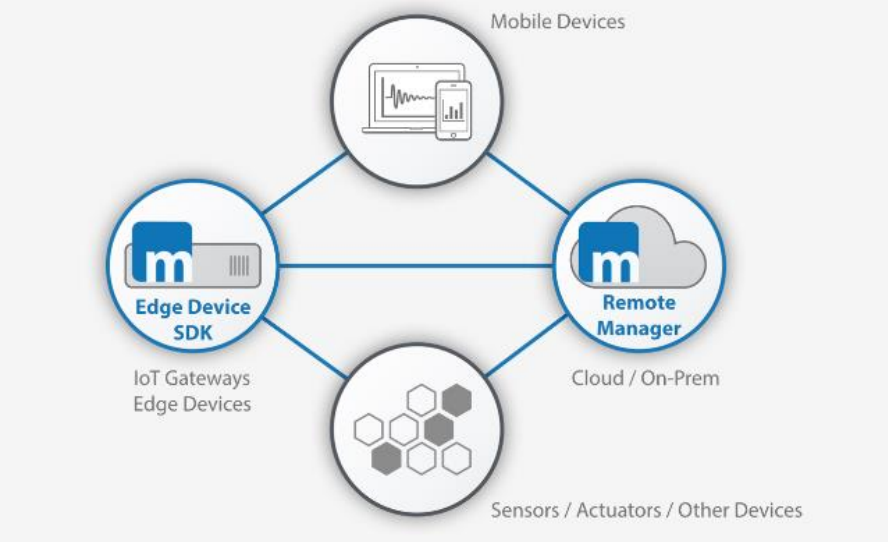
- Ofrece un cuadro de mandos personalizable para customizar con widgets drag and drop, permitiendo un control total sobre el proyecto.
- Visualización de datos, estado y acciones de los dispositivos mediante los widgets.
- Habilitar los mensajes de texto SMS y las notificaciones de correo electrónico en función de los eventos desencadenados.
- Permite controlar remotamente y crear nuevos proyectos a partir de las aplicaciones móviles.

- Automatización en unos pocos y sencillos pasos con potentes sentencias If/Then basadas en datos y acciones en tiempo real.
- Programar eventos para ordenadores, microcontroladores, sensores y actuadores conectados.
- Acceso y visualización en tiempo real y datos históricos de dispositivos y sensores.
- Seguimiento basado en la localización de cualquier dispositivo conectado, para obtener la ubicación del dispositivo, el estado y el historial de ubicación.
- Fácil conexión de dispositivos equipados con LoRa® para recopilar, transferir y mostrar datos inteligentes procesables.

2.1.17. Macchina IO

2.1.17.1. Definición

Macchina.io es una plataforma software segura, fiable y probada en la industria. La plataforma se basa en un potente servidor de aplicaciones web que soporta JavaScript además de C++, lo que permite el acceso al desarrollo de software a un gran número de usuarios. Además, permite programar la aplicación de su dispositivo de forma independiente del hardware, lo que facilita el cambio de plataforma de hardware o el soporte de múltiples plataformas de hardware diferentes con un único script. Por tanto, es ideal para aplicaciones de computación de alto rendimiento, así como para dispositivos con recursos limitados en términos de CPU y memoria. El diseño extremadamente modular y la escalabilidad flexible hacen de Macchina.io la solución perfecta para una amplia gama de dispositivos y aplicaciones (Macchina, 2019).



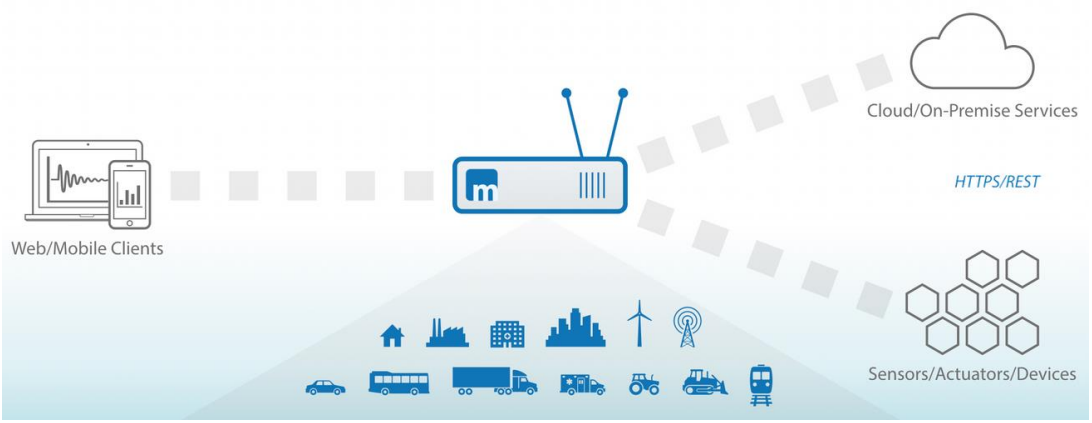


Figura 17: macchina IO. (Fuente: https://macchina.io/sdk.html).

2.1.17.2. Características

- Ofrece multitud de APIs para acceder a varios sensores y dispositivos, que pueden ser usados tanto desde JavaScript como desde C++ nativo.
- Proporciona un sistema modular flexible gracias a un potente servidor de aplicaciones web.
- Optimiza el rendimiento creando aplicaciones de forma más rápida al incluir el motor JavaScript V8.
- Ofrece una arquitectura potente de componentes y servicios obteniendo aplicaciones modulares fácilmente extensibles de forma segura.
- Mejora de la integración de redes de sensores, dispositivos de automatización y servicios en la nube debido a la amplia compatibilidad con protocolos de comunicación.
- Implementación en C++ de la plataforma IoT para una mejor eficiencia, bajo consumo y rendimiento.
- Macchina.io Remote Manager proporciona administración remota segura y acceso remoto vía Web, SSH y VNC.
- Macchina.io utiliza SQLite como base de datos que permite registrar los datos del sensor.

2.2. Comparativa de plataformas SW

Plataformas	SDK/ Lenguajes soportados	Protocolos soportados	Ventajas	Desventajas
ThinkSpeak	MATLAB	MQTT, HTTP	La plataforma es de código abierto. Ofrece gran número de integraciones de redes sociales. Creación de prototipos de sistemas IoT sin crear servidores o desarrollar softwares.	Es necesario utilizar un servicio de terceros para alertas que no sean de Twitter. Difícil configuración y sistema poco intuitivo. Documentación limitada a HW.
Altair SmartCore/ Carriots	Groovy	MQTT , HTTP	Ofrece la opción de generar las aplicaciones básicas mediante una herramienta visual con bloques. Utiliza un lenguaje funcional y moderno para el desarrollo de aplicaciones.	Su sistema de reglas y alertas es más simple que el ofrecido por otras soluciones del mercado. Es necesario exportar los datos a una base de datos externa.
Electric Imp	Squirrel	HTTP	Incorporación de HW, dispositivos y la nube. Muchos ámbitos de aplicación. Ecosistema propio. Escalable.	No es recomendable para iniciarse.
Spark	Scala, Phyton, Java	MQTT	Ideal para iniciarse. Escalable Comunidad al alza.	Compatibilidad HW.
Blaulabs	Python, Java	MQTT, HTTPs, OPCUA	Especialistas en Smart City con aplicaciones orientadas.	HW limitado.
Thinking things	Node.js	HTTP	Fácil configuración. Integración fácil entre HW-plataforma Ideal para prototipos. Ideal para iniciarse	Poca variedad sensores. Sólo es posible usar su HW.
Zatar	C, JavaScript, Java, Python, Android, iOS, C++	CoAP, HTTPs, LWM2M	Integración con impresoras. Escalable.	Software propietario.
Amazon Web Services IoT	C, JavaScript, Java, Python, Android, iOS, C++	MQTT, HTTP	Plataforma líder, con multitud de posibilidades por sus diferentes servicios.	Requiere mucha formación para conectar los diferentes servicios entre sí. La plataforma cambia constantemente, por lo que hay que estar actualizado.
Azure IoT Hub	C, Python, Node.js, Java, .NET	MQTT, AMQP, HTTP	Plataforma con multitud de servicios y una arquitectura muy bien definida por capas. Posee de un sistema de interacción con el dispositivo muy completo.	Los mensajes entre servicios no se incluyen dentro de la tarifa base. Los precios se engloban en 4 categorías poco flexibles.
Oracle Internet of Things Cloud Service	Android, C, iOS, Java SE, JavaScript	MQTT, HTTP	Facilita la conexión a los dispositivos con clientes y puertas de enlace que gestionan todo el proceso. Herramientas de análisis de datos muy completas.	Plataforma de reciente creación con un número limitado de servicios. Categorías de precios muy difíciles de entender.
Watson IoT	Node.js, Java, Python, C#, C, C++	HTTP, MQTT	Plataforma muy completa para analizar los datos para machine learning y minería de datos.	Carece de una solución interna para la representación de los datos. Los servicios disponibles son más reducidos.
Xively	iOS, Android	MQTT, HTTP, WebSocket	Muy sencilla de gestionar y utilizar. Gestión de dispositivos muy completa, mediante agrupaciones por localización o función.	Plataforma que solo provee de los servicios básicos para la recolección, análisis y representación de los datos.
Samsung Artik	C, C++, Node.js	HTTP, MQTT, WebSockets, CoAP	Facilidad de gestión del proceso con placas propias y sencillas de conectar. Integraciones con herramientas de terceros configuradas automáticamente.	Más dificultad para permitir la personalización. Plataforma más costosa y menos flexible.
Adafruit IO	Arduino, Ruby, Python, Node.js	MQTT, HTTP	Amplia sección de tutoriales. Provee de la mayoría de los controladores necesarios para los sensores. Los paneles que presentan la información gráficamente son muy fáciles de crear.	No es posible exportar la información fuera de la plataforma. No dispone de integraciones con terceros. No es posible configurar alertas o reglas para actuar según los valores recogidos por los sensores.
Ubidots	Python, Java, C, PHP, Node.js, Ruby	MQTT, HTTP	Dispone de numerosas librerías de placas de desarrollo específicas y de lenguajes de programación. Potente herramienta de representación gráfica de datos.	No permite la gestión de dispositivos por grupos. Carece de ofertas flexibles por uso.
My Devices Cayenne	Arduino, Raspberry Pi	MQTT	Plataforma capaz de gestionar directamente los conectores de las placas de desarrollo, evitando programar la interacción del sistema con los sensores. Documentación muy extensa y con muchos ejemplos.	Limitado soporte para placas de desarrollo y sensores, ya que tienen que ser compatibles con el software proporcionado por la plataforma. Uso excesivo de su aplicación móvil para la configuración y gestión de los dispositivos IoT.
Macchina IO	JavasScript	MQTT COAP ModBus	La plataforma es de código abierto. Dispone de un sistema de alertas con reglas.	Utiliza una base de datos SQLite muy limitada. La herramienta de representación gráfica es demasiado limitada y requiere la implementación de los componentes JavaScript por parte del usuario.

2.3. Dispositivos HW para IoT

2.3.1. Raspberry Pi

2.3.1.1. Descripción

Raspberry Pi fue lanzada en 2012. Los desarrolladores empezaron un proyecto para hacer que la computación sea divertida para los estudiantes creando interés en cómo funcionan las computadoras a un nivel básico. Posee un software de código abierto que ofrece a los usuarios la oportunidad de explorar el código subyacente. Por tanto, se considera una herramienta de aprendizaje ideal, ya que es barata de fabricar, fácil de reemplazar y sólo necesita un teclado y un televisor para funcionar (Raspberry, 2019).

Raspberry Pi es una computadora de bajo coste que se conecta a un monitor de ordenador o a un televisor y utiliza un teclado y un ratón estándar. Es un pequeño dispositivo destinado a cualquier usuario que busque explorar la informática y aprender a programar en lenguajes como Scratch y Python. Realiza multitud de actividades como un ordenador de sobremesa, desde navegar por Internet y reproducir vídeo de alta definición, hasta crear hojas de cálculo, procesar textos y jugar a juegos. Además, tiene la capacidad de interactuar con el mundo exterior, y ha sido utilizado en una amplia gama de proyectos de creadores digitales gracias a su bajo coste y a que sea de código abierto.

Sobre Raspberry Pi se pueden ejecutar diferentes sistemas operativos, como Android, Firefox OS, Raspbian, OpenWebOS o Unix. La mayoría de estos sistemas operativos están basados en el Kernel de Linux. También se pueden instalar sistemas operativos similares a Windows.

Este dispositivo también cuenta con infinidad de accesorios como videocámara, reloj, e innumerables sensores y actuadores.

Debido a su versatilidad, potencia y muy bajo coste, Raspberry Pi es una de las placas más utilizadas para el desarrollo/prototipado de proyectos basados en IoT.

Figura 18: Raspberry Pi Model 3

2.3.1.2. Características

- Bajo precio y gran versatilidad. Dispone de gran variedad de accesorios, lo que facilita el desarrollo de proyectos basados en IoT
- Soporta multitud de sistemas operativos, como Android, Firefox OS, Raspbian, OpenWebOS o Unix. Soporta lenguajes de programación como Python, BBC Basic, C y Perl.
- Dispone de conexión a internet vía WiFi y/o ethernet (dependiendo del modelo). Está diseñada para utilizar tarjetas SD para el arranque y el almacenamiento.
- Facilita el aprendizaje de la programación y la computación para todo el tipo de usuario.

2.3.2. Arduino

2.3.2.1. Descripción

Arduino es un dispositivo basado en el hardware y software de fácil uso. Las placas de Arduino son capaces de leer entradas, como luz en un sensor, un dedo en un botón o un mensaje de Twitter, y convertirlas en salidas, como la activación de un motor, encender un LED o publicar algo en internet. Además, puedes decirle a la placa qué hacer mandando un conjunto de instrucciones al microcontrolador de la placa. Para ello, se necesita usar el lenguaje de programación Arduino (basado en C++) y el software Arduino (Arduino, 2019).

Arduino fue creado como una herramienta fácil para realizar prototipos rápidamente dirigido a estudiantes sin conocimientos en electrónica y programación. Se ha ido adaptando para satisfacer nuevas necesidades y retos, diferenciando su oferta de simples tarjetas de 8 bits a productos para aplicaciones de IO, portátiles, impresión 3D y entornos embebidos. Todas las placas Arduino son de Hardware libre, permitiendo a los usuarios tener acceso a las especificaciones y diagramas electrónicos (permitiendo que cualquier persona/empresa pueda crear su propia placa basada en Arduino). El software también es un recurso abierto, es decir, que el código es accesible para todos los usuarios y además es modificable. El desarrollo software sobre la placa Arduino crece a partir de las contribuciones de todos los usuarios. Por tanto, el software Arduino es fácil de usar para principiantes, pero lo suficientemente flexible para usuarios avanzados, siendo compatible con Mac, Windows y Linux (Arduino, 2019).

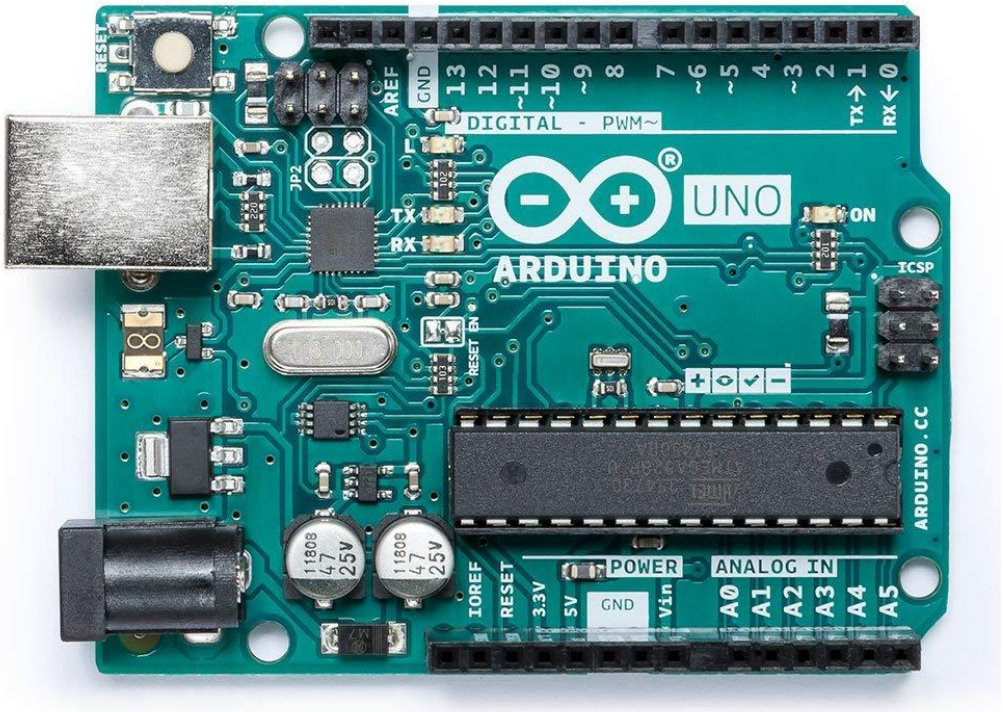


Figura 19: Arduino UNO R3

2.3.2.2. Características

- Se trata de un sistema económico, ya que las placas Arduino son relativamente económicas comparadas con otros microcontroladores.
- Ofrece una plataforma cruzada en la que el software de Arduino (IDE) se ejecuta en sistemas operativos Windows, Macintosh OSX y Linux.
- Proporciona un entorno de programación simple y claro, siendo un software de usar para principiantes, pero lo suficientemente flexible como para que los usuarios avanzados también lo aprovechen.
- Es un software de código abierto y extensible. El lenguaje puede ser expandido a través de las librerías C++. Además, permite aprender y aplicar el lenguaje de programación AVR C en el que se basa.
- Facilita un código abierto y hardware extensible, ya que los planos de las placas Arduino se publican bajo una licencia Creative Commons, por lo que permite personalizarlo.
- Dispone de una extensa comunidad de usuarios que sirve de fuente de información para el desarrollo de proyectos basados en Arduino.

2.3.3. Waspmote

2.3.3.1. Descripción

Waspmote está basado en una arquitectura modular. Waspmote es un dispositivo orientado a la creación de Redes Sensoriales Inalámbricas de bajo consumo. La arquitectura modular permite personalizar el hardware al integrar sólo aquellos módulos que se necesiten para cada dispositivo, pudiendo cambiar los módulos o expandirlos. Waspmote puede comunicarse con dispositivos externos a través de distintos puertos de entrada y salida (Gracia, 2012).

La plataforma fue creada por una multinacional española llamada Libelium para desarrollar tecnología que monitorizara de forma inalámbrica cualquier parámetro ambiental. La plataforma se compone principalmente de la placa Waspmote con un microcontrolador de la familia Atmel, memoria, batería, acelerómetro y sockets para ir añadiendo módulos. Usa el protocolo de comunicación Zigbee con gran alcance y dispone de diferentes módulos para dotarles de diferentes medios de comunicación, como, Bluetooh, GPS, y GPRS. La aplicación principal es para Smart City. La placa Waspmote usa el mismo compilador y librerías principales que Arduino, por lo que el mismo código es

prácticamente compatible en ambas plataformas solo hay que ajustar algunas líneas de código.

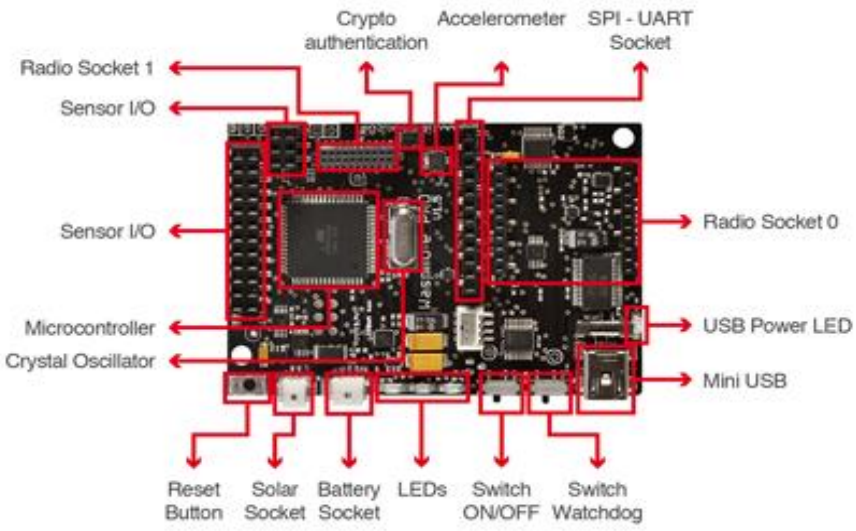


Figura 20: Waspmote. (Fuente: https://unpocodejava.com/2012/08/21/que-es-waspmote/).

2.3.3.2. Características

- Intercambio de información entre Waspmotes vía inalámbrica a través de distintos protocolos, agricultura y Smart cities.
- Tiene incorporado un reloj en tiempo real y un acelerómetro de tres ejes.
- Presenta modos de bajo consumo, pudiendo despertarse con una alarma programada o generando interrupciones.
- Ofrece gran cantidad de diferentes tipos de sensores de gases, líquidos y aguas “Smart water”,
- Proporciona una gran cantidad de información en código abierto para sus proyectos y ejemplos de aplicación.

2.3.4. Spark (Apache Spark)

2.3.4.1. Descripción

Apache Spark es un dispositivo Hardware con un motor de código abierto desarrollado específicamente para el procesamiento y el análisis de datos a gran escala. Spark ofrece la posibilidad de acceder a datos de multitud de fuentes,

incluyendo Hadoop Distributed File System (HDFS), OpenStack Swift, Amazon S3 y Cassandra. Apache Spark está diseñado para acelerar el análisis en Hadoop a la vez que proporciona un completo conjunto de herramientas complementarias que incluyen una biblioteca de aprendizaje con todas las funciones (MLlib), un motor de procesamiento de gráficos (GraphX) y procesamiento de secuencias (Apache Spark, 2019).

Apache Spark fue creado en 2009 en el AMPLab de UC Berkeley y fue donado a la Apache Software Foundation en 2013. Se ha convertido en el proyecto más activo en términos de contribuciones, ya que su popularidad destaca, tanto entre los desarrolladores como en las empresas, por su velocidad y eficiencia. Spark ejecuta programas que se encuentran guardados en la memoria hasta 100 veces más rápido que Hadoop MapReduce. Es una placa de un tamaño reducido que se compone de un módulo WiFi, cuya finalidad es la de dotar de conexión a Internet prácticamente a cualquier cosa. Spark está diseñado de forma nativa para funcionar en memoria, lo que le permite soportar el análisis iterativo y una compresión de datos más rápida y menos costosa (Apache Spark, 2019).

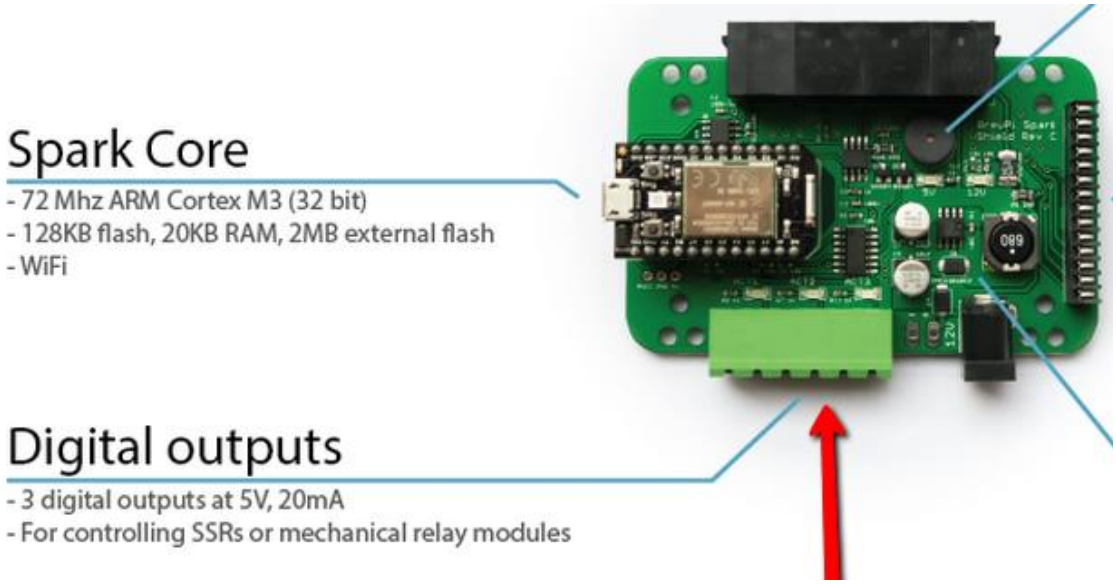


Figura 21: Apache Stark. (Fuente: https://www.webopedia.com/TERM/A/apache-spark.html).

2.3.4.2. Características

- Proporciona APIs de alto nivel en Java, Scala, Python y R, y un motor optimizado que soporta gráficos de ejecución general.
- Soporta un rico conjunto de herramientas de alto nivel, incluyendo Spark SQL, MLlib, GraphX y Spark Streaming.

- Es un dispositivo fácil de instalar y que se puede programar sin cables, gracias a su conexión WiFi, combinando la simplicidad de un Arduino con todo el poder del chip ARM cortex M3.
- La nube de la compañía Spark permite realizar desarrollos, actualizar y otras ventajas a nivel de software.
- Todos los diseños de firmware y hardware son de código abierto, lo que permite una integración libre en diversos proyectos.
- Si queremos ampliar la funcionalidad existe la posibilidad de ampliarlo con los SHIELD.

2.3.5. Intel Galileo

2.3.5.1. Descripción

Galileo es una placa de microcontrolador basada en el procesador de aplicaciones Intel® Quark SoC X1000, un sistema Intel Pentium de 32 bits en un chip. Es la primera placa basada en la arquitectura Intel® diseñada para ser compatible con los Shields Arduino diseñados para Arduino Uno R3. Los pines digitales 0 a 13 (y los pines adyacentes AREF y GND), las entradas analógicas 0 a 5, el cabezal de potencia, el cabezal ICSP y los pines del puerto UART (0 y 1), están todos en las mismas ubicaciones que en el Arduino Uno R3. Esto también se conoce como pinout de Arduino 1.0 (Arduino, 2019).

Intel Galileo es compatible con el entorno de desarrollo de software de Arduino (IDE), lo que hace que la usabilidad y la introducción sean muy sencillas. Además, la placa Galileo tiene varios puertos de E/S estándar de la industria de PC y características para expandir el uso y las capacidades nativas más allá del ecosistema de escudos de Arduino. Es multiplataforma por lo que se puede usar tanto en Linux, Mac como en Windows. Sin embargo, actualmente este producto se encuentra retirado.

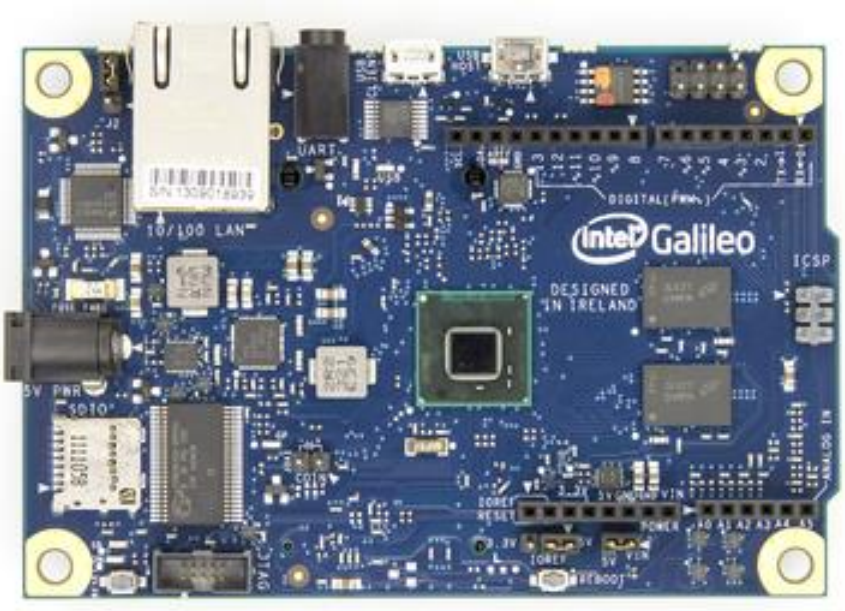


Figura 22: Intel Galileo. (Fuente: https://www.arduino.cc/en/ArduinoCertified/IntelGalileo).

2.3.5.2. Características

- Esta placa ejecuta un sistema operativo Linux libre con librerías de software de Arduino, lo que permite una mayor escalabilidad y reutilizar el software ya existente.
- Galileo tiene una serie de puertos y características que son estándares en la industria del PC.
- Ofrece un desarrollo software capaz de conectar cualquier dispositivo a internet gracias al uso de SDK, la potencia de proceso y su conectividad.
- Su sencillez facilita el aprendizaje a estudiantes o personas que se inician en los microcontroladores.

2.3.6. Zigbee

2.3.6.1. Descripción

ZigBee es un protocolo de comunicaciones inalámbricas. Zigbee es un protocolo global y abierto. Está diseñado con un conjunto de protocolos de alto nivel para utilizar señales de radio digitales de baja potencia para redes de área personal. ZigBee funciona con la especificación IEEE 802.15.4 y se utiliza para crear redes que requieren una velocidad de transferencia de datos baja, eficiencia energética y redes seguras. Se emplea en aplicaciones como dispositivos médicos, control de calefacción y refrigeración y sistemas de automatización de edificios, ya que está especialmente diseñado para domótica. ZigBee está diseñado para ser más simple y menos costoso que otras tecnologías de red personales como Bluetooth.

Hay tres tipos de dispositivos ZigBee según su papel en la red. Coordinador ZigBee (ZC) requiere memoria y capacidad de computo, ya que se encarga de controlar la red y los caminos seguidos para conectarse entre ellos. Router ZigBee(ZR) interconecta dispositivos separados en topología de red. Dispositivo final ZigBee (ZED) tiene la funcionalidad necesaria para comunicarse con su nodo padre (ZC o ZR), pero no puede transmitir información destinada a otros dispositivos.

Zigbee soporta tres tipos de topologías de red:

1) Star o Estrella: presenta larga vida útil como consecuencia del bajo consumo que requiere.

2) Mesh o Malla: en la cual existen múltiples rutas para alcanzar un destino, obteniéndose alta confiabilidad.

3) Cluster Tree o Racimo de Árbol: es una topología del tipo Mesh-Star que encierra los beneficios de ambas.

El departamento de ingeniería de Electrocomponentes S.A. ha desarrollado “plataformas” hardware como apoyo al diseñador de aplicaciones ZigBee que facilitan el trabajo de desarrollo en el ámbito de la radiofrecuencia.

Las placas disponibles son:

• Placa ZigBee --- ZigBee1v1.0.

• Antena ZigBee ---- ANT – ZIGBEE.

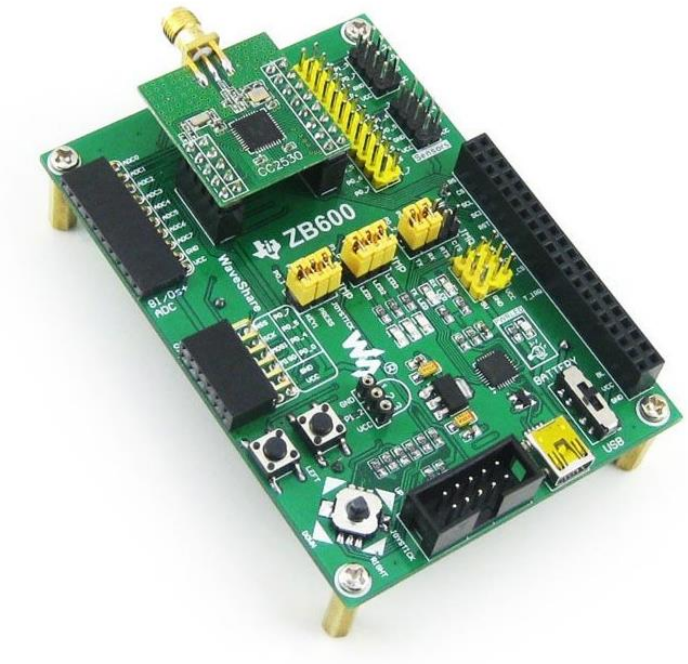


Figura 23: Placa Zigbee. (Fuente: https://es.farnell.com/).

La placa Zigbee está basada en un transceptor de radio frecuencia manejado por un microcontrolador (MC9S08GT32), tiene disponible para el usuario entradas / salidas digitales y analógicas e interfaz I2C y UARTs.

Es ideal para aplicaciones de corto alcance y de bajo consumo a un costo reducido entre las cuales podemos citar: monitoreo de sensores, control de accesos, sistemas de seguridad, etc.

2.3.6.2. Características

- Permite trabajar a los usuarios con estandarización para poder introducir nuevos productos dedicados a la domótica o automatización del hogar.
- Proporciona la red de comunicaciones, seguridad con algoritmos empotrados y servicios de apoyo para la capa de aplicación.
- Tiene un programa de certificación, que asegura que los productos son interoperables y de calidad.
- Destaca el Smart Home que tiene el fin de que los dispositivos funcionen de una manera conjunta de forma inteligente.
- Ideal para aplicaciones de corto alcance y bajo consumo a un bajo precio.

2.4. Comparativa de plataformas HW

Plataforma	Ámbito de aplicación	Ventajas	Desventajas	Microprocesador	Memoria	Alimentación
Raspberry Pi	Servidor web, robótica, domótica	Proyectos de alta complejidad. Bajo coste.	Es difícil de usar en proyectos simples. Poca memoria RAM.	Chipset Broadcom BCM2387 ARM Cortex-A53	1GB LPDDR2	3,3 - 5 V Hasta 2,5 A
Arduino	Prototipos, educación	Extensa documentación Precio. Variedad dispositivos. Ideal para iniciarse.	En proyectos más profesionales quizás no sea la plataforma adecuada.	ATmega328	Modelos de 32Kb y otros como Mega de 256Kb.	5V - 9V Consumo según modelo
Waspmote	Smart City, Smart agricultura	Amplia variedad de sensores. Bajo consumo. Compatibilidad con IDE Arduino.	Diseñado para unos requerimientos específicos. No apta para gente que se está iniciando.	ATMega 1281	128KB	3,3V - 4,2V On: 15mA Sleep: 55uA Hiberna: 0,7ua
Spark	Smart Home, educación, Prototipos	Comunidad al alza. Facilidad de configuración. Precio. Ideal para iniciarse.	Poca variedad de dispositivos propios, pero con compatibilidad con otros fabricantes.	ARM-32-Cortex	128KB	3V-6V Max: 300mA Min: 50mA
Intel Galileo	Prototipos, Educación	Arquitectura Intel. Compatible con Arduino. Potencia de cómputo. Buena opción para iniciarse.	Precio. Documentación escasa.	Intel Quark.SOCx1000	512KB	7V-15V Max: 800mA
Zigbee	Smart Home	Protocolo de comunicación. Bajo consumo. Variedad de sensores y dispositivos.	No recomendable para iniciarse.	Según fabricante	Según fabricante	Según fabricante

2.5. Arquitectura IoT

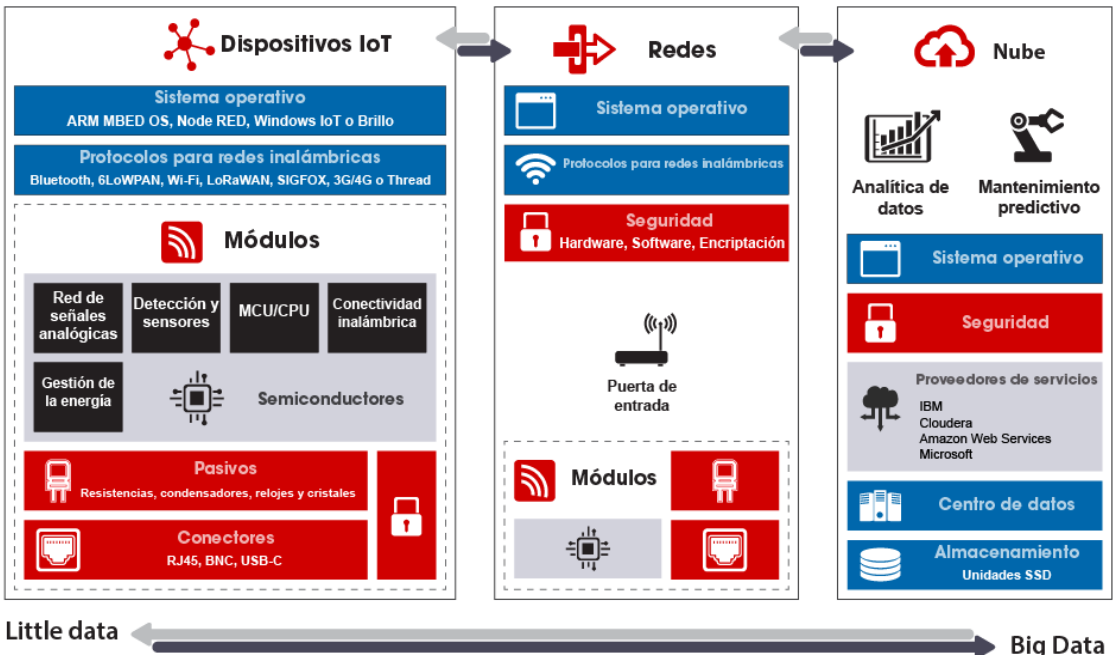


Figura 24: Arquitectura típica IoT.

Con el fin de que una plataforma IoT se considere una opción válida para el desarrollo de un producto de este tipo deberá ser capaz de gestionar de manera efectiva toda la información, esto significa que:

- Esta deberá ser capaz de enviar y recibir de forma correcta toda la información.
- Deberá ser capaz de almacenar y analizar la información.
- Transmitir la información al usuario de manera efectiva.
- Garantizar que los datos se transmiten de forma segura al usuario.

Por lo tanto, una plataforma IoT debería estar constituida, al menos, las siguientes características:

- Conectividad y protocolos: Permitir la conexión mediante protocolos, y la recepción de diferentes formatos de datos en una interfaz que garantice la precisa transmisión de datos y la interacción con los dispositivos. Esta conectividad dispositivo-dispositivo y dispositivo-nube ha de ser estable, segura y rápida, garantizando así una buena comunicación entre los diferentes elementos.

- Almacenamiento de datos: Los datos deben ser almacenados para un posterior análisis, representación o integración con una herramienta propia o de terceros. A menudo, los datos son leídos y enviados directamente a la nube, por lo que es ésta la que se encarga de la parte del almacenamiento.

- Procesamiento y gestión de la acción: Los datos deben ser procesados para, según un conjunto de normas reglas o disparadores, ejecutar acciones dependiendo del valor resultante. Una de las características de la mayoría de proyectos basados en la IoT es el procesamiento y análisis de datos leídos a través de los sensores, comúnmente para hacer que estos datos sean más entendibles.

- Analítica y Visualización: Los datos deben de poder ser analizados y transformados, para luego poder ser visualizados mediante gráficos o expuestos en APIs para aplicaciones externas a la plataforma.

- Gestión de dispositivos: La capacidad de gestionar los dispositivos de una forma flexible, mediante agrupaciones por localización, función u otros criterios, facilita la escalabilidad de las soluciones IoT basadas en una plataforma.

- Herramientas adicionales e interfaces externas: Si la empresa que ha desarrollado la plataforma cuenta con otras soluciones de software que podrían cubrir necesidades de sus clientes, o llega a un trato con algún proveedor de ciertos servicios relacionados, el ecosistema de esta plataforma crecerá y ofrecerá un conjunto de herramientas más completas para el usuario. Además, la capacidad de integrarse con sistemas o servicios de terceros permite suplir necesidades del negocio que no están en la plataforma.

2.5.1. Plataformas seleccionadas

Debido a la gran cantidad de soluciones basadas en las diferentes plataformas Software y los dispositivos Hardware que se han descrito en apartados anteriores, es necesario que quede argumentado en este apartado.

La plataforma y dispositivo que se han elegido sirve para demostrar la viabilidad técnica de un producto mínimo viable.

Para este trabajo experimental, se ha optado por la plataforma Software MyDevices Cayenne, ya que nos ofrece la posibilidad de conectarnos a la nube y poder interconectar los distintos elementos HW que componen el sistema. También nos permite crear diferentes Widgets que se podrán visualizar tanto en el PC como en el Smatphone (mediante App). También los proporciona la capacidad de poder automatizar con diferentes secuencias los dispositivos, activando una determinada salida cuando se lea una determinada entrada.

También se ha elegido Cayenne, debido a la gran escalabilidad y flexibilidad que ofrece. Al hacer un estudio sobre un sistema de alarma, es interesante poder añadir o quitar dispositivos cuando se precise.

Por último, ofrece una interfaz completamente personalizable, que permite visualizar los datos y actuar sobre los diferentes dispositivos de forma remota. Esta plataforma también dispone de un sistema de aviso mendiante correo electrónico o SMS.

Por otra parte, se han utilizado Raspberry Pi y Arduino como dispositivos Hardware.

Se ha optado por el uso de Raspberry Pi debido a que se trata de una placa de muy bajo coste, que además de contar con la posibilidad de utilizar numerosos

sistemas operativos, dispone Wifi/Ethernet (lo que permite establecer una conexión a internet sin necesidad de módulos, antenas, etc…).

Raspberry Pi también dispone de USB (por lo que se podría utilizar una cámara o cualquier periférico que se precise).

Por otro lado, se ha elegido también Arduino para el desarrollo de este prototipo. Los motivos por los que se ha elegido Arduino como principal dispositivo, es que permite la lectura de sensores analógicos. Arduino, al igual que Raspberry Pi, es una placa de bajo coste que dispone de gran cantida de información acerca de proyectos que otros usuarios han ido desarrollando sobre esta placa.

Ambas placas, son soportadas por el software que se ha elegido, y con el fin de hacer un desarrollo más completo, se ha optado por usar una placa Raspberry Pi 3 Model B y un Arduino UNO.

En el Capítulo 4 se describirá la funcionalidad completa de la plataforma Software y de los dispositivos Hardware que se han seleccionado.

CAPÍTULO 3.

PROCESO DE DESARROLLO DE UN PRODUCTO IOT

3. Proceso de desarrollo de un producto IoT

Para el desarrollo de un producto IoT, la primera premisa que se ha de tener en cuenta es la identificación de un problema. Dicho problema puede ser abordado desde distintos puntos de vista, pero como se ha hablado en este Trabajo de Fin de Master, lo que se pretende es buscar una solución basada en la IoT, seleccionando una plataforma Software y un Dispositivo Hardware para conseguir un producto tecnológicamente avanzado que cumpla con las tendencias actuales del mercado electrónico.

Para desarrollar un producto IoT, el siguiente paso es la realización de un Plan de Negocio (aplicando el Lean Startup model).

El plan de negocios es un documento que describe, de manera general, un negocio y el conjunto de estrategias que se implementarán para su éxito. En este sentido, el plan de negocios presenta un análisis del mercado y establece el plan de acción que seguirá para alcanzar el conjunto de objetivos que se ha propuesto.

Como tal, el plan de negocios tiene un uso interno, desde el punto de vista de gestión y planificación, y otro externo, como herramienta de promoción y comunicación de la idea del negocio, bien sea para venderla, bien para obtener financiación.

El plan de negocios, en este sentido, sirve de brújula para el emprendedor, pues de permite tener un mejor entendimiento del negocio, al mismo tiempo que lo obliga a investigar, reflexionar y visualizar todos los factores, tanto internos como externos, que incidirán en la marcha de su negocio.

Para la elaboración de un plan de negocio, se ha de realizar los siguientes estudios.

3.1. Modelo de Negocio

Un modelo de negocio es una herramienta previa al plan de negocio que permitirá definir con claridad qué se va a ofrecer al mercado, de qué forma se va a hacer, a quién se va a vender, cómo se va a vender y de qué forma se van a generar ingresos. Es una herramienta de análisis que permitirá saber cómo se va a realizar el producto, a qué coste, con qué medios y qué fuentes de ingresos se van a tener. Definir el modelo de negocio es saber cómo funciona el negocio,

cómo está hecho, cómo se puede modificar, cómo pulir, cómo cambiar, cómo moldear etc…

Cuando se habla, coloquialmente, de modelo de negocio se suele concretar en la forma que tiene una empresa de ganar dinero. Y también es eso, pero es mucho más. El modelo de negocio habla no sólo de cómo ganar dinero sino también de quiénes son los clientes potenciales, cómo se va a llegar a ellos, qué cosas se tienen que hacer para entregarles la propuesta de valor, qué es lo que te hace único, cuál es la estructura de costes tienes, etc...

Los modelos que están funcionando son aquellos que son capaces de crear valor para el cliente, es decir, que tienen una propuesta de valor clara, que son capaces de llegar al cliente, de diferenciarse, de establecer fuertes lazos con el cliente, de fidelizar y que son capaces de producirlos también de una manera especial.

La manera de validar un modelo de negocio es teniendo clientes que paguen por tu producto y/o servicio. Esa es la manera de validar tu propuesta de valor. ¿Cómo se crea valor? Estando muy cerca del cliente. Estableciendo una relación muy estrecha desde el principio para saber cuáles son sus necesidades o problemas que tienen. Y una vez en el mercado puedes encontrarte con que tu modelo de negocio necesita modificarse. El modelo de negocio puede variar constantemente. De hecho, no cambiar de modelo de negocio o no hacer variaciones importantes es aterrador.

Los puntos más importantes del modelo de negocio son:

- Propuesta de valor: En este punto se añadirán todos los aspectos que convierten el producto especial. Aquí es donde se verá cuál es la diferencia frente a la competencia y que valor añadido se ofrece al mercado.

  Para este proyecto se va a ofrecer un gadget inteligente ubicado en un hogar convencional, el cual proporciona por una parte la posibilidad de tener un sistema de control centralizado, el cual nos permite disponer de un sistema domótico de bajo presupuesto (y controlado desde el PC/Smartphone), y por otra parte nos sirve como sistema de alarma. Todo esto con sistema operativo Linux. Por último, el gadget también cuenta con conexión a internet y conexión remota con sensores y actuadores.

- Relación con los clientes: Aquí se define cómo serán las relaciones con los clientes, de qué tipo, cómo se van a fidelizar, etc…

  Para este modelo de negocio se van a establecer relaciones con los clientes mediante los siguientes medios:

- Página Web: Se dispondrá de una página web que servirá como portal de comunicación con los clientes, ofreciendo toda la información acerca del dispositivo (incluyendo manuales de uso). También se dispondrá de servicio post-venta.

- Redes sociales: Se usarán las redes sociales como canal para la difusión del gadget, además como instrumento de Marketing. P.e. Facebook, Instagram, Twitter, etc…

- Videotutoriales: Se creará un canal de YouTube donde se pondrán a disposición publica numerosos tutoriales de uso del dispositivo, incluyendo también la resolución de posibles dudas.

- Canales: A través de que canales queremos que nuestros clientes sean alcanzados. Estos canales serán los siguientes:

  - Web: Se realizará un posicionamiento web con el fin de dar alcance a la página web.

  - Redes Sociales: Se realizarán campañas de marketing en las redes sociales... P.e. Facebook, Instagram, Twitter, etc…

  - Portales de comercio electrónico: Se realizarán también campañas en numerosos comercios online. P.e. Amazon, Ebay, Aliexpress, etc…

  - YouTube: Se creará contenido audiovisual en esta plataforma para promocionar el gadget.

- Segmentos de clientes: Esto se refiere al público principal al que va dirigido el dispositivo. En este caso el dispositivo está pensado para uso doméstico, por lo que va dirigido a un amplio segmento de la población. Generalmente núcleos familiares, compuestos principalmente por adultos entre 25-60 años e hijos mayores.

- Las actividades clave: corresponden al conjunto de acciones que se requieren para el desarrollo de nuestra propuesta de valor.

Es este caso las actividades claves que se van a realizar serán las siguientes:

- Desarrollo de prototipo: Programación de la Raspberry, instalación de sistema operativo, conexionado de sensores y actuadores, pruebas de funcionamiento, etc…

- Producto definitivo: Desarrollo del producto en masa, creación de PCB, marcado CE, empaquetamiento final del producto, búsqueda de proveedores, fabricación del dispositivo.

- Marketing: Creación de campañas que permitan dar el alcance necesario al dispositivo.

- Recursos clave: Son todos los recursos que nuestro negocio requiere para la propuesta de valor, para los canales, para las relaciones de nuestros clientes, nuestra fuente de ingresos.
  Los recursos necesarios para la creación del negocio serán los siguientes:

  - Almacenes: Almacén necesario para tener disponibles productos en stock.

  - Empresa de transporte: Empresa que se dedique a la realización de los envíos del producto.

  - Infraestructura para la venta online: Recursos necesarios para poder realizar la venta online

  - Financiación: Financiación necesaria para poder realizar el producto final

- Socios claves: principales aliados:

  - Socios clave: Tiendas de E-commerce que puedan distribuir nuestro producto.

  - Proveedores clave: compraremos la materia prima (Raspberry Pi + fuente de alimentación + tarjeta SD + sensores/actuadores) a distribuidores como RS Components. También serán claves los proveedores de los elementos del packaging.

- Recursos clave: Raspberry Pi, fuente de alimentación, tarjeta SD, sensores y actuadores.

- Actividades de los socios clave: distribución del producto.

- Estructura de costes: los principales costes quedarán divididos en:

  - Diseño del prototipo: Costes de desarrollo de prototipo, materiales.

  - Fabricación en serie: Costes de fabricación en serie, proveedores, ensamblajes.

  - Campaña de marketing: Costes de posicionamiento web, campañas de marketing en diferentes medios, etc…

- Fuente de ingresos:

  - Venta del producto: Las unidades vendidas supondrán prácticamente la totalidad de los ingresos.

  - Anuncios: Posibles ingresos recibidos de anuncios, pagina web, donaciones, etc…

  - Subvenciones públicas: Se realizará una campaña de información consistente en buscar ayudas y subvenciones públicas para lanzar el producto.

3.2. Estudio de Mercado

En este apartado se pretende ubicar el dispositivo propuesto dentro de un mercado global, observando cuál es su competencia directa y viendo también cuales son las principales ventajas y desventajas respecto a sus competidores directos.

En España, la apuesta por el eCommerce es evidente, y las perspectivas para el 2018 ya sitúan las ventas en datos cercanos a los 32.000 millones de euros. España es el tercer país europeo en ventas eCommerce a través de la electrónica, un dato que indica la buena salud del sector en el país.

En la **Figura 25** se puede ver un gráfico donde observamos la tendencia de ventas

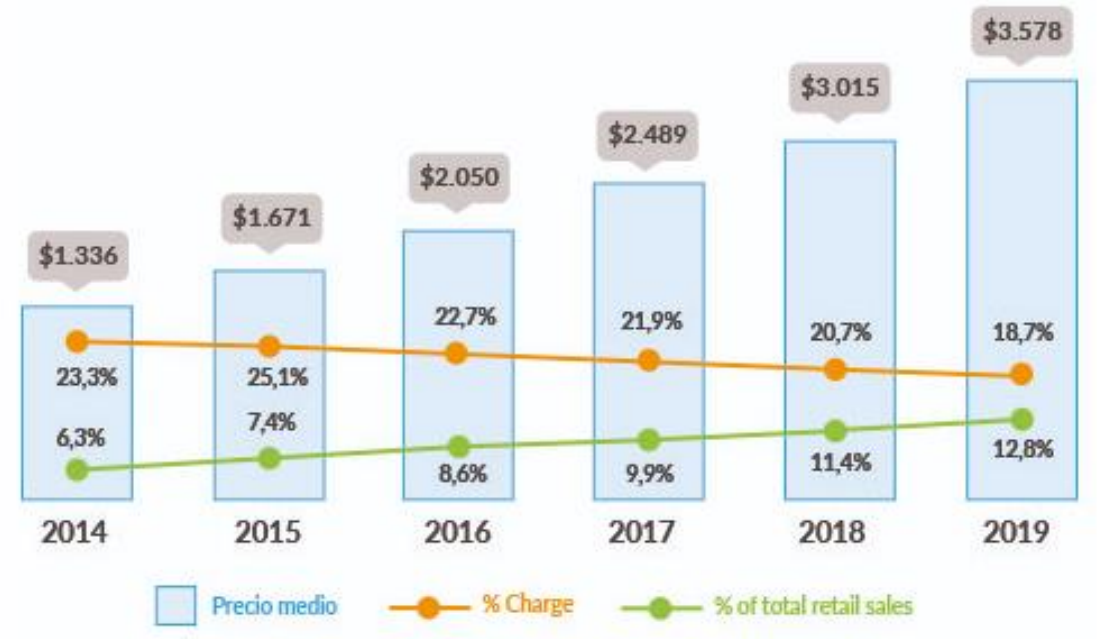
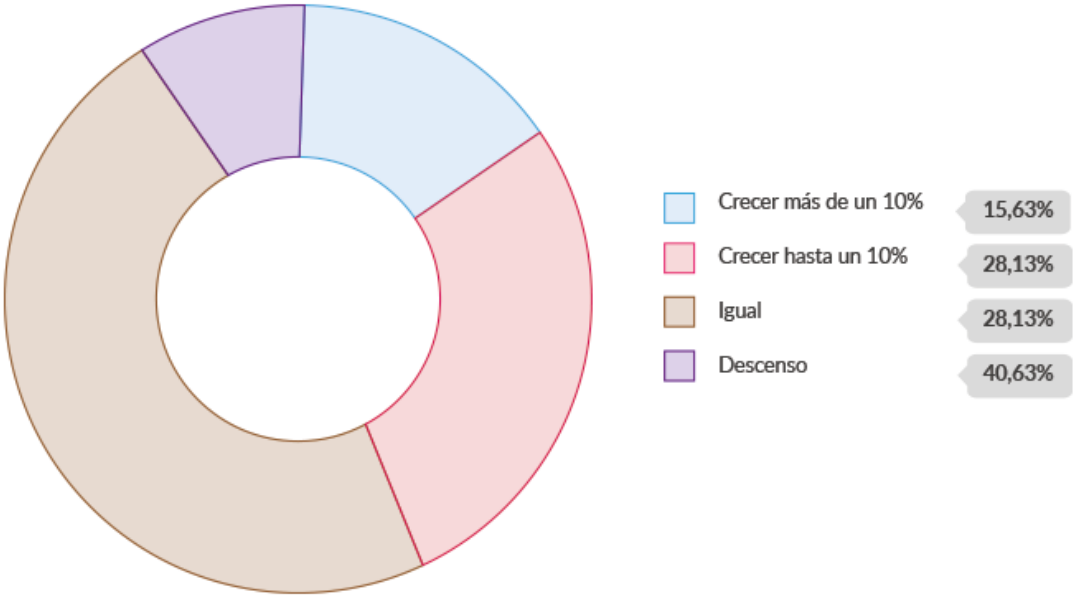


Figura 25: Ventas a nivel mundial de electrónica. (Fuente: Observatorio Ecommeece)

Las perspectivas de ventas de las empresas de comercio electrónico van al alza en los próximos años. Además del incremento del número de compradores, un

elemento esencial es que el valor medio de la cesta (compra) de cada usuario también se incrementará. Por lo tanto, habrá más clientes y la gran mayoría desembolsarán más dinero en el eCommerce.

En la **Figura 26** se puede observar un gráfico donde se puede apreciar la evolución que se estima para el comercio de electrónica, quedando reflejado la tendencia al continuo crecimiento.



Fuente: Observatorio eCommerce

Figura 26: Previsión de la evolución en el comercio electrónico.

Como se ha visto en este punto el comercio electrónico es un mercado que se encuentra en plena evolución. Por lo que a priori se puede ver que el producto objeto de estudio en este trabajo de Fin de Máster será introducido en un mercado de rápida evolución y en constante crecimiento.

A continuación, se verá cómo es el mercado al que va destinado este dispositivo, viendo sus características geográficas, demográficas y características de nivel socioeconómico.

- Características geográficas: El comercio electrónico en España está distribuido geográficamente siguiendo las tendencias de la siguiente figura. Observando que desde España la mayoría del comercio electrónico se realizará en la Unión Europea. **Figura 27.**

- Demografía: Se trata de un público joven y mediana edad, desde los 18 hasta los 45 años, ya que estos son el público objetivo al que va dirigido esta tecnología; sin embargo, debido a su sencillo funcionamiento y su gran funcionalidad, podría ampliarse el límite de edad hasta los 50-60 años.

- Nivel socioeconómico: Se trata de un producto de bajo precio (entre los 50 y los 100€, por lo que este dispositivo resulta accesible para cualquier persona con un poder adquisitivo medio.

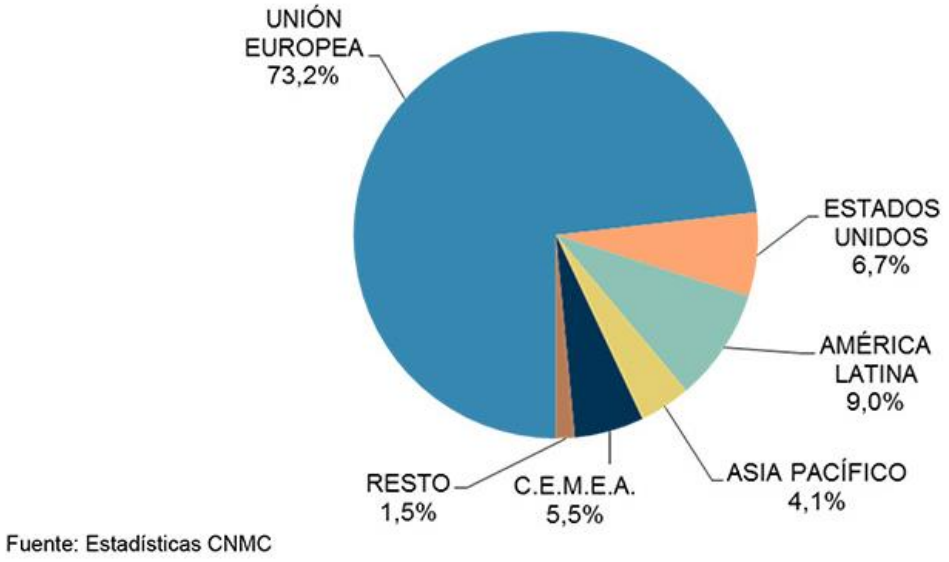


Figura 27: Distribución del volumen de negocio del comercio electrónico desde el exterior de España por áreas geográficas.

La competencia directa de este producto sería la de un dispositivo de alarma profesional, con todos los servicios que incluyen. Por contrapartida cabe destacar que su precio es mucho más elevado (además de disponer de una cuota mensual).

Las principales ventajas y desventajas de este producto frente a la competencia son las siguientes:

- Ventajas: Precio reducido, en torno a los 50-100 €, integración sencilla, incorporación de controlador inteligente gran versatilidad y bajo consumo.

- Desventajas: Se necesita conexión a Internet para que recibamos la notificación correspondiente a nuestro Smartphone.

3.3. Plan de desarrollo de Clientes

En este apartado se tratarán los aspectos fundamentales que caracterizan la captación de clientes del producto.

El primer aspecto que hay que destacar es la estimación de ventas del dispositivo, la cual se puede observar en la **Figura 28**, para la realización de esta estimación se ha tenido en cuenta la venta de dispositivos similares en su lanzamiento. Teniendo en cuenta que se trata de un dispositivo innovador y desconocido se han estimado las siguientes ventas:

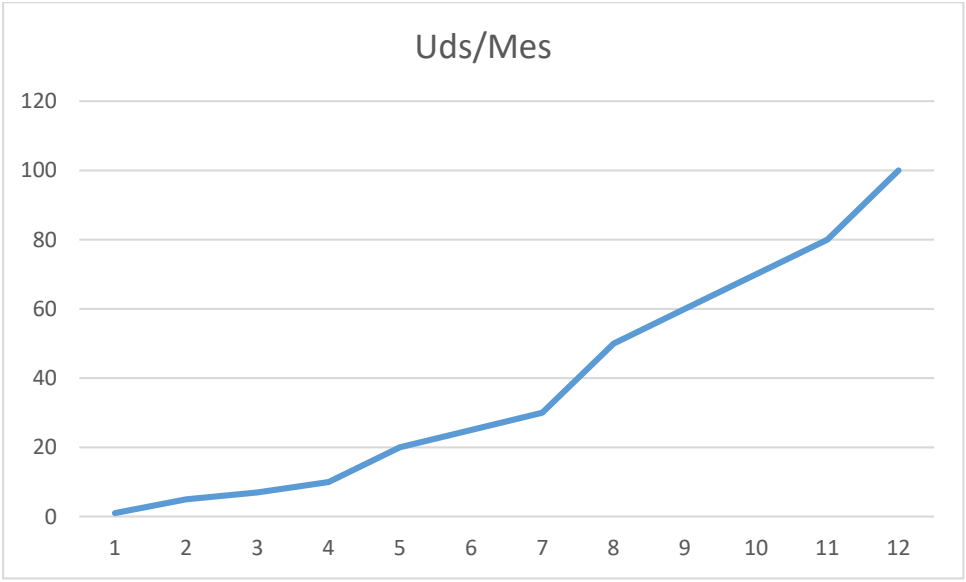


Figura 28: Estimación de ventas del primer año.

La intención que se pretende en todo momento es mantener a los clientes satisfechos, y no solo por el valor que le añadimos a nuestro producto, sino por los siguientes motivos:

- Atención personalizada.
- Calidad del producto ofertado.
- Calidad de la comunicación con los clientes.
- Videoturoriales y soporte online.

En la **Figura 29** se muestra un gráfico de la estimación de los clientes satisfechos.

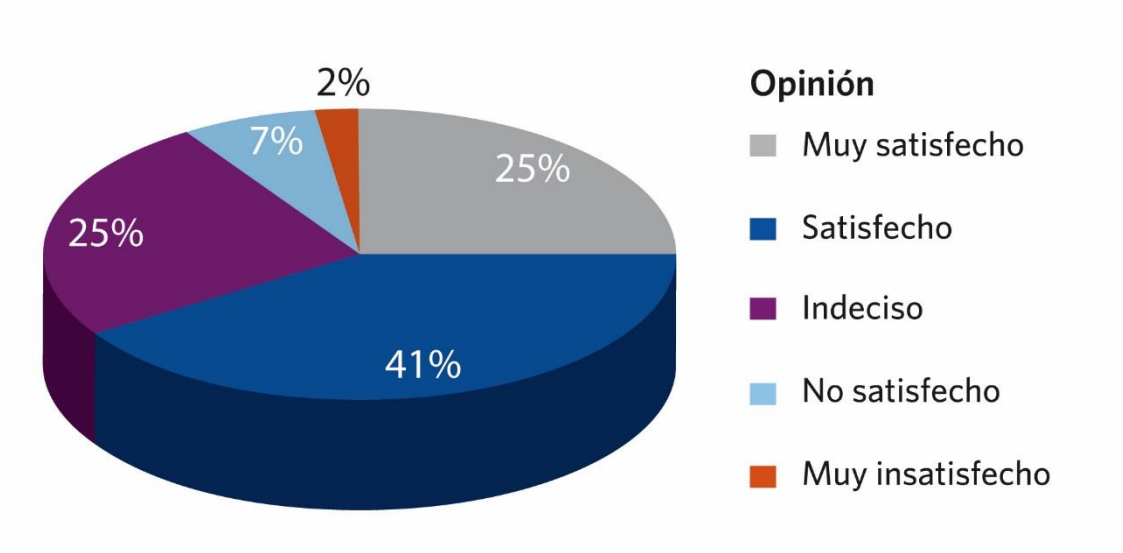


Figura 29: Porcentaje de clientes satisfechos.

En cuanto a lo referente al posicionamiento en el mercado cabe destacar lo siguiente:

- Mercado evolutivo: Será fácil el posicionamiento dentro del comercio electrónico debido a su constante y creciente evolución.

- Producto único: Propuesta de valor única e innovadora en el mercado, con interesantes ventajas respecto a su competencia.

- Gadgets: Nos posicionaremos dentro de este segmento, al tratarse de un complemento o gadget.

Por otra parte, se sabe que para dar a conocer el producto a un mercado internacional es necesario realizar campañas de marketing. Para este negocio se van a realizar campañas de marketing a través de los siguientes medios:

- Web: Se realizará un posicionamiento web con el fin de dar alcance a la página web.

- Redes Sociales: Se realizarán campañas de marketing en las redes sociales... P.e. Facebook, Instagram, Twitter, etc…

- Portales de comercio electrónico: Se realizarán también campañas en numerosos comercios online. P.e. Amazon, Ebay, Aliexpress,etc…

- YouTube: Se creará contenido audiovisual en esta plataforma para promocionar el gadget.

Por último, en este apartado, se deben de incluir los métodos usados para la denominación del precio del producto. Para establecer el precio se ha tenido en cuenta lo siguiente:

- Basado en costes: El precio de los componentes será un gran factor determinante del precio final.

- Basado en valor: La funcionalidad del dispositivo también hace que se pueda elevar el coste, aumentando los beneficios.

- Basado en competencia: No se dispone de una competencia directa (sin productos similares) por lo que no se ve mermado por compañías que distribuyan el mismo producto.

En la **Figura 30** se encuentra la estructura de costes.

COSTO DE PRODUCCION		PRECIO FABRICACION UNIDAD
Mano de obra	3	**49,55**
Materia prima	37	
		PRECIO DE VENTA (40% BENEFICIO)
Gastos de fabricación	3	
TOTAL	**43**	**82,5**
GASTOS DE VENTAS		
Personal	1	
Materiales	1,5	
Publicidad	0,5	
Otros	3	
TOTAL	**6**	
GASTOS FINANCIEROS		
Intereses	0,5	
Otros	0,05	
TOTAL	**0,55**	

Figura 30: Estructura de costes.

En la figura anterior se puede ver que se ha establecido un precio de unos 82€ para la venta de un dispositivo, manteniendo así un beneficio del 40%.

3.4. Plan de Operaciones

El primer paso para realizar un plan de operaciones es establecer una buena propuesta estratégica en cuanto a lo que se refiere a la externalización.

Como ya se sabe, la externalización supone un ahorro de costes, además supone una gran ventaja a la hora de buscar profesionales en alguno de los procesos de nuestro producto. Sin embargo, debido a que la carga técnica del producto reside en la programación y el desarrollo SW, se ha optado por no externalizar ninguno de los procesos para la fabricación del dispositivo. **Figura 31.**

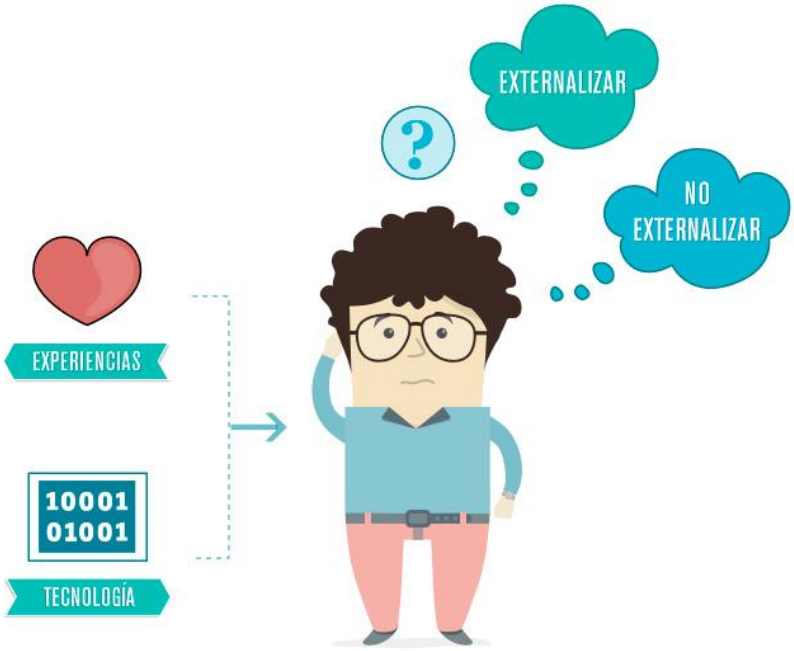


Figura 31: Externalización. (Fuente: https://www.transportfredgirones.com)

Uno de los aspectos fundamentales del plan de operaciones es establecer el flujo de trabajo. En la **Figura 32** se puede ver gráficamente como es el flujo de fabricación del dispositivo.

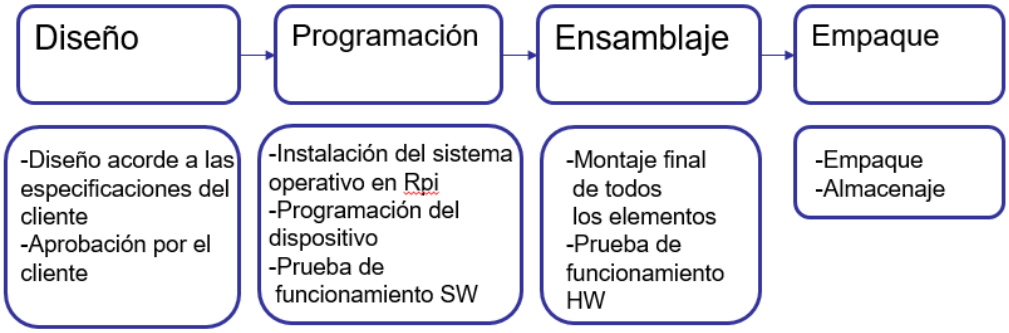


Figura 32: Flujograma de proceso productivo.

Para finalizar este apartado, se pondrá en manifiesto la importancia de asegurar la calidad del producto, para ello, se harán test para la comprobación de que todo está correcto.

- Test de funcionamiento SW: Comprobación de que la programación es correcta, atendiendo a todas las casuísticas.
- Test de funcionamiento HW: Comprobación de que todos los sensores y actuadores funcionan correctamente.
- Test de caída del dispositivo: se verificará que el dispositivo resista ante una caída a cierta distancia.

3.5. Plan de Organización

En este punto tenemos que dejar reflejado como quedará estructurada la empresa.

Para comenzar, en la **Figura 33** veremos cómo está constituida la empresa.

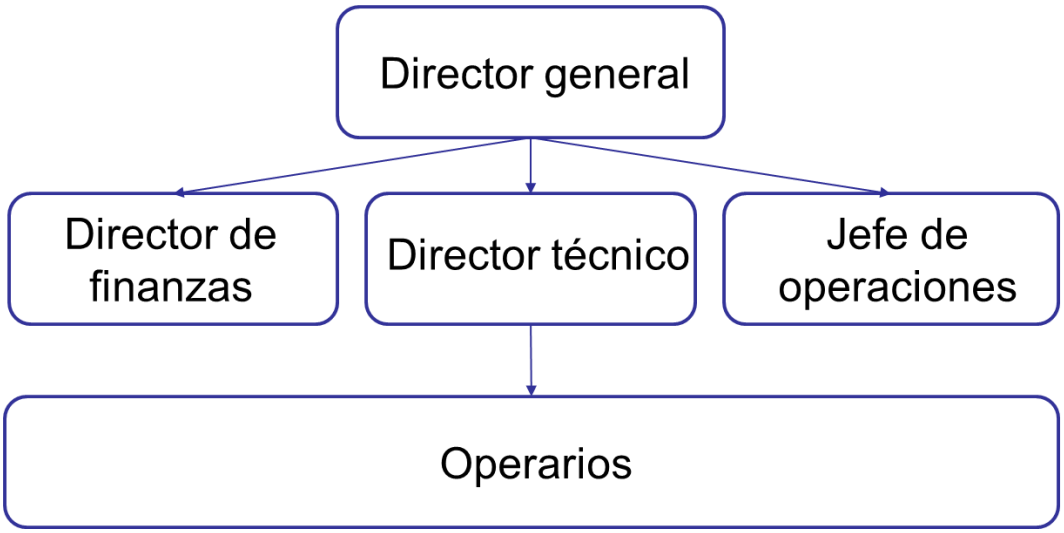


Figura 33: Organigrama de la empresa.

Como se puede ver en la figura anterior, en la empresa se dispondrá de un director general, puesto que estará ocupado por uno de los 4 socios constituyentes.

Los otros 3 socios ocuparán los puestos de Director financiero, el cual se encargará de gestionar la parte financiera de la empresa, Director técnico, que se encargará de la parte técnica, y, por último, un jefe de operaciones, encargado de la parte de compras y ventas.

Por último, se dispondrán de operarios encargados de montaje y programación de los dispositivos. Estos últimos también serán encargados de dar soporte técnico y de atención al cliente.

Se ha elegido la configuración de sociedad limitada debido a la sencillez burocrática y debido también al bajo capital social exigido (3000€).

Para la constitución de una Sociedad de Responsabilidad Limitada, se deben llevar a cabo los pasos reflejados en la **Figura 34.**

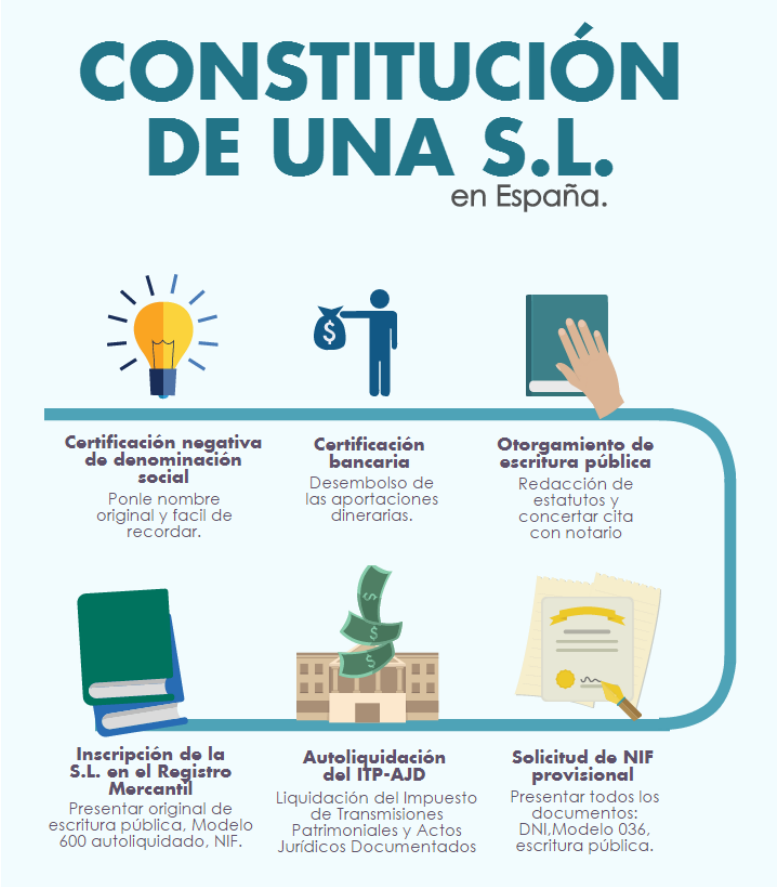


Figura 34: Constitución de una S.L. (Fuente: www.gesron.es)

En este apartado también se ha que hablar de la estructura de costes. Para el presente dispositivo se ha estimado necesaria la inversión que vemos en la **Figura 35.**

Costes salariales del proyecto	5000
Costes de recursos de proyecto	175
Coste de fabricación de 500 ud	12500
Costes de constitución de empresa	3000
Costes de campaña de marketing	500
	21175

Figura 35: Inversión inicial necesaria.

El flujo de caja se muestra en la **Figura 36**, como podemos ver en los dos primeros meses se observan pérdidas, en parte es debido a que inicialmente se fabrica un lote de 500ud. Es por eso por lo que no es necesario fabricar más unidades hasta el tercer año.

INGRESOS	0	8200	14760	20500	28700	38500
GASTOS						
INVERSION	21175	0	0	0	0	0
Gastos administrativos	1000	1000	1000	1000	1000	1000
Gastos en marketing	500	500	500	500	500	500
Gastos en fabricación	0	0	0	6000	8750	11750
BALANCE	-21175	-7915	5345	18345	36795	62045

Figura 36: Flujo de caja en los primeros 5 años

El periodo de recuperación se define como el número esperado de años que se requieren para que se recupere una inversión original.

En la figura anterior podemos observar que la tasa de recuperación de capital es de 1 año y medio aproximadamente.

La Tasa Interna de Retorno (TIR) es la tasa de interés o rentabilidad que ofrece una inversión. Es decir, es el porcentaje de beneficio o pérdida que tendrá una inversión para las cantidades que no se han retirado del proyecto.

En la **Figura 37**, observamos que el TIR es del 39.77%

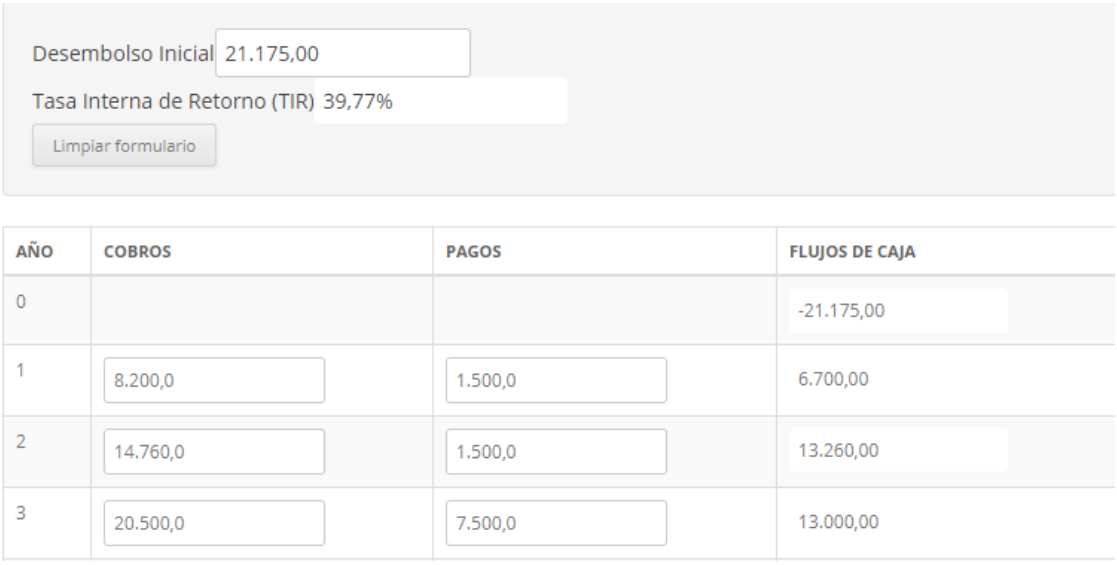


Figura 37: TIR.

El Valor Actual Neto (VAN) es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión.

En la **Figura 38** observamos que nuestro VAN es de 18.73€

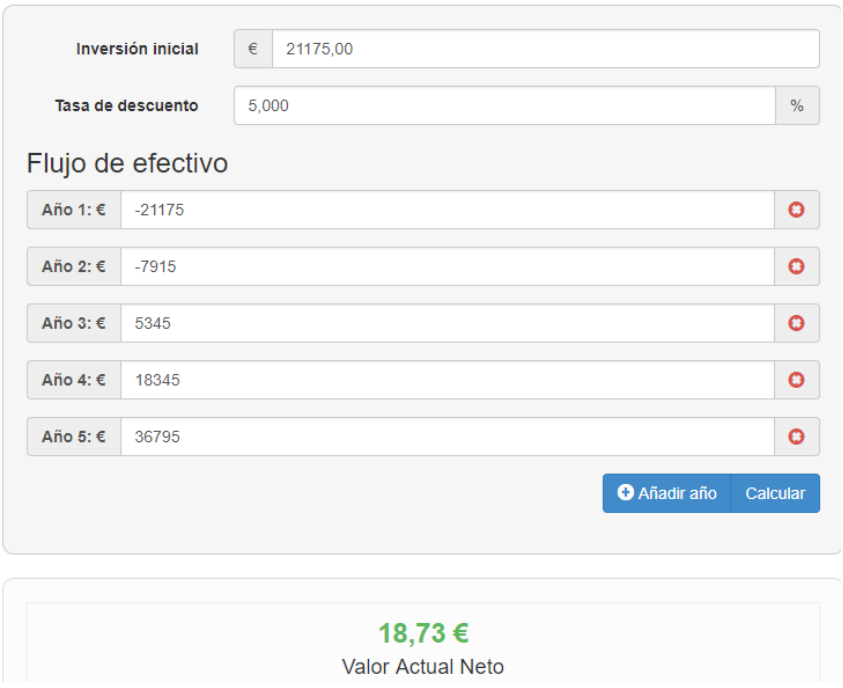


Figura 38: VAN.

3.6. Producto Mínimo Viable

Un producto mínimo viable reúne las características suficientes como para satisfacer las necesidades de los clientes; necesidades de las que ya se ha hablado en los apartados anteriores.

En el modelo Lean Startup, el conocimiento acerca del éxito que se pueda tener en el mercado, se obtiene en parte a la experimentación. Esto es, se puede obtener una realimentación de la satisfacción de los clientes a través del lanzamiento de un producto mínimo viable (o varias versiones del mismo).

El producto mínimo viable, también se traduce en el producto más barato y rápido que se pueda construir, con el fin de que, con poco coste y en un tiempo muy corto se pueda observar el comportamiento del cliente, y se pueda crear la necesidad y deseo sobre tu producto.

Según Eric Ries, el cliclo de vida que tiene el producto es el que aparece en la **Figura 39**.

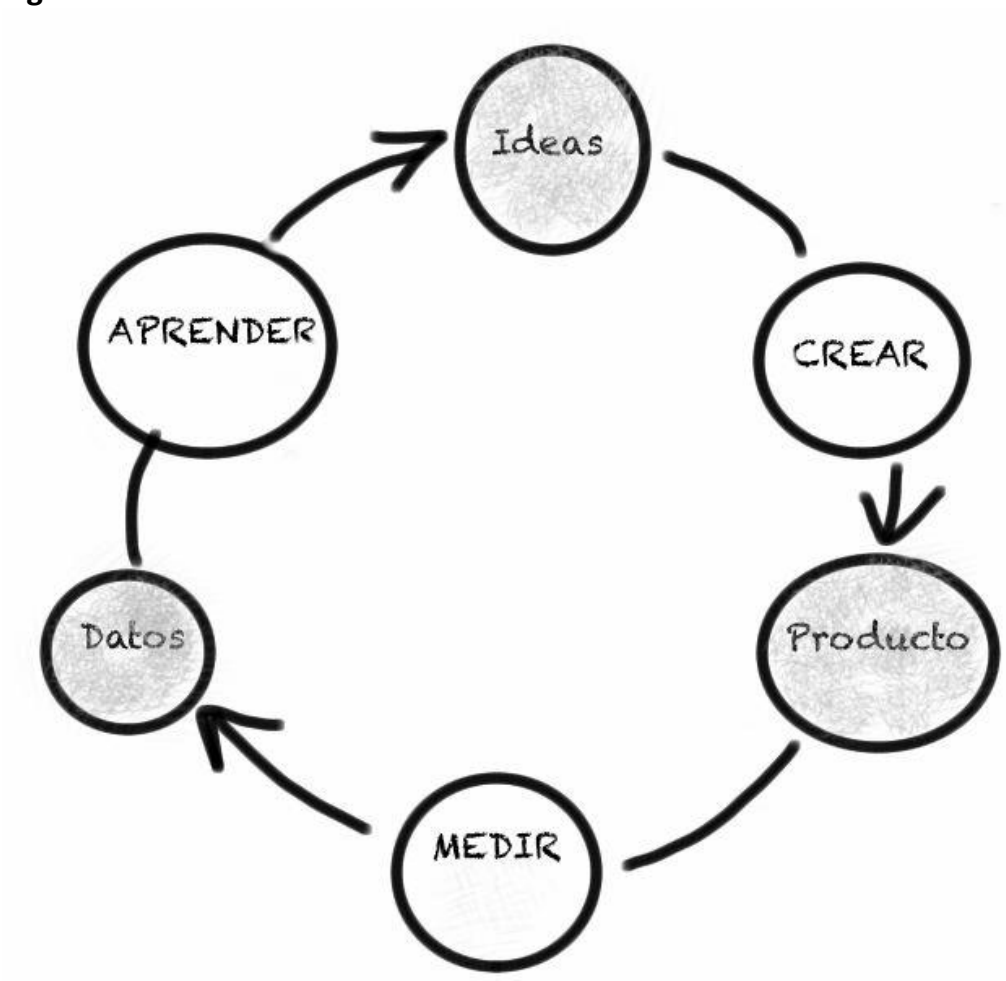


Figura 39: Ciclo de vida del producto. (Fuente: http://alexosterwalder.com/)

Como ya se ha comentado, existen varias formas de experimentar una vez se tenga el producto mínimo viable. Se puede lanzar/promocionar el producto mediante Land Pages o páginas de aterrizaje, se pueden hacer entrevistas a usuarios con el fin de conocer la opinión global sobre el producto, etc…

También cabe destacar la diferencia que existe entre un prototipo y un PMV (Producto Mínimo Viable). Un prototipo es una representación visual o real de lo que se pretende lanzar al mercado, mientras que un PMV es una versión beta del producto que permite obtener feedback para volver a lanzar la versión definitiva.

La creación de un PMV permite conversar con los primeros clientes y saber si estos, están dispuestos a pagar por el producto en cuestión. También sirve para evitar crear un producto que no genere interés sobre el público, no desperdiciar recursos en la fabricación de un dispositivo IoT y poder empezar a comercializar un producto sobre unas bases sólidas.

El principal problema que existe una vez se llega a este punto de desarrollo de un producto IoT es la viabilidad comercial, económica, financiera y legal (ya descritas en este capítulo. Además de estos tipos de viabilidad se ha de demostrar la viabilidad técnica del producto, es decir, demostrar que el producto es viable desde el punto de vista tecnológico.

En los siguientes capítulos se demostrará que el producto objeto de estudio de este TFM es viable desde el punto de vista tecnológico.

Si se sigue el Lean StartUp Model, vemos que el paso que hay después de la elaboración del plan de negocio es la creación de un PMV. Esto puede tener por contrapartida una gran pérdida de tiempo, pues a menudo, se realiza el plan de negocio, pero cuando llega la hora de materializarlo se comprueba que no es viable tecnológicamente.

La principal propuesta de este trabajo de Fin de Máster, es utilizar las plataformas software y los dispositivos hardware descritos en el Capítulo 2 para demostrar (antes de elaborar un plan de negocio) si el producto IoT que se pretende comercializar es viable tecnológicamente hablando.

Todas estas herramientas (plataformas software + dispositivos) permiten realizar un prototipo funcional, que demuestre desde la fase inicial del desarrollo que el producto final es viable tecnológicamente.

En el Capítulo 6 se pueden leer todas las conclusiones.

CAPÍTULO 4.

TRABAJO EXPERIMENTAL

4. Propuesta de dispositivo.

En este capítulo se pretende definir al completo la funcionalidad y las especificaciones de este dispositivo. Para ello se hablará primero de la descripción general de este gadget (que es lo que se propone con él y qué ventajas tiene respecto a la competencia). Por otra parte, se hablará específicamente de las funcionalidades de este dispositivo, al detalle, explicando también la arquitectura IoT que se ha utilizado. Además, se definirán la funcionalidad de los distintos módulos que incorpora.

Finalmente se dará una visión general de las ventajas y desventajas de respecto a otros dispositivos comerciales.

4.1. Descripción general

Como se ha comentado anteriormente, el dispositivo tiene un objetivo doble; por una parte, se pretende crear un sistema de alarma de bajo presupuesto, pero completamente funcional, y por otra parte se implementarán en el mismo dispositivo diferentes funciones domóticas (las cuales se explicarán a lo largo de este capítulo), todo ello trabajando sobre una plataforma IoT. Otra de las principales características a destacar es que los distintos módulos de los que estará compuesto tendrán conexión WiFi, por lo que se podrá controlar, monitorizar y programar todos los sensores y actuadores del sistema a través del PC o por medio de un Smartphone con una App de forma remota.

4.2. Especificación funcional de diseño

Como se ha visto en el apartado 2.1.16., Cayenne es una plataforma de IoT orientada a simplificar la creación de soluciones para el mundo conectado. Esta plataforma destaca por la sencillez a la hora conectar dispositivos entre sí, además de permitir controlar, monitorizar y programar (accediendo directamente a la GPIO) de forma remota.

La plataforma utiliza una API MQTT para gestionar la conexión y envío de datos. Para utilizar esta API son necesarias las librerías y los controladores de sensores proporcionados por myDevices.

Al ser necesario el uso de las librerías, la compatibilidad de hardware es bastante reducida por el momento, aunque, sí permite utilizar Arduino y Raspberry Pi.

La arquitectura utilizada para nuestro dispositivo es la que aparece en la **Figura 40.**

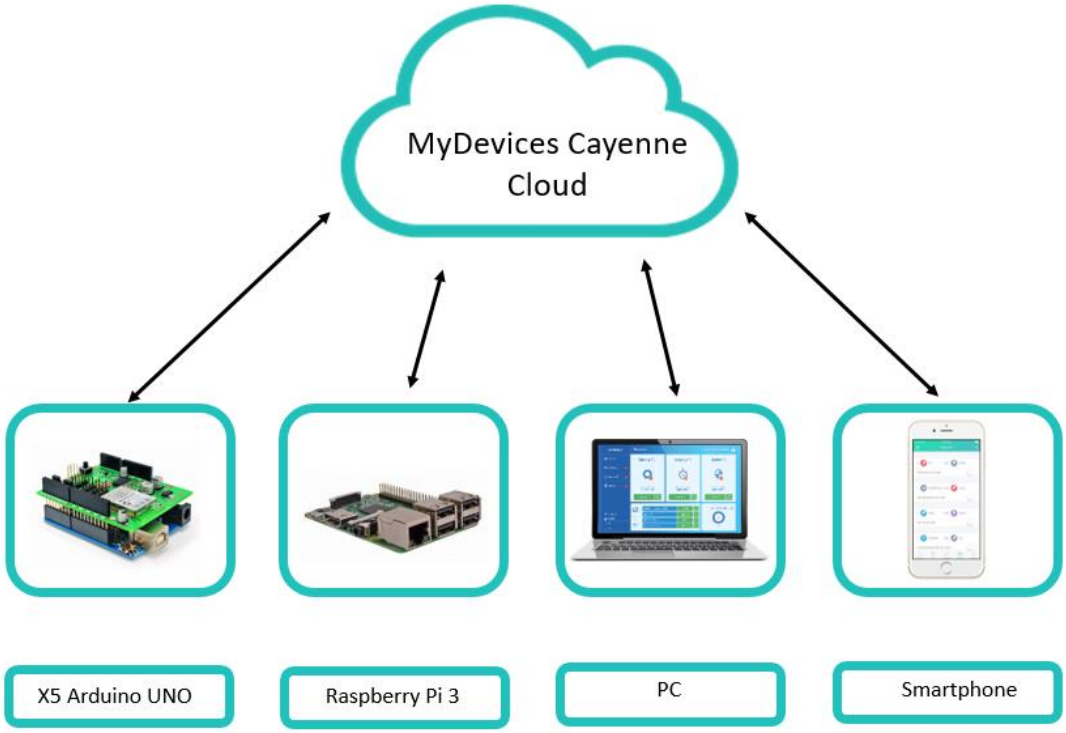


Figura 40: Arquitectura del sistema de alarma

Como se puede observar en la imagen anterior, el sistema de alarma propuesto basado en IoT sigue un modelo de sistema de control distribuido. Como se puede apreciar, todos los dispositivos envían y reciben datos de la nube, en este caso de MyDevices Cayenne. La principal ventaja que se obtiene con esto, es que indirectamente, todos los dispositivos están interconectados entre sí. Es por esto por lo que, desde cualquiera de los dispositivos conectados a la nube, se podrán enviar o recibir datos a cualquiera de los otros elementos.

Dentro del diseño propuesto se utilizará una Raspberry Pi, utilizada en este caso como "Maestro" de nuestro sistema. Dentro de un sistema de alarma convencional, la Raspberry Pi equivaldría a una centralita.

Por otra parte, se van a utilizar 5 Arduinos como módulos sensoriales "esclavos". Más adelante se explicará por qué se ha optado por este diseño.

Este diseño será utilizado para una vivienda de tamaño medio (unos 120m2) y como se ha comentado se compondrá de una Raspberry Pi 3 y 5 Arduinos UNO. Además, para el control y la monitorización del sistema se ha de disponer de un PC o un Smartphone.

La Raspberry Pi 3 "Maestra" se ubicará en una posición estratégica, donde se ubicaría una centralita habitual de un sistema de alarma tradicional. Las funciones que va a llevar a cabo la Raspberry serán las siguientes:

- Aviso acústico.
- Aviso luminoso de diferentes colores (dependiendo del tipo de aviso)
- Detector de Presencia
- Detector de Temperatura y Humedad Relativa
- Detector de Calidad del Aire
- Videovigilancia (cámara USB)

En el Capítulo 5 se verán los sensores y actuadores que se conectarán a la Raspberry y como se programarán para que cumplan esas funciones.

Los 5 Arduinos UNO, se encontrarán repartidos por el resto de habitaciones de la vivienda, en este caso, entrada, cocina, garaje y dos dormitorios.

Los Arduino UNO, serán utilizados para adquirir los datos del habitáculo, y enviárlos a la Raspberry, actuando ésta en consecuencia. En el Capítulo 5 se explicará por qué se ha optado por esta placa para la realización de esta función.

Las funciones de los Arduino Uno, serán las siguientes:

- Detector de Temperatura y Humedad
- Detector de Presencia
- Sensor Magnético para apertura de puerta (en la entrada)
- Sensor de vibración (en los dormitorios con ventana)
- Detector de calidad del Aire
- Regulación de la Luz automática

En el caso de la Raspberry, se dispondrá de conexión a Internet mediante Ethernet, ya que esta es necesaria para enviar y recibir datos de la nube. En la **Figura 41** podemos la Raspberry utilizada.

Figura 41: Raspberry Pi3.

El Arduino UNO, debe tener conexión a Internet, para ello deberemos de utilizar el WiFi Shield de Arduino (podemos verlo en la **Figura 42**).

Figura 42: Arduino UNO + WiFi Shield.

Para controlar y monitorizar todo el sistema se utilizará la herramienta *Cayenne.* En la **Figura 43** se puede ver una captura de pantalla del PC, donde se ve un panel de control y monitorización de ejemplo. En esta figura se puede ver como aparecen la temperatura, la indicación de presencia y demás elementos a controlar/monitorizar. A través de este panel, se pueden observar y controlar todos los parámetros (tanto de la Raspberry Pi como Arduino).

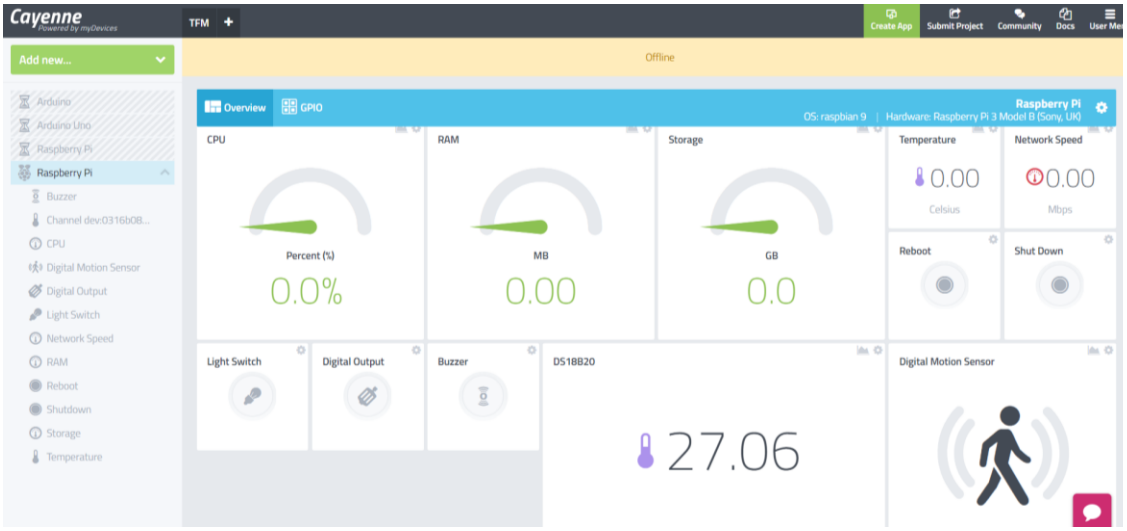


Figura 43: Captura de pantalla del panel de control.

Este mismo panel, se mostrará en la App de MyDevices Cayenne en un Smartphone Android o Apple. Esto ofrece la posibilidad de observar el estado del sistema desde cualquier sitio (con acceso a internet).
A continuación, en la **Figura 44**, se representan las causas y efectos que se originan entre los distintos módulos. Estas causas producen estos efectos automáticamente.

CAUSAS	TRIP POINT	EFECTOS
Temperatura de DS18B20	60º	Aviso luminoso
Detector de presencia activado (detecta presencia)	ON	Aviso acústico
Sensor mangético activado (detecta puerta abierta), Contacto normalmente cerrado	OFF	Videocámara activada
Sensor de vibración (detecta vibraciones en las ventanas)	ON	Envío de notificación a SmartPhone
Detector de calidad de aire activado (detecta CO2)	ON	Iluminación al mínimo

Figura 44: Tabla de disparo automático de la alarma

En definitiva, además de las características que se muestran en la tabla anterior, el sistema de alarma también tendrá las siguientes funciones:

- Conseguir la comunicación inalámbrica entre los sensores y el controlador con una arquitectura Maestro-esclavo entre Raspberry y Arduino.
- Monitorizar el estado del controlador (nivel de uso de CPU, RAM y temperatura del procesador).
- Botón SW para rearme de la alarma de forma remota.
- Control manual y automático de iluminación.
- Lectura de la temperatura en Tiempo Real.
- Utilizar el sensor PIR para el control de la iluminación.
- Monitorización del estado de la red Wireless.
- Intervención directa sobre los distintos actuadores del sistema.

4.3. Módulos Raspeberry Pi/Arduino

Como se ha comentado a lo largo de este trabajo, una de las principales características del dispositivo que se presenta en este trabajo es la flexibilidad y la escalabilidad del sistema. Presentando un modelo de dispositivos basado en módulos.

En el módulo Raspberry se utilizará un sensor de temperatura DS18B20 que tiene un protocolo de entrada Digital. También contará con una entrada de calidad de aire digital, la cual nos dará alarma con la presencia de CO2. La Raspberry (usada como maestra, como se ha comentado en apartados anteriores) dispondrá de una bocina y de unas luces que servirán como aviso ante cualquier tipo de alarma (ver **Figura 44**, disparos de alarma). También tendrá conectado un detector de presencia PIR, cableado a una entrada digital también (configurado desde la aplicación de MyDevices Cayenne). Contará además con un botón físico para rearmar la alarma en caso de disparo, y también un botón software con la misma función.

Por otra parte, se conectará una webcam al puerto USB de la Raspberry, que será utilizado para tomar imágenes en caso de que se dispare la alarma.

A los módulos Arduinos (utilizados como esclavos para la adquisición de datos en los diferentes puntos del hogar) se les conectará un sensor PIR para detectar la

presencia, un módulo DHT11 para medir la temperatura y la humedad relativa, una fotoresistencia para poder controlar la iluminación, sensores de vibración de ventanas, sensor mágnetico de apertura de puertas y salida a relé para encender o apagar la luz del habitáculo.

Es por esto por lo que se distinguen dos módulos principales, el módulo Raspberry Pi utilizado como maestro, donde se conectarán cámaras y avisos en caso de disparo de la alarma. Y, por otro lado, los módulos Arduinos "esclavos" que se utilizarán como tarjetas de adquisición de datos en las diferentes habitaciones.

Se ha optado por este reparto de módulos, ya que se pretende aislar la centralita (Raspberry Pi) en un sitio seguro, y, por otra parte, se quiere dar cobertura a un gran número de habitaciones, por lo que se necesita adquirir los datos y comunicarlos a la centralita.

En la siguiente figura se puede ver un diseño para un hogar de pequeño tamaño, donde se observa la distribución de los distintos dispositivos que conforman el sistema de alarma basado en IoT.

En esta figura se puede ver que la Raspberry se ha dispuesto en el dormitorio principal. También se han colocado Arduino UNO en cada habitación para poder monitorizar todas las habitaciones de la casa.

Para completar el sistema de alarma, se han dispuesto dos cámaras y una sirena que se activarán en caso de disparo de la alarma.

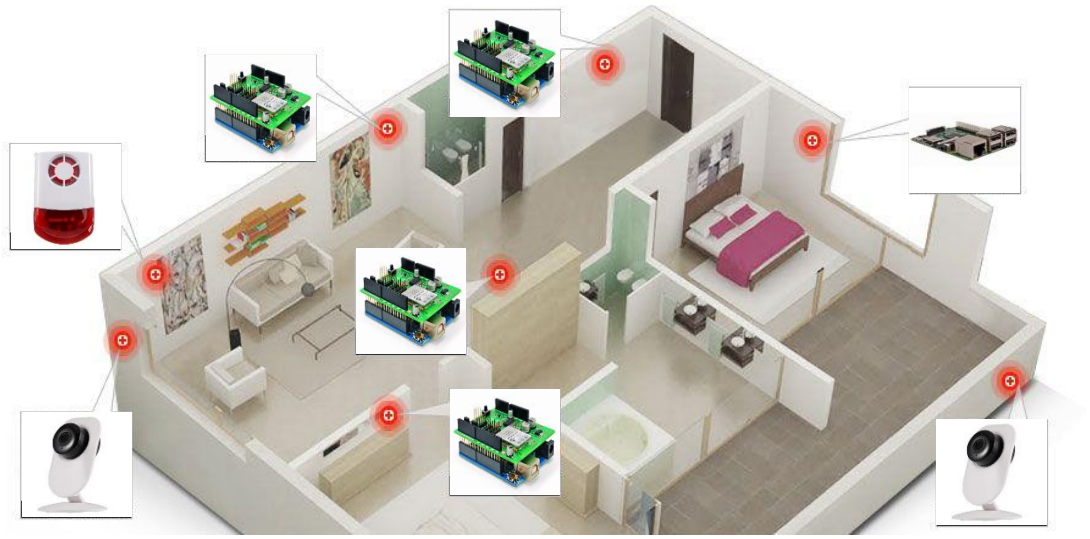


Figura 45: Disposición de dispositivos.

4.4. Sensores y actuadores

En este apartado se muestran los sensores y actuadores que se han utilizados en este proyecto, explicando cómo funcionan y para qué se utilizan.

4.4.1. Detector de movimiento pasaivo (PIR)

Los detectores PIR (Passive Infrared) o Pasivo Infrarrojo, reaccionan sólo ante determinadas fuentes de energía tales como el calor del cuerpo humano o animales. Básicamente reciben la variación de las radiaciones infrarrojas del medio ambiente que cubre. Es llamado pasivo debido a que no emite radiaciones, sino que las recibe. Estos captan la presencia detectando la diferencia entre el calor emitido por el cuerpo humano y el espacio de alrededor.

Su componente principal son los sensores piroeléctrico. Se trata de un componente electrónico diseñado para detectar cambios en la radiación infrarroja recibida. Generalmente dentro de su encapsulado incorporan un transistor de efecto de campo que amplifica la señal eléctrica que genera cuando se produce dicha variación de radiación recibida.

La información infrarroja llega al sensor piroeléctrico a través de una lente de fressnell que divide el área protegida en sectores.

En la siguiente Figura se pueden ver los dos potenciómetros que permiten modificar el tiempo de refresco y la longitud de detección.

Figura 46: Sensor PIR.

4.4.2. Sensor de temperatura DS18B20

El sensor de temperatura DS18B20 es un sensor de coste reducido con un rango de temperatura de -55ºC a +125ºC y una precisión de +-0.5ºC en el rango de -10ºC a 85ºC.

Una de las ventajas del DS18B20 es que se comercializa tanto en un integrado TO-92 como en forma de sonda impermeable, lo que permite realizar mediciones de temperatura en líquidos y gases.

El DS18B20 emplea un bus de comunicación denominado 1-Wire. La principal ventaja del bus 1-Wire es que necesita un único conductor para realizar la comunicación (sin contar el conductor de tierra). Los dispositivos pueden ser alimentados directamente por la línea de datos, o mediante una línea adicional con una tensión de 3.0 a 5.5V.

Dentro del mismo bus 1-Wire se pueden instalar tantos sensores como deseemos. Además, el bus 1-Wire permite emplear cables más largos que otros sistemas antes de que se deteriore la comunicación.

Se necesita una resistencia de Pull-UP de 4.7kOhm entre Vcc y Vq, por lo que en la siguiente Figura vemos como se ha de cablear.

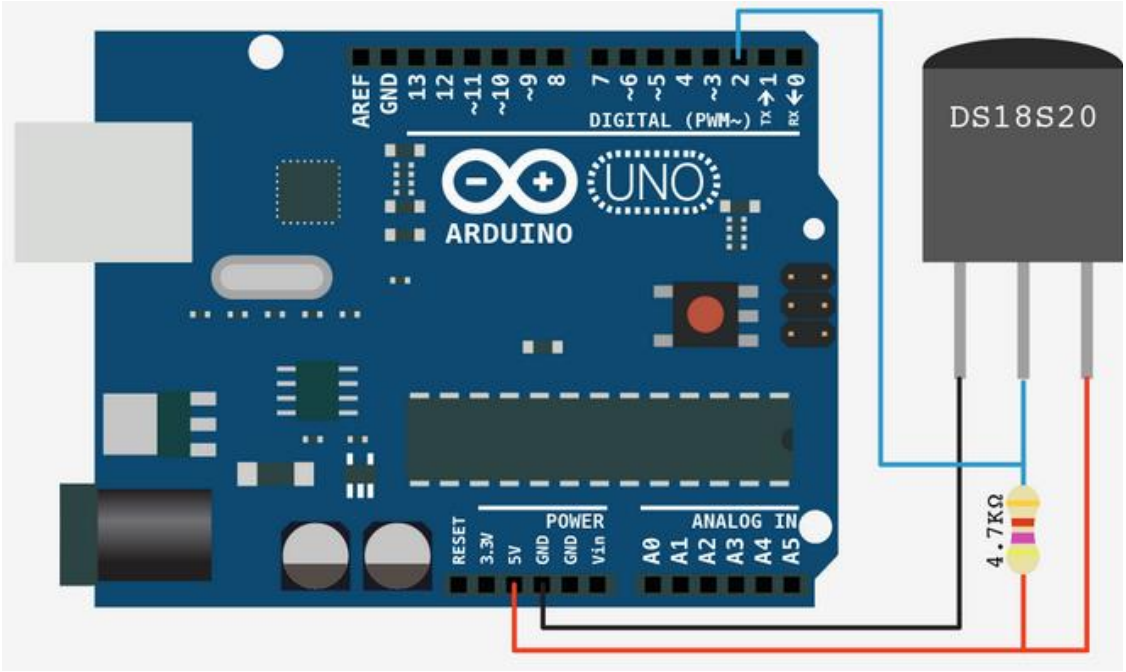


Figura 47: Sensor de temperatura DS18B20. (Fuente: www.programafacil.com/blog)

4.4.3. Sensor de temperatura y humedad DHT11

El DHT11 es un sensor de temperatura y humedad digital de bajo costo. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no hay pines de entrada analógica). Es bastante simple de usar, Estos sensores disponen de un procesador interno que realiza el proceso de medición, proporcionando la medición mediante una señal digital, por lo que resulta muy sencillo obtener la medición desde un microprocesador como Arduino. El único inconveniente de este sensor es que sólo se puede obtener nuevos datos una vez cada 2 segundos, así que las lecturas que se pueden realizar serán como mínimo cada 2 segundos.

En comparación con el DHT22, este sensor es menos preciso, menos exacto y funciona en un rango más pequeño de temperatura / humedad, pero su empaque es más pequeño y menos caro.

Características:

- Alimentación: 3Vdc ≤ Vcc ≤ 5Vdc
- Rango de medición de temperatura: 0 a 50 °C

- Precisión de medición de temperatura: ±2.0 °C .
- Resolución Temperatura: 0.1°C
- Rango de medición de humedad: 20% a 90% RH.
- Precisión de medición de humedad: 4% RH.
- Resolución Humedad: 1% RH
- Tiempo de muestreo: 1 seg.

En la **Figura 48** puede verse como ha de quedar cableado este sensor:

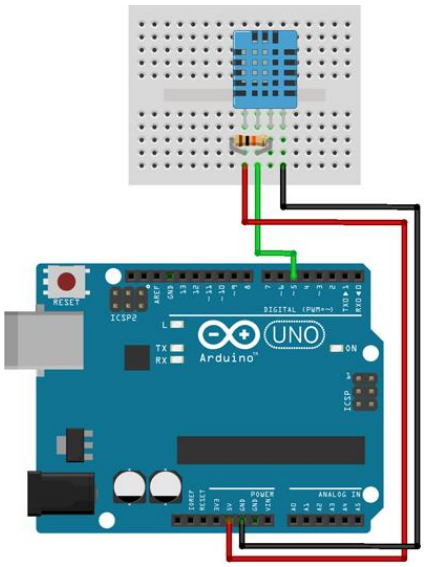


Figura 48: Sensor de temperatura DHT11. (Fuente: www.arduino.com)

4.4.4. Sensor de Vibración SW420

El módulo sensor de vibración SW-420 es un circuito que consta del sensor de vibración SW-420 y en conjunto con el OPAMP LM393 detectan si hay alguna vibración más allá del umbral fijado. La sensibilidad puede ser ajustada a través del potenciómetro integrado en el módulo.

El SW-420 es un sensor de vibración el cual está constituido interiormente por un resorte y un pequeño poste en su interior, por lo que cada vez que el sensor sea sometido a una vibración o golpe su salida se verá afectada.

Se utilizará este módulo para comprobar si alguna puerta o ventana está siendo forzada. A continuación, una imagen del sensor:

Figura 49: Sensor de Vibración SW-420. (Fuente: www.inven.es)

4.4.5. Sensor magnético para puertas y ventanas

El Sensor magnético para ventanas y puertas consta de un imán y un reedswitch (interruptor magnético). Este sensor funciona como un switch normalmente abierto, mientras hay campo magnético. Cuando la puerta o ventana se abre, el circuito eléctrico también se cierra y es posible detectar la apertura de la misma.

La conexión con Arduino se realiza mediante cables que vienen previamente instalados en el sensor y que habrá que unir con un cable más largo.

Este sensor nos proporciona una entrada digital al Arduino, por lo que su conexión se realizaría tal y como se ve en la siguiente figura.

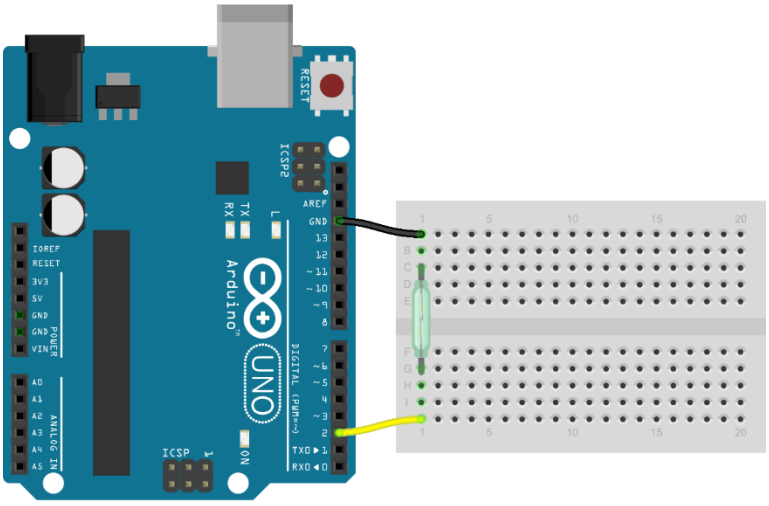


Figura 50: Sensor magnético para puertas y ventanas. (Fuente: www.arduinogeek.com)

4.4.6. Fotorresistencias LDR

Un fotorresistor, o LDR (light-dependent resistor) es un dispositivo cuya resistencia varía en función de la luz recibida. Podemos usar esta variación para medir, a través de las entradas analógicas, una estimación del nivel de la luz.

Un fotoresistor está formado por un semiconductor, típicamente sulfuro de cadmio (CdS). Al incidir la luz sobre él algunos de los fotones son absorbidos, provocando que electrones pasen a la banda de conducción y, por tanto, disminuyendo la resistencia del componente.

Por tanto, un fotoresistor disminuye su resistencia a medida que aumenta la luz sobre él. Los valores típicos son de 1 Mohm en total oscuridad, a 50-100 Ohm bajo luz brillante.

Por otro lado, la variación de la resistencia es relativamente lenta, de 20 a 100 ms en función del modelo. Esta lentitud hace que no sea posible registrar variaciones rápidas, como las producidas en fuentes de luz artificiales alimentadas por corriente alterna. Este comportamiento puede ser beneficioso, ya que dota al sensor de una gran estabilidad.

Se utilizarán LDRs para la regulación automática de la luz.

En la siguiente figura, se puede ver cómo quedaría conectado el LDR con el Arduino.

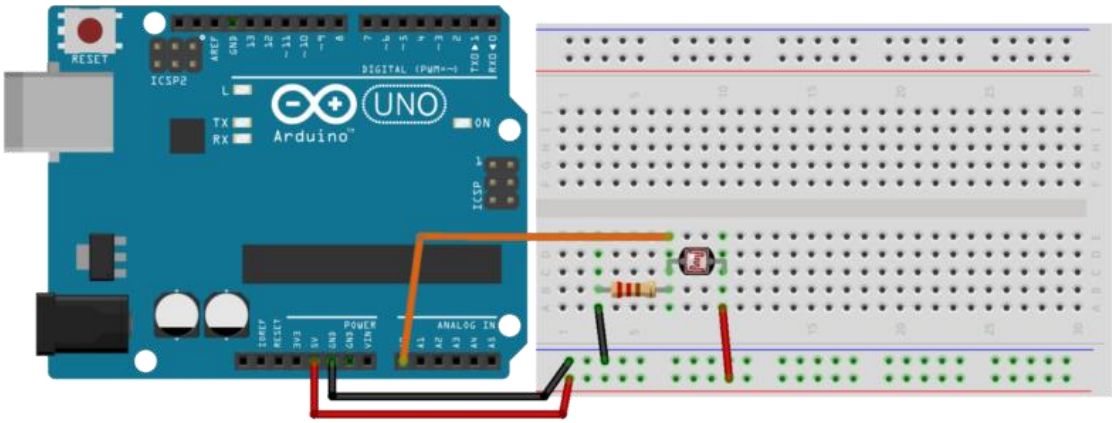


Figura 51: LDR y Arduino. (Fuente: www.Arduinoforum.com)

4.4.7. Actuadores y otros elementos

Por último, para videovigilancia, se utilizará una webcam USB que tomará grabaciones cuando el sistema de alarma se dispare. Las condiciones en las que el sistema de alarma se dispara pueden verse en este mismo apartado.

Para indicar el disparo de la alarma también se utilizarán sirenas de emergencia y avisos luminosos mediante Leds.

El sistema también cuenta con botones físicos para el rearme de la alarma.

CAPÍTULO 5.

DESARROLLO DEL DISEÑO DEL PROTOTIPO

5. Desarrollo del diseño del prototipo

En este apartado se explicará cómo se ha llevado a cabo la fase de diseño del producto. Viendo paso a paso cómo se ha de programar y cablear.

5.1. Conexión entre Cayenne y Raspberry/Arduino

Para la realización de este proyecto se ha utilizado la herramienta Online MyDevices Cayenne. En apartados anteriores se ha discutido el por qué se ha elegido esta herramienta y qué ventajas nos ofrece.

Esta herramienta es completamente gratuita, aunque existe una versión de pago para ampliar el número de dispositivos que se pueden conectar.

Lo primero que se debe hacer es crear una cuenta de MyDevices, permitiendo así la creación de un nuevo proyecto con Cayenne.

Con la cuenta creada deberemos de introducir la Raspberry/Arduino en la herramienta, para ello, lo primero que se ha de tener en cuenta es que deberemos de tener conexión a internet en el PC y en la placa a conectar.

Para conectar la **Raspberry Pi** se han de realizar los siguientes pasos:

- Instalación del sistema operativo Raspbian en la placa Raspberry
- Instalación de las librerías de Mydevices. Para ello se han de introducir las siguientes líneas de comando en el terminal de la Raspberry Pi:
    - wget https://cayenne.mydevices.com/dl/rpi_cjqccradod.sh
    - sudo bash rpi_cjqccradod.sh -v

La Raspberry cuenta con conexión a Internet por WiFi y por Ethernet, por lo que la placa debe estar conectada durante el proceso.

Una vez que se hayan realizado estos pasos, ya podremos utilizar la Raspberry dentro de la herramienta Cayenne.

Para conectar un **Arduino** se han de seguir los siguientes pasos:

- Primero se debe contar con un Shield (Ethernet o WiFi). Para este diseño se ha optado por el conjunto Arduino + WiFi Shield.

- Se ha de conectar el arduino al PC vía USB.
- Instalar Arduino IDE
- Añadir la "*Cayenne Library*" al Arduino IDE
- Seleccionar el modelo de Arduino.
- Cargar en el Arduino el Script proporcionado por MyDevices.
- Una vez cargado el Script en el Arduino, y verificado en el PC se podrá utilizar el Arduino con la herramienta Cayenne.

5.2. Desarrollo del diseño

Una vez que todos los módulos se han incluido dentro de la herramienta, se procederá a la conexión (Software y Hardware) de todos los sensores y actuadores.

Para conectar un sensor primero se debe ver el Pinout de la placa, para ello, Cayenne ofrece la posibilidad de ver todos los pines disponibles (viendo también cuál es su función).

En la **Figura 52,** podemos ver la GPIO de nuestra Raspberry.

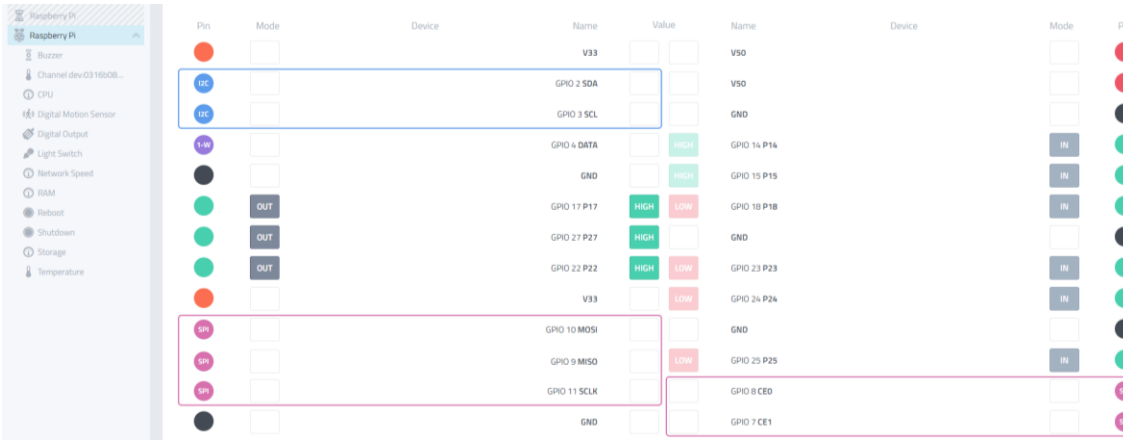


Figura 52: GPIO Raspberry.

Con este overview de los pines se pueden seleccionar los pines a los que se van a cablear los diferentes sensores.

En esta ventana también se pueden configurar los pines (si son entradas o salidas, lógica negada, etc…).

Una vez que se ha comprobado cual es la señal que vamos a cablear y cómo se quiere hacer, se debe cablear, tal y cómo se explica en el capítulo 4.

Una vez que el sensor queda conectado se debe seleccionar el pin al que se ha conectado el instrumento. A continuación, se debe hacer el gráfico a través de la herramienta Software. Para ello debemos seleccionar el icono que se mostrará en la pantalla, se mostrará una pequeña animación cuando se active, se configurarán también los rangos de alarma (aplicable sólo a las entradas analógicas), etc…

Con todos los sensores conectados a cada uno de los módulos se deben de introducir todos los valores de disparo. En el caso que se traten de entradas digitales, serán 0 ALARMA, 1 NO ALARMA.

Con las entradas analógicas se introducirán los valores de disparo (por ejemplo, 60º para el sensor de temperatura).

Con todos los disparos configurados a continuación debemos de establecer las causas y efectos de la alarma (es decir, en qué condiciones dispara). Para esto Cayenne ofrece la posibilidad de introducir la causa que origina un efecto determinado. Por lo que se han de configurar los disparos de la alarma tal y como aparecen en el Capítulo 4, **Figura 44** "Tabla de disparo automático de la alarma".

En la siguiente figura se puede ver cómo se crean las condiciones de disparo.

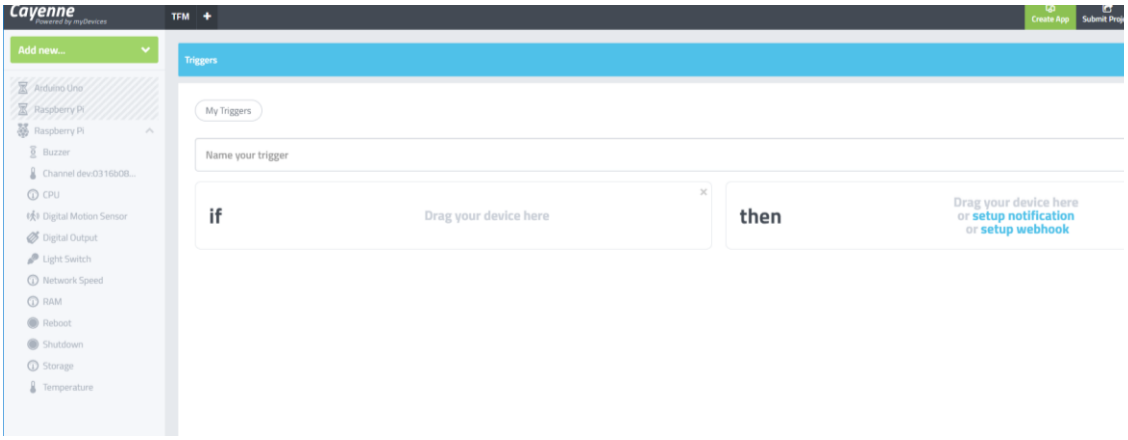


Figura 53: Creación de eventos en Cayenne.

En esta pantalla también se puede seleccionar la opción de que nos envíe un SMS o un Mail cuando se dispare la alarma. Para ello tan solo se debe incluir un número de teléfono y/o un email.

Las causas y los efectos pueden estar en cualquier dispositivo indistintamente. Esto es posible gracias a que todos los módulos están conectados en la nube, por

lo que la causa puede estar en un Arduino y el efecto puede estar en una Raspberry Pi.

Cuando se añadan todas las causas de disparo el sistema quedará completamente operativo. Por lo que tan solo se necesitará conexión a internet y alimentación a 5V para que funcionen.

5.3. Otros sistemas de alarma

Aquí se pretenden mostrar otros sistemas de alarma comparables a este con el fin de tener una referencia.

Blaupunkt SA 2500: Se trata del kit autoinstalable compuesto por una central con sirena incorporada, un sensor magnético de apertura de puertas, otro volumétrico que el propio usuario ubicará en el lugar de paso que considere oportuno y un mando a distancia para activarla o desactivarla. Este modelo requiere de una tarjeta SIM con la que informará al usuario mediante SMS de las posibles intrusiones y la configuración de la alarma se lleva a cabo mediante una aplicación gratuita para iOS y Android.

Una de las principales ventajas del kit, que puede adquirirse por 259 euros, es que los sensores vienen ya configurados con lo que basta con colocarlos en el domicilio. Se pueden adquirir sensores adicionales para proteger desde la central más zonas de la vivienda.

Figura 54: Blaupunkt. Fuente: www.blaupunkt.com)

2. G5 Touch: Se trata de una alarma muy similar a la anterior, aunque en este caso es de fabricación china, y como puede suponer, notablemente más económica. Cuenta con un sensor magnético para la puerta, otro volumétrico, dos mandos a distancia y un par de llaves sin contacto para la activación o desactivación de la alarma.

Al igual que el modelo anterior, requiere de una tarjeta SIM para las notificaciones, un gasto que hay que tener en cuenta, aunque con la oferta actual es muy reducido. El sistema G5 puede gestionarse igualmente mediante una aplicación para Android y iPhone, y puede adquirirse por un precio de partida de 195 euros, aunque posteriormente se pueden ir añadiendo más sensores al conjunto.

Figura 55: G5 Touch. (Fuente: www.todoelectronica.com)

3. Wattio Seguridad: Se trata de un kit de seguridad muy sofisticado que cuenta además con la ventaja de no necesitar de una tarjeta SIM, ya que se conecta al wifi de la casa a través de un gateway. En realidad, este kit no es más que una extensión de su sistema domótico modular, mediante el cual el usuario puede regular la iluminación y calefacción del hogar. No se trata de una alarma en sí, puesto que carece de sirena, pero sí notifica al propietario de la vivienda de todo lo que sucede en la misma mediante una aplicación Android o iOS.

A este sistema se le pueden ir añadiendo componentes y programar acciones, como encender la luz de forma automática cada vez que se acceda al domicilio. El pack básico compuesto por gateway y sensores se comercializa por menos de 200 euros, y con la ventaja de no necesitar de una tarjeta SIM para las notificaciones.

Figura 56: Wattio Seguridad. (Fuente: www.wattio.com)

5.4. Ventajas y Desventajas con un sistema de Alarma convencional

En este apartado se discutirán las ventajas y desventajas del sistema propuesto ante un sistema de alarma convencional.

Ventajas:

- Precio Reducido.
- Acceso desde Internet y desde app de Android/IOS.
- Sistema ampliable (añadiendo nuevos sensores/actuadores).
- Programación sencilla.
- Sensores y actuadores fácilmente sustituibles.
- Programación de forma remota

Desventajas:

- Es necesaria la conexión a internet.
- Se necesitan unos conocimientos medios para su mantenimiento (reemplazar cualquier sensor o programar sensores nuevos).
- No dispone de asistencia telefónica ni aviso a policía/empresa de seguridad.

CAPÍTULO 6.

RESULTADOS FINALES

6. Conclusiones y análisis de resultados.

Resulta esencial realizar un análisis de los resultados obtenidos, hacer un recuento de los objetivos cumplidos a lo largo de este proyecto y obtener conclusiones que nos permitan llevar a cabo una valoración global del trabajo.

6.1. Conclusión

Cuando se ha llegado a este punto del proyecto es momento de realizar un balance de los objetivos marcados al inicio de este TFM.

- Se ha realizado un estudio de las principales plataformas IoT software que existen en la actualidad, con sus características, ventajas y desventajas.

- Se ha realizado un estudio de los principales dispositivos hardware IoT.

- Se ha estudiado la validez de la arquitectura planteada para el prototipo desarrollado en este TFM.

- Se ha realizado un plan de negocio para el producto que se ha propuesto en este TFM, siguiendo la metodología Lean Startup.

6.2. Futuros trabajos

Como parte de los futuros trabajos que se pueden realizar a raíz del desarrollo de este TFM se encuentran los siguientes:

- Como trabajo futuro se podría hacer un estudio sobre la aceptación de un producto mínimo viable (mediante land pages, encuestas a usuarios, etc…).

- También se puede repetir el estudio realizado en este TFM utilizando una plataforma software y un dispositivo hardware distintos (haciendo una comparación con las herramientas utilizadas en este TFM).

- Investigar sobre distintas metodologías que permitan agilizar el desarrollo de un producto IoT.

CAPÍTULO 7.

BIBLIOGRAFÍA

7. Bibliografía.

Adafruit. (2019). *Adafruit IO*. Recuperado el 12 de Diciembre de 2018, de https://io.adafruit.com/

Altair. (2019). *Altair Smartworks*. Recuperado el 13 de Diciembre de 2018, de https://www.altairsmartworks.com/smartcore-overview

Apache Spark. (2019). *Apache Spark*. Recuperado el 6 de Diciembre de 2018, de http://spark.apache.org/faq.html

Arduino. (2019). *Arduino*. Recuperado el 16 de Diciembre de 2018, de https://www.arduino.cc/en/Main/FAQ#toc2

AWS IoT. (2019). *Amazon Web Services*. Recuperado el 4 de Diciembre de 2018, de https://docs.aws.amazon.com/es_es/iot/latest/developerguide/iot-security-identity.html

BBVAOpen4U. (21 de Marzo de 2016). *BBVA API Market*. Recuperado el 16 de Diciembre de 2018, de https://bbvaopen4u.com/es/actualidad/apis-para-el-internet-de-las-cosas-thingspeak-pachube-y-fitbit

Cayenne. (2019). *My devices*. Recuperado el 16 de Diciembre de 2018, de https://mydevices.com/cayenne/docs_stage/intro/

Electric Imp. (2019). *Electric Imp*. Recuperado el 6 de Diciembre de 2018, de https://www.electricimp.com/platform/how-it-works/

Google Cloud. (2019). *Google Cloud IoT*. Recuperado el 5 de Diciembre de 2018, de https://cloud.google.com/solutions/iot/

Gracia, L. (21 de Agosto de 2012). *Un poco de Java Y+*. Recuperado el 25 de Diciembre de 2018, de https://unpocodejava.com/2012/08/21/que-es-waspmote/

Macchina. (2019). *Macchina IoT*. Recuperado el 17 de Diciembre de 2018, de https://macchina.io/sdk.html

Microsoft Azure. (2019). *Microsoft*. Recuperado el 13 de Diciembre de 2019, de https://azure.microsoft.com/es-es/services/iot-hub/

Oracle. (2019). *Oracle*. Recuperado el 16 de Diciembre de 2018, de https://www.oracle.com/es/solutions/internet-of-things/

Raspberry. (2019). *Raspberry*. Recuperado el 20 de Diciembre de 2018, de https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/

Samsung ARTIK. (12 de Enero de 2016). *Samsung ARTIK.* Recuperado el 4 de Diciembre de 2018, de http://static.artik.io/files/Samsung_ARTIK_Overview.pdf

SensorFlare. (2019). *SensorFlare*. Recuperado el 7 de Diciembre de 2018, de https://sensorflare.docs.apiary.io/#reference/location-operations/update-distance-to-beacon

Telefonica. (2019). Recuperado el 15 de Diciembre de 2018, de Telefonica Internet of Things: https://iot.telefonica.com/es/thinking-things

Ubidots. (2019). *Ubidots*. Recuperado el 15 de Diciembre de 2018, de https://ubidots.com/

Watson Internet of Things. (2019). *Watson Internet of Things*. Recuperado el 13 de Diciembre de 2018, de https://www.ibm.com/internet-of-things

Xively. (2019). *Xively*. Recuperado el 18 de Diciembre de 2018, de https://xively.com/

Zatar IoT Platform. (Enero de 2014). *Zatar IoT Platform*. Recuperado el 14 de Diciembre de 2018, de https://www.iotone.com/software/zatar-iot-platform/s69

Agradecimientos.

Me gustaría mostrar mi agradecimiento a todas aquellas personas que de una forma u otra han hecho posible la realización de este trabajo gracias a su ayuda y cariño.