

**UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO**  
**FACULTAD DE INGENIERIA MECANICA ELECTRICA,**  
**ELECTRONICA Y SISTEMAS**  
**ESCUELA PROFESIONAL DE INGENIERIA ELECTRONICA**



**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL**  
**MONITOREO DE CULTIVOS NATIVOS UTILIZANDO**  
**INTERNET DEL TODO Y REDES FOG”**

**TESIS**

**PRESENTADO POR:**

**GIANMARCO PALACIOS FRISANCHO**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO ELECTRÓNICO**

**PUNO – PERÚ**

**2017**

**UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO**  
**FACULTAD DE INGENIERIA MÉCANICA ELÉCTRICA, ELECTRÓNICA Y SISTEMAS**  
**ESCUELA PROFESIONAL DE INGENIERIA ELECTRÓNICA**

“DISEÑO E IMPLEMENTACION DE UN SISTEMA PARA EL MONITOREO DE  
CULTIVOS NATIVOS UTILIZANDO INTERNET DEL TODO Y REDES FOG”

TESIS PRESENTADA POR:

**GIANMARCO PALACIOS FRISANCHO**

PARA OPTAR EL TITULO PROFESIONAL DE:

**INGENIERO ELECTRÓNICO**

FECHA DE SUSTENTACION: 07/12/17



**APROBADO POR EL JURADO REVISOR CONFORMADO POR:**


**PRESIDENTE:**

  
Mg. TEOBALDO RAUL BASURCO CHAMBILLA

**PRIMER MIEMBRO:**

  
Dr. EUDES RIGOBERTO APAZA ESTAÑO

**SEGUNDO MIEMBRO:**

  
Mg. LUIS ENRIQUE BACA WIESE

**DIRECTOR / ASESOR:**

  
Ing. FERDINAND EDGARDO PINEDA ANCCO

**PUNO – PERU**

**2017**

Área : Telecomunicaciones y redes de datos

Tema : Internet del Todo y Fog Networks

**DEDICATORIA**

El presente proyecto va dedicado principalmente a mi familia en cielo y tierra que desde pequeño me inculcaron los valores necesario para formarme en el ser que soy ahora, y que con todo su afecto supieron guiarme hasta llegar a este punto donde gracias a ellos puedo cumplir un objetivo más, así también a todas los seres que demostraron cariño hacia mi persona y me impulsaron a seguir adelante frente a cualquier adversidad.

Gianmarco Palacios Frisancho

### **AGRADECIMIENTOS**

Agradezco profunda y enormemente a todas las personas que me brindaron una mano en todo momento, y que fueron figuras importantes en mi desarrollo personal, ya que de ellas aprendí mucho, a todas aquellas personas que me compartieron un poco de su conocimiento en todo ámbito pues ahora puedo compartir con más personas un poco de lo aprendido, a todas las personas que fueron parte de mi desarrollo profesional y aunque quizás ya no estén presente fueron personas importantes que me permitieron caminar con firmeza en este camino, a todas las personas que me escucharon y compartieron ideas locas de la vida, a todas las personas que hacen un poco mejor este mundo y nunca se dan por vencidas, y a todas las personas con las que algún momento compartieron un segundo de su vida conmigo.

Gianmarco Palacios Frisancho

## INDICE GENERAL

RESUMEN .....	18
ABSTRACT .....	19
CAPITULO I .....	20
I. INTRODUCCION .....	20
1.1. Justificación del proyecto .....	20
1.2. Antecedentes del proyecto.....	21
1.3. Hipótesis general .....	25
1.4. Hipótesis específicas.....	25
1.5. Objetivo general .....	26
1.6. Objetivos específicos.....	26
CAPITULO II.....	27
II. REVISION DE LITERATURA .....	27
2.1. Cultivos nativos .....	27
2.1.1. Los cultivos nativos.....	27
2.1.2. Agricultura en el Perú .....	28
2.1.3. Cultivos nativos del Perú .....	29
2.2. Cultivo de la quinua.....	30
2.2.1. Parámetros de cultivo.....	30
a) Radiación solar .....	30
b) Temperatura .....	31

c) Precipitación .....	32
d) Humedad relativa .....	34
e) Vientos .....	35
f) Temperatura del suelo.....	36
2.2.2. Parámetros de cultivo de la quinua.....	37
2.3. Problemas y cuidados en el cultivo de la quinua.....	39
2.3.1. Problemas en el cultivo de la quinua.....	39
a) Insectos plaga del cultivo de la quinua.....	40
b) Aves plaga .....	40
c) Humeda .....	41
d) Temperatura .....	41
e) Heladas .....	42
2.3.2 Cuidados en el cultivo de la quinua.....	43
a) Riego .....	43
b) Control de plagas.....	44
c) Control de enfermedades.....	44
d) Control de heladas.....	44
i. Riego .....	44
ii. Calentamiento del aire y las plantas .....	46
2.4 Sistemas de monitoreo.....	47
2.4.1 Sistema de monitoreo de cultivos de hoja de coca en el Perú .....	47

2.4.2 Aplicaciones móviles .....	48
2.4.3 Beneficios de la agricultura de precisión.....	49
2.5 Internet del Todo .....	49
2.5.1 Evolución de Internet .....	50
a) Fase 1 .....	50
b) Fase 2 .....	50
c) Fase 3 .....	51
d) Fase 4 .....	51
2.5.2. Transición a IdT.....	53
2.5.3. Pilares de IdT .....	53
Objetos, Datos, Personas Y Procesos .....	54
2.5.3.1. Objetos.....	54
Conexiones informáticas comunes .....	56
a) Dispositivos comunes.....	56
b) Conexión de dispositivos .....	56
c) Conexión a objetos informáticos no tradicionales.....	57
d) Sensores .....	57
e) RFID .....	58
f) Controladores.....	60
2.5.3.2 Datos.....	61
a)Administración de datos.....	61

b) Datos estructurados .....	62
c) Datos no estructurados .....	62
d) Almacenamiento de datos .....	63
e) Transporte de datos.....	64
i. Proveedores de servicios de Internet .....	64
ii. Direccionamiento IP .....	65
iii. Paquetes IP .....	65
iv. Administración de direcciones IP .....	66
v. Datos en movimiento.....	67
vi. Administración de datos masivos .....	67
vii. Análisis de datos masivos .....	68
viii. Virtualización.....	69
ix. Computación en la nube .....	70
2.5.3.3 Personas .....	71
2.5.3.4 Procesos .....	71
2.5.4 Conectando los pilares .....	71
2.5.4.1 Conexiones M2M .....	71
2.5.4.2 Conexiones M2P .....	72
2.5.4.3 Conexiones P2P.....	72
2.5.4.4 Interacciones de IdT .....	73
2.5.5 La red.....	75
a) Componentes de las redes .....	75



b) Dispositivos finales .....	76
c) Dispositivos de red intermediarios .....	77
d) Medios de red .....	78
e) Tipos de redes .....	78
f) Internet .....	79
g) La red convergente .....	80
h) Programación .....	80
i) Definición .....	80
j) Tipos de programas .....	81
k) Transición a IdT .....	82
l) Creación de un modelo para una solución mediante IdT .....	82
2.6 Redes Fog .....	83
2.6.1 Concepto .....	83
2.6.2. Algunos inconvenientes del Fog Computing .....	84
2.6.3. Controladores en la niebla .....	85
CAPITULO III .....	86
MATERIALES Y METODOS .....	86
3.1 El Electron .....	86
3.1.1 Configurando el Electron .....	86
3.1.2 Conectando vía USB .....	89
3.1.3 Instalando Node.Js .....	89

3.1.4 Instalando el driver de Particle.....	91
3.1.5 Conectando nuestro dispositivo .....	92
3.1.6 Modos del dispositivo .....	93
a) Modos normales .....	94
b) Firmware Reset .....	95
c) Factory Reset.....	95
d) Tinker .....	95
3.1.7 Flashear Apps con Particle Build .....	96
3.1.7.1 Web IDE.....	96
3.1.8 Aplicaciones y librerías – Particle .....	98
a) Agregando archivos a las aplicaciones.....	98
b) Usando librerías .....	98
c) El Firmware .....	100
3.1.9 El servicio de facturación de Particle.....	100
3.1.9.1 Periodo de facturación .....	101
3.1.10 Optimizando el uso de datos .....	101
a) Cambiado a UDP desde TCP .....	101
b) Reducción de la frecuencia de ping.....	102
c) Menos handshakes y sesiones más largas .....	102
d) La estructura de la transmisión de datos .....	103
e) ¿Qué consumen los datos?.....	103

f) Formas de reducir el uso de datos .....	105
3.2 Creación del modelo como solución IdT.....	108
3.3. Diagrama de flujo del modelo .....	108
3.4 Adquisición de los datos.....	109
3.5. Usando las redes en la niebla (Fog Network).....	110
3.6 Inicio del Internet del Todo .....	112
3.6 ThingSpeak.....	115
3.7 Webhooks .....	117
3.8 Matlab Visualizations .....	118
3.9 Pushbullet .....	119
3.10 Integración con servicios adicionales .....	121
3.11 Resumen del proceso de diseño e implementación .....	127
CAPITULO IV .....	129
RESULTADOS Y DISCUSIÓN .....	129
4.1. El Electron como medio de transmisión.....	129
4.2 Ventajas del Electron frente a otros dispositivos .....	131
4.3 Inconvenientes en el uso del Electron .....	132
4.4. Un sistema de monitoreo económico.....	132
4.5. La agricultura de precisión .....	133
4.6. Fog Networks .....	139
4.7. Internet del Todo como solución y sus interacciones.....	139
4.8. Convergencia.....	141

CONCLUSIONES.....	142
RECOMENDACIONES .....	144
REFERENCIAS .....	145
ANEXOS .....	144

## INDICE DE FIGURAS

Figura 1: Distribución de Internet mundial .....	52
Figura 2: Sensor de oxígeno para autos.....	58
Figura 3: Conexión RFID.....	59
Figura 4: Función de los controladores. ....	60
Figura 5: Red troncal de Internet.....	64
Figura 6: Dispositivos conectados por año.....	67
Figura 7: Ejemplo de virtualización. ....	70
Figura 8: Integración de las conexiones en IdT.....	73
Figura 9: Componentes de una red.....	76
Figura 10: Infraestructura LAN-WAN.....	79
Figura 11: Modelo de solución basado en IdT.....	83
Figura 12: Login en Particle .....	86
Figura 13: Activación del Electron.....	87
Figura 14: Set up del Electron.....	88
Figura 15: Número de ICCID.....	88
Figura 16: Instalación de Node.js. paso 1.....	90
Figura 17: Instalación de Node.js. paso 2.....	90
Figura 18: Instalación de Node.js. paso 3.....	91
Figura 19: Instalación del driver del Electron.....	91
Figura 20: Instalación de la CLI de Particle.....	92

Figura 21: Comandos de Particle..... 93

Figura 22: Login de cuenta mediante la CLI. .... 93

Figura 23: IDE de Particle. .... 96

Figura 24: Metadatos de un código del Electron. .... 99

Figura 25: Modelo basado en IdT para un cultivo nativo. .... 108

Figura 26: Diagrama de flujo del modelo de monitoreo. .... 109

Figura 27: Publicación de datos en la consola de Particle..... 114

Figura 28: Creación de un canal en ThingSpeak..... 116

Figura 29: API Keys..... 116

Figura 30: Programación del API key y channel ID. .... 117

Figura 31: Visualización de datos en ThingSpeak. .... 117

Figura 32: Visualización de la humedad frente la temperatura ..... 119  
mediante programación Matlab. .... 119

Figura 33: Notificación push en un celular ..... 121  
mediante Pushbullet..... 121

Figura 34: Creación de un topic en Google Cloud. .... 123

Figura 35: Creación de un SUB/PUB en Google Cloud ..... 123  
mediante API. .... 123

Figura 36: Permisos en Google Cloud..... 123

Figura 37: Creación de la integración en Particle..... 124

Figura 38: Creación de la suscripción mediante el CLI de Google Cloud. .... 124

Figura 39: Obtención de los datos visualizados mediante el CLI de Google Cloud. ... 125

Figura 40: Cadena de transporte de datos hacia .....	125
el Datastore de Google Cloud.....	125
Figura 41: Configuración y permisos en el topic de Google Cloud. ....	126
Figura 42: Clonación del repo de Google Cloud desde HitGub.....	127
Figura 43: Datastore creado en Google Cloud. ....	127
Figura 44: Uso de la cantidad de datos del Electron. ....	130
Figura 45: Integraciones realizadas en Particle. ....	131
Figura 46: Comunicación I2C Electron-Arduino/envió de datos del Arduino.....	134
Figura 47: Comunicación I2C Electron-Arduino .....	134
recepción el Electron. ....	134
Figura 48: Medición del pH del agua. ....	134
Figura 49: Medición de la humedad del suelo.....	135
Figura 50: Medición de la temperatura del suelo. ....	135
Figura 51: Medición de la temperatura ambiente.....	136
Figura 52: Medición de la humedad ambiente. ....	136
Figura 53: Plot de la temperatura ambiente.....	137
Figura 54: Plot de la humedad ambiente. ....	137
Figura 55: Comparación entre humedad y temperatura ambiente.....	138
Figura 56: Histograma de la temperatura ambiente.....	138
Figura 57: Aviso de detección de movimiento.....	138

**INDICE DE TABLAS**

Tabla n° 1: Parámetros de calidad para el cultivo de la quinua.....	38
Tabla n° 2: Porcentaje de pérdidas en el cultivo de quinua según el motivo. ....	42
Tabla n° 3: Consumo calórico en el método del riego. ....	46
Tabla n° 4: Atributos de las interacciones de IdT. ....	74
Tabla n° 5: Modos de funcionamiento del Electron.....	94
Tabla n°6: Presupuesto mínimo para la implementación del sistema de monitoreo....	133
Tabla n° 7: Pilares e interacciones del modelo de monitoreo para cultivos.....	140



**INDICE DE ANEXOS**

Anexo A: componentes del sistema.....	148
Anexo A1: Electron (Particle) .....	148
Anexo A2: Arduino (Mega) .....	148
Anexo A3: Arduino (Nano).....	149
Anexo A4: Sensor humedad-temperatura (suelo) .....	149
Anexo A5: Sensor de Ph (pH electrode) .....	150
Anexo A6: Modulo del sensor de pH .....	150
Anexo A7: Sensor de humedad-temperatura (ambiente – DHT22) .....	151
Anexo A8: Sensor de movimiento (PIR).....	151
Anexo A9: Diagrama de conexiones del circuito .....	152
Anexo B: Programación de las funciones.....	153
Anexo B1: programación de la comunicación I2C (Electron) .....	153
Anexo B2: programación de la comunicación I2C (Arduino) .....	154
Anexo B3: programación PH (Arduino) .....	154
Anexo B4: Programación humedad-temperatura del suelo (Electron).....	155
Anexo B5: Programación humedad-temperatura ambiente (Electron) .....	157
Anexo B6: Programacion detección de movimiento (Electron).....	159
Anexo B7: Programación control por internet .....	160

**INDICE DE ACRÓNIMOS**

API (Pág. 97): Application Programming Interfaces / Interfaces de programación de aplicaciones

IdC (Pág. 20): Internet de las Cosas

IdT (Pág.20): Internet del Todo

IDE (Pág. 96): Integrated Development Environment / Entorno de desarrollo integrado

CLI (Pág.89): Command line interface / Interfaz de línea de comandos

## RESUMEN

Nuestro país uno de los países más reconocidos por su agricultura, llegando a estar entre los primeros puestos en varios productos de exportación, sin embargo no está libre de pérdidas, por diversos aspectos, pudiendo perderse cultivos antes, durante y después de la cosecha así como también se puede llegar a perder cultivos completos por la falta de prevención y estimación de posibles amenazas.

Junto a ello viviendo en una época de auge tecnológico, en el capítulo I veremos cómo podremos hacer posible esta difícil unión de estos dos aspectos: tecnología y naturaleza, sin alterar el uno al otro y mejorar la productividad en el campo agrícola.

En el capítulo II se desarrollara los conceptos generales sobre la quinua y de manera más amplia y profundo el de Internet del Todo, que es el nuevo concepto revolucionario de las telecomunicaciones, ya que nos dará ahora una nueva visión del futuro, un futuro donde podremos comunicar los campos con las personas sin la necesidad de estar presentes, poder monitorizar de manera remota y dejar que el mismo campo tome sus decisiones, poder prever posibles atentados en contra de un cultivo y lograr un mejor producto final.

Para lo cual en el capítulo III se desarrollara el uso de la plataforma de desarrollo de la empresa Particle, denominado Electron, un dispositivo electrónico que nos permitirá comunicar los sensores, de donde recolectaremos los datos para luego ser procesados y tenerlos en nuestra Red Fog, para luego poder ser visualizados de manera gráfica por Internet en cualquier dispositivo que se encuentre conectado, finalmente en el capítulo IV veremos los resultados de estas interacciones.

**Palabras claves:** Internet del todo, Internet de las cosas, Redes FOG, Computación en la nube y Electron.

## ABSTRACT

Our country is one of the most recognized countries for its agriculture, reaching the top positions in several export products, however it is not free of losses, due to several aspects, being able to lose crops before, during and after the harvest as well as it is also possible to lose complete crops due to the lack of prevention and estimation of possible threats,

Along with it living in a time of technological boom, in chapter I we will see how we can make possible this difficult union of these two aspects: technology and nature, without altering one another and improving productivity in the agricultural field.

Chapter II will develop the general concepts about quinoa and more broadly and profoundly the Internet of Everything, which is the new revolutionary concept of telecommunications, since it will now give us a new vision of the future, a future where we can communicate fields with people without the need to be present, to monitor remotely and let the same field make its decisions, to be able to foresee possible attacks against a crop and achieve a better final product.

For which in Chapter III the use of the development platform of the company Particle, called Electron, will be developed, an electronic device that will allow us to communicate the sensors, from where we will collect the data and then be processed and have them in our Fog Network, then to be visualized graphically by Internet in any device that is connected, finally in chapter IV we will see the results of these interactions.

**Keywords:** Internet of Everything, Internet of Things, Fog Networks, Cloud computing and Electron.

## CAPITULO I

### INTRODUCCION

#### 1.1. Justificación del proyecto

En la actualidad a pesar del gran desarrollo en el campo agrícola en nuestro país, aun somos vulnerables a grandes pérdidas de cultivos (antes, durante y después de su cosecha), puesto que siendo un país donde su mayor eje económico está basado en el sector agropecuario, es necesario plantear nuevas soluciones para afianzarnos en esa posición y ofrecer productos de mejor calidad y solucionar problemas que afectan los cultivos.

He aquí donde entra un gran elemento del siglo XXI llamado “tecnología”, a pesar de que aún existe una gran brecha entre la tecnología y la naturaleza, debido a la naturaleza antagónica que existen entre ellos, es posible reducir esa brecha y así ofrecer soluciones nuevas y amigables que hagan posible una gran mejora en el sector agrícola.

Es aquí donde entran los dos grandes conceptos que se tomaran en el presente proyecto:

- IdC (Internet de las Cosas)
- IdT (Internet del Todo)

Estos nuevos conceptos harán posible que la interacción naturaleza-tecnología se desarrolle de una mejor manera, en el presente proyecto conectaremos: objetos, datos, procesos y personas, para lograr una interacción del todo y solucionar el problema planteado, para lo cual se hará un monitoreo de los procesos de cultivos, donde la mano del agricultor no intervenga directamente, y pueda permitir tomar decisiones o tomarlas el, logrando así un procesamiento de datos que nos permitirán dar una estimación a futuro de los factores principales de los cultivos, para tomar mejores decisiones y poder prever posibles pérdidas.

Todo este proceso se hará de la mano con las nuevas tendencias de procesamiento de datos y estamos hablando de la computación en la nube (Cloud Computing), que no solo nos facilitará todo el procesamiento, sino que también nos ahorrará muchos costos en la instalación del equipamiento.

Mediante todos estos procesos será posible insertar la tecnología al campo sin necesidad de ser intrusivos, reduciremos perdidas mediante el monitoreo de los factores climáticos, temperatura, humedad, y mejoraremos la calidad del producto haciendo que los procesos de cuidados sean aplicados en los momentos correctos, logrando así un gran ahorro y una mejora en el producto final.

## **1.2. Antecedentes del proyecto**

El presente proyecto, no cuenta con antecedentes directos, sin embargo, de manera simultánea varias empresas están desarrollando modelos de sistemas basados en Internet de las Cosas e Internet del Todo, como lo hace la empresa mcelectronics con su prototipo ERMA, basado en comunicaciones LoRa, así también podemos tomar como

antecedentes, otros sistemas de monitoreo y análisis de datos que no están basados en IdC o IdT, pero que sirven como base para poder aplicar con la tecnología actual.

En la actualidad, más del 99% de los objetos que se encuentran en el mundo físico aún no están conectados a Internet, sin embargo, el fenómeno llamado “Internet del Todo” activará todo lo que pueda imaginarse, para el año 2020, habrá 50 000 millones de objetos inteligentes conectados a internet, mediante el uso de micro sensores en la red, los objetos cotidianos se conectaran y se volverán inteligentes, Internet del Todo conecta el mundo físico a Internet.

En muy poco tiempo, Internet modificó radicalmente la forma en que trabajamos, vivimos, jugamos y aprendemos. Sin embargo, esto recién es el comienzo. Mediante tecnologías existentes y nuevas, conectamos el mundo físico a Internet. La transición de Internet a Internet de todo se da cuando se conecta aquello que no está conectado.

En 2012, la cantidad de dispositivos conectados a Internet excedió la cantidad de habitantes en la Tierra.

Con este aumento en el uso de la tecnología, actualmente, las personas intercambian más información, ideas y opiniones que antes. Internet está cambiando la forma en que nos comunicamos, colaboramos y aprendemos. Esta tecnología modificó la manera de interactuar con otros y con los productos que utilizamos.

Bogotá, abril 2016. Schneider Electric, especialista global en gestión de energía y automatización, anunció hoy que está conectando más de 4.000 estaciones climáticas en áreas rurales para proveer una visión más holística de los patrones climáticos del clima rural a lo largo de Estado Unidos. Al apalancarse en la plataforma WeatherSentry capacitada para IoT y Big Data, la empresa presenta una previsión del clima más precisa

que aumenta la eficiencia, la rentabilidad y los resultados de la industria agrícola de Estados Unidos, lo que incluye a granjeros, agricultores y propietarios de la tierra.

Los sensores especializados de WeatherSentry capturan el clima a nivel del suelo y las condiciones de la tierra que se usarán para crear previsiones confiables y localizadas sobre la temperatura y las precipitaciones. Esta plataforma de Schneider Electric genera más datos del área rural agrícola que cualquier otro proveedor.

“A pesar de los abundantes avances tecnológicos logrados en la agricultura en el siglo pasado, el clima aún es una variable de alto costo y alto riesgo que impacta todos los rincones de la industria”, dijo Ron Sznaide, Vicepresidente de servicios cloud de Schneider Electric. “Al eliminar las suposiciones de los eventos climáticos se logra que los granjeros, los agricultores y los propietarios de la tierra tomen mejores decisiones operacionales y financieras que contribuyan directamente a la sostenibilidad de la operación y en últimas, a su salud.

Al intensificarse el impacto del cambio climático, los eventos relacionados con este aspecto afectan la productividad y la salud de las cosechas. El sistema de alertas patentado, basado en el Sistema de Información Geográfica (GIS), de WeatherSentry, provee datos en tiempo real para permitir a los granjeros la planificación de sus cosechas, la optimización del uso del suelo y el agua y priorizar la actividad basada en pronósticos de 15 días, Con el objetivo de reducir el desperdicio, incrementar la seguridad y aumentar la eficiencia.

“Reconocemos la importancia de las tecnologías IoT y Big Data en la mitigación del cambio climático y en resolver uno de los más críticos desafíos para los granjeros de todo el mundo”, dijo Sznaider. “El Internet de las Cosas revolucionará cómo nos



aproximamos a la producción sostenible de alimentos y nos emociona estar en la avanzada de la oferta y tecnologías de precisión que ayuden a suplir esta necesidad mundial”.

Está demostrado que WeatherSentry es entre 40% y 63% más confiable que otras fuentes sobre el clima. Schneider Electric en la actualidad provee servicios climáticos a más de 100.000 usuarios de la industria agrícola en todo el mundo. La empresa desarrolla de forma activa soluciones que midan el impacto del clima en la agricultura y explora nuevos modelos de negocios para desplegar estas soluciones en todo el planeta.

En nuestro país, el término aún no está arraigada y las tecnologías aún no se han adaptado a Internet del Todo, así que en este aspecto aún nos falta mucho por invertir, sin embargo, ya se está tomando en cuenta para los próximos años, como dice la siguiente nota del diario Gestión:

El Internet de las cosas es un concepto tal vez no muy común entre los peruanos, pero tiene un potencial de desarrollo muy grande teniendo en cuenta los avances del Internet, y según José Carlos Escudero, gerente de M2M y Movilidad del segmento Grandes Empresas de Telefónica, se requiere de mayor inversión para una mayor expansión del servicio.

“El desarrollo del Internet de las cosas pasa por inversión en equipamiento, en infraestructura y en cobertura porque hoy día cuando una red es intensiva en transmisión de datos, que es lo que hace el Internet de las cosas, eso ocupa mucha más ancho de banda y capacidad de la red”, explicó Telefónica.

Indicó que el potencial para desarrollar el Internet en el Perú es enorme si se tiene en cuenta que existen 31.2 millones de celulares, lo que implica una penetración de 100% en cuanto al número de aparatos, de los cuales al fin de año que entre 50% y 60% tendrán conexión a Internet.

Viendo el panorama actual de Internet de las Cosas e Internet del Todo en nuestro país, y sabiendo que aún está entrando a nuestra coyuntura, empezar a crear soluciones y aplicaciones como lo es en la agricultura, estaremos dando un gran paso para el desarrollo de la tecnología y el Internet del futuro y dando mejores enfoques para el desarrollo de nuestro País.

### **1.3. Hipótesis general**

El Diseño de un sistema para el monitoreo utilizando Internet del Todo y Redes FOG, permitirá mejorar la producción de los cultivos nativos, controlando sus parámetros de producción y cultivo, para tener de manera instantánea cualquier deficiencia que el campo pueda tener.

### **1.4. Hipótesis específicas**

- a) Diseñando un sistema para el monitoreo, se podrá disminuir perdidas en los cultivos nativos para así aumentar su valor agregado.
- b) El uso de redes fog y una comunicación móvil, permitirá mantener una telecomunicación entre el agricultor y el campo, de manera remota y controlable desde cualquier lugar con un acceso a internet.

### **1.5. Objetivo general**

Diseñar e implementar un sistema para el monitoreo de cultivos nativos utilizando Internet del Todo y Redes FOG.

### **1.6. Objetivos específicos**

- a) Disminuir pérdidas en los cultivos y poder aumentar su valor agregado
- b) Implementar una interacción con el sistema de monitoreo (Electron) para lograr una telecomunicación entre el agricultor y sus campos.

## CAPITULO II

### II. REVISION DE LITERATURA

#### 2.1. Cultivos nativos

##### 2.1.1. Los cultivos nativos

Los cultivos nativos también denominados especies indígenas o autóctonas, según la biogeografía, son especies que pertenecen a una región o ecosistema determinados. Su presencia en esa región es el resultado de fenómenos naturales sin intervención humana (pasada o actual). Todos los organismos naturales, en contraste con organismos domesticados, tienen su área de distribución dentro de la cual se consideran nativos. Fuera de esa región si son llevadas por los humanos se las considera especies introducidas.

Por ejemplo la hormiga argentina *Linepithema humile* es una especie nativa de Argentina, Uruguay, Paraguay y sur de Brasil. Fue llevada accidentalmente a otros lugares del mundo, incluyendo África, Europa, Australia, Nueva Zelanda, Japón y Norteamérica, donde se ha establecido como especie introducida.

Una especie nativa no es necesariamente endémica. En biología y ecología endémica quiere decir nativo exclusivamente de una biota específica. Una especie

indígena puede ocurrir en más de un local. Por ejemplo dos especies de grullas africanas, la *Anthropoides paradisea* o grulla del paraíso es nativa y endémica del sur de África, mientras que la grulla *Bugeranus carunculatus* o grulla carunculada es nativa del sur y norte de África. La última tiene una distribución más amplia, por consiguiente no es considerada endémica.

Tampoco quiere decir que se originó en ese lugar. Las especies pueden trasladarse o ser trasladadas a otros locales donde se establecen en el curso de su historia evolutiva.

El conjunto de especies nativas constituyen la flora y fauna autóctonas. Están bien adaptadas entre sí y forman un ecosistema. Cuando las especies nativas son destruidas u otras especies son introducidas ese equilibrio se altera y el ecosistema puede llegar a degradarse hasta el punto que no pueda restablecerse por sí mismo. En ciertos casos se realizan esfuerzos para restaurar la vegetación y fauna de lugares degradados, lo cual requiere planeamiento cuidadoso. Ciertas especies no pueden restablecerse a menos que ya estén presentes otras, es decir que es necesario seguir un proceso de sucesión empezando por plantar especies pioneras, seguidas por especies secundarias y terminando con las de clímax.

### **2.1.2. Agricultura en el Perú**

Alrededor del 1% de la población activa del Perú se dedica a las actividades agrícolas. La mayor parte de las tierras de la región de La Costa se destinan al cultivo de productos de exportación, mientras que los productos para consumo interno provienen de las regiones de La Sierra y La Selva. Junto a las predominantes propiedades agrarias de pequeño tamaño, dedicadas a cultivos de subsistencia, han surgido las grandes cooperativas agrícolas y ganaderas. En 2006 algunos de los principales productos

agrícolas (con el rendimiento anual aproximado en toneladas métricas) fueron: caña de azúcar (7,60 millones), papa o patata (3,29 millones), arroz (2.225.000), maíz (1.230.000), algodón (70.000), café (174.955) y trigo (195.000). El Perú es el principal productor de coca, de cuyas hojas se obtiene, después de un largo proceso de refinado, la cocaína.

"La agricultura puede ser tan importante como la minería. El sector absorbe el 25% de la mano de obra en el país, genera unos US\$6.000 millones en agro exportaciones y ha sido el impulsor del pleno empleo en Ica, La Libertad y Piura".

(<http://elcomercio.pe/economia/peru/agricultura-puede-tan-importante-como-mineria-noticia-1904599>)

### 2.1.3. Cultivos nativos del Perú

En nuestro país existen muchísima diversidad de cultivos nativos, tanto en la costa sierra y selva, sin embargo en los últimos años, muchos han ganado mucha presencia como son los siguientes.

- Quinoa
- Maca
- Sacha Inchi
- Uña de gato
- Aguaymanto
- Yacón
- Hercampuri
- Mashua
- Kiwicha

En el presente proyecto, se tomara en cuenta el cultivo de la quinua, por ser unos de los cultivos nativos que más se desarrolla en nuestra región, sin embargo se puede aplicar a cualquier cultivo nativo, ya que para su aplicación solo dependerá de los parámetros de cultivos para que este funcione de la misma forma en cualquier cultivo.

## **2.2. Cultivo de la quinua**

Para poder definir un cultivo en este caso de la quinua (puede ser cualquier otro cultivo nativo), debemos tener en cuenta sus parámetros de cultivo ya que estos variaran según el lugar autóctono del cultivo nativo, en el presente proyecto tomaremos a nuestro grano estrella de nuestra región, la Quinua, para su análisis de cultivo.

### **2.2.1. Parámetros de cultivo**

Para un cultivo puede haber diversidad de parámetros, por lo tanto primeramente enunciaremos los parámetros generales para un cultivo y posteriormente tomaremos los más importantes que se procesaran en el presente proyecto, tomando como referencia el cultivo de la quinua.

#### ***a) Radiación solar***

“Es la cantidad de energía que el sol suministra a la tierra en forma de radiación electromagnética. Esta energía se emplea en los procesos fotosintéticos, calentamiento del suelo, calentamiento del aire, evaporación y transpiración. El clima a su vez, está determinado por el efecto de esta energía sobre el calentamiento del aire y la evaporación.” (Colón y Rodríguez, 1996).

“La radiación solar es el principal factor que determina el microclima de un área geográfica. Su energía condiciona la temperatura del aire y del suelo, el movimiento del

viento, la evapotranspiración y la fotosíntesis, de tal manera que la intensidad de la radiación, el grado de interceptación y la eficiencia en el uso de la energía radiante, son factores determinantes en la tasa de crecimiento de las plantas.” (Jaramillo, 1999).

“Las plantas son organismos transformadores de la energía solar, ya que mediante la fotosíntesis se obtiene el rendimiento potencial de los cultivos, e indirectamente por su influencia en la evaporación, afecta el balance hídrico del suelo, el que a su vez, define el suministro de agua en las plantas.” (Gómez y Guzmán, 1995). Frere et al. (1974), destacan la radiación como el principal elemento climático por las siguientes consideraciones:

- La cantidad de radiación solar recibida rige los procesos fundamentales de transformación de energía a través de la fotosíntesis.
- La cantidad de radiación solar recibida en cualquier punto de la superficie terrestre y los continuos intercambios de radiación de onda corta y larga entre la tierra y la atmósfera, condicionan la temperatura local y las variaciones estacionales y anuales de la temperatura, regulando de este modo la distribución de los cultivos en la superficie terrestre.
- La radiación constituye la fuente fundamental de energía en el ciclo hidrológico y ejerce gran influencia en las posibilidades agrícolas de cada región a través de las características de la lluvia, que a su vez está sujeta a la influencia, en las zonas tropicales como la nuestra, del movimiento aparente estacional del sol y de las consiguientes variaciones de la distribución de la radiación.

#### ***b) Temperatura***

“La luz y la temperatura son los factores del medio ambiente que influyen directamente sobre la tasa de crecimiento foliar bajo condiciones óptimas de



disponibilidad hídrica. La intensidad de luz determina la tasa fotosintética y por consiguiente el suministro de asimilados para la hoja. La temperatura afecta la tasa de división y expansión celular.” (Arcila y Chávez, 1995).

“Cuando existen pequeñas diferencias de radiación solar incidente en varios sitios, la temperatura se constituye en el principal factor climático que caracteriza el potencial de crecimiento. La tasa de crecimiento de las plantas está relacionada con la cantidad de radiación solar interceptada por el cultivo y por consiguiente la cantidad de biomasa producida.” (Lambert et al., 1999).

“La temperatura optima está claramente relacionado con el clima y especialmente con la temperatura del aire.” (Giraldo, 1989).

### *c) Precipitación*

“La precipitación se considera como la primera variable hidrológica y es la entrada natural de agua dentro del balance hídrico de los agro-ecosistemas y de las cuencas hidrográficas. Se le puede llamar precipitación a la caída del agua de las nubes, ya sea en estado sólido o en estado líquido. Las pequeñas gotas de agua que forman las nubes son de dimensiones tan diminutas que se necesita reunir cientos de miles de estas gotitas para formar una gota de llovizna, y varios millones para formar una gota grande de lluvia.” (Fuentes, 1989). Castañeda (1995) indica: “Por precipitación se entiende el agua proveniente del vapor de agua atmosférico que ha sufrido un proceso de condensación depositándose, bajo cualquiera de sus formas, sobre la superficie de la tierra.”

“La lluvia ejerce sobre los terrenos influencia mecánica, ya que tiene la capacidad de arrastrar sedimentos de la capa superficial del suelo, ejerce disgregación de las partículas, es el medio de transporte de nutrientes desde el suelo hacia la planta ya que

actúa como disolvente de los mismos, a la vez que aporta cantidades de nitrógeno amoniacal y nítrico presentes en la atmósfera. Una gota de lluvia que cae al suelo posee una energía cinética, que al chocar puede producir movimiento de partículas o romper terrenos u otros agregados, llegando a causar erosión.” (Torres, 1995). La distribución de las lluvias es una herramienta muy acertada para efectuar estudios de zonificación climática e identificación de zonas de vida, en las regiones tropicales como Colombia, en donde la radiación y la temperatura no presentan grandes variaciones a lo largo del año; el comportamiento y estacionalidad de la agricultura y la vegetación a una misma altitud están directamente relacionados con la disponibilidad del agua en el suelo proveniente de las lluvias. Las decisiones que se toman en la agricultura, tales como cambios de cultivos, siembra de nuevas variedades, técnicas de siembra, y manejo de sistemas de producción, presentan un riesgo que puede ser disminuido a partir del conocimiento de la probabilidad de ocurrencia satisfactoria de los elementos favorables a la producción.

“Desde un punto de vista exclusivamente climático, estas probabilidades corresponden a factores que influyen en los rendimientos, tales como la disponibilidad de agua. Entre tanto la consideración de promedios en climatología puede llevar a tomar decisiones erradas, es necesario por tal motivo hablar de probabilidades de ocurrencia.” (Eldin, 1980).

“Dentro de los elementos meteorológicos, la lluvia es la que presenta mayor variabilidad en nuestro medio, por lo que es el factor de mayor riesgo para la agricultura.” (Verá y González, 1995).

#### *d) Humedad relativa*

La humedad relativa es una medida que permite saber qué tan húmedo o seco se encuentra el aire. Se expresa en unidades enteras correspondiendo el 0 (cero) a la sequedad absoluta y el 100% a la saturación. El aire ejerce una fuerza en la superficie de todos los objetos con los que está en contacto. El aire, como cualquier otro cuerpo está sujeto a la acción de la gravedad, tiene un peso, y ejerce por lo tanto, una presión sobre la superficie terrestre. La cantidad de presión producida por la composición de las moléculas de gas en el aire depende de la masa de las moléculas, el empuje de la gravedad y la actividad dinámica de las moléculas. Dentro de la atmósfera, el agua se encuentra en tres fases; vapor de agua, que es invisible; pequeños cristales y gotas de agua en nubes visibles. De la cantidad de vapor de agua contenido en la atmósfera dependen las posibilidades de lluvia, y de éstas la producción de energía hidráulica.

“El vapor de agua además absorbe el calor, y por consiguiente actúa como regulador de la temperatura, su acción es decisiva sobre los seres vivos, y su ausencia o proporciones mínimas en regiones áridas es el principal obstáculo que se opone al desarrollo de la vida en tales regiones.” (Candel, 1971).

“La cantidad de vapor de agua presente en el aire por unidad de masa o de volumen, determina el grado de humedad en la atmósfera.” (Jaramillo, 1999).

“La fuente principal de humedad del aire es la superficie de los océanos, de donde se evapora el agua de manera constante, pero también contribuyen a su formación los lagos, ríos, glaciares, campos de nieve, la transpiración de las plantas” (Moran y Morgan, 1991; Fuentes, 1989; Candel, 1971).

“La atmósfera no puede contener una cantidad ilimitada de vapor de agua, sino que llega un momento en que ya no puede contener más. Cuando llega ese momento se dice que la atmósfera está saturada.” (Fuentes, 1989).

“El punto de saturación de la atmósfera depende de la temperatura: cuanto más caliente está el aire, mayor cantidad de vapor de agua puede contener; para toda solución se cumple que entre mayor sea la temperatura mayor es su punto de saturación hasta llegar a un punto máximo. Cuando el aire está saturado de vapor de agua y baja la temperatura, ocurre que el aire no puede contener todo el vapor de agua que contenía anteriormente; el exceso de vapor de agua pasa a formar pequeñas gotas de agua o cristales de hielo según sea la temperatura del aire.” (Fuentes, 1989).

#### e) *Vientos*

“El viento es el movimiento de aire en la superficie terrestre. Es generado por la acción de gradientes de presión atmosférica producida por el calentamiento diferencial de las superficies y masas de aire.” (Jaramillo, 1999).

La superficie de la tierra se calienta por la radiación solar; esta radiación solar no se recibe con la misma intensidad en todas las zonas del planeta, lo que origina un calentamiento desigual de las masas de aire. El aire de las capas atmosféricas más bajas se calienta bajo la influencia de la superficie terrestre, siendo su calentamiento más o menos intenso según la temperatura que alcanzan las diferentes zonas de la superficie terrestre con las que se mantiene en contacto.

“En general existe la tendencia a que cualquier desequilibrio presente a nivel de la atmósfera, tiende a equilibrarse de manera natural. El desequilibrio creado por la diferencia de presión tiende a equilibrarse de una forma natural mediante el

desplazamiento de aire de la zona de mayor presión a la de menor presión; este desplazamiento de aire horizontal recibe el nombre de viento.” (Fuentes, 1989).

Desde el punto de vista ecológico, el conocimiento de la variabilidad del viento tiene implicaciones amplias en la agricultura y en el manejo de los suelos. Los vientos influyen en:

- La remoción de CO<sub>2</sub>
- Transferencia y/o remoción de vapor de agua
- Transporte de insectos, polen y esporas de enfermedades
- Cambios en la humedad atmosférica local
- Aumento en las tasas de evapotranspiración
- Pérdidas en las aplicaciones de agua en los sistemas de riego por aspersión
- Cambios térmicos en las primeras capas del suelo
- Pérdidas de suelos por erosión eólica

#### *f) Temperatura del suelo*

La Temperatura del suelo agrícola condiciona los procesos microbianos que tienen lugar en el suelo. La temperatura también influye en la absorción de los nutrientes, especialmente del fósforo que es menor en suelos fríos.

La temperatura del suelo es importante porque influye en los procesos bióticos y químicos. Es importante para la germinación y cada planta tiene sus requerimientos determinados. Normalmente por encima de los 5° C es posible la germinación.

La capa superficial del suelo agrícola sufre las mayores oscilaciones de temperatura. La temperatura de la capa superficial se trasmite hacia arriba al aire, y hacia abajo a capas inferiores del suelo y al subsuelo, amortiguándose rápidamente sobre todo hacia abajo.

### 2.2.2. Parámetros de cultivo de la quinua

- Región Puno, las mayores áreas de siembra se dan entre fines del setiembre hasta la quincena de noviembre. Es la zona, que siembra de manera adelantada porque la gran mayoría de sus unidades productivas se encuentra por encima de los 3500 msnm.
- En las regiones de Ayacucho, Apurímac, Junín y Cusco, en su mayoría las grandes siembras inician entre la quincena de octubre hasta fines de noviembre.  
  
Sin embargo en las zonas bajas de estos departamentos las siembras se prolongan hasta fines de diciembre, en el caso particular de Ayacucho (zonas bajas) que cuentan con sistemas de riego las siembras son todo el año.
- La zona de Sierra Norte (caso La Libertad), las siembras son las más retrasadas, inician a la quincena de noviembre y terminan a fines de diciembre, en las zonas bajas por debajo de los 3000 msnm, las siembras terminan en el mes de enero.
- En la costa norte (Lambayeque y Piura) y Costa Sur (Ica y Arequipa) la mejor época de siembra es de mayo hasta agosto; sin embargo, dependerá de la disponibilidad de agua, de la variedad y altitud.

Debemos tener en cuenta que la temperatura óptima media varía en un rango de 5 – 15 C y una oscilación térmica de 5-7 C.

No solo es necesario saber los parámetros como tales, sino que los productos tienen estándares de calidad, como se muestra en la tabla n°1:

*Tabla n° 1: Parámetros de calidad para el cultivo de la quinua.*

Estándares de Calidad	
Característica	Parámetro
Pureza varietal	100% (debe corresponder a la variedad negociada)
Pureza física	Sin restos de cosecha, materias extrañas y/o semillas de malezas
Granulometría	Mayores a 1.4 mm de diámetro
Granos amorfos	Menor a 1% (granos partidos)
Granos Cristalinos (ojo de gallo)	No hay tolerancia (0.00 %)
Granos infestados - infectados	0% (libre de plagas - enfermedades)
Color de grano	Característico de la variedad (Blanco – Amarillo - Rojo – Negro)
Textura de suelo	Franco, franco- arenoso
Humedad	10% - 11.50%
Clima	Semiseco frío
Precipitación	250 - 500 mm
Temperatura óptima	6°C a 17° C
pH	5.5 – 7.8
En Almacén	
Temperatura máxima	21°C (70°F)
Humedad máxima	70%

*Fuente: INIA*

Del cuadro podemos decir entonces, que la mayoría de los parámetros están dados en base a los principales parámetros, que son los de cultivo: la temperatura, humedad, pH y precipitación, manteniendo esos parámetros en sus valores óptimos, los demás parámetros también serán óptimos puesto que estos ya se dan cuando el cultivo está listo para cosechar.

### **2.3. Problemas y cuidados en el cultivo de la quinua**

Como se expuso se optó por usar la quinua, ya que es un cultivo nativo de nuestra región, sin embargo uno de los principales motivos, fue porque nuestro cultivo puede llegar a padecer diversos problemas en el proceso de cultivo, causando de esa manera pérdidas porcentuales e incluso totales del cultivo, para lo cual en el presente capítulo explicaremos los principales problemas y también las formas empíricas y avanzadas de su cuidado.

#### **2.3.1. Problemas en el cultivo de la quinua**

Entre 2005 y 2014 se registró un incremento en áreas cultivadas de 250%, alcanzando las 68,000 hectáreas y más de 100,000 toneladas de producción, así como un crecimiento importante de sus exportaciones, sobrepasando los 142 millones de dólares en 2014.

Si bien la región Puno representa el 32% de la producción nacional de la quinua, enfrenta aún brechas de productividad que inciden en la pérdida del volumen por hectárea y en la calidad del grano cosechado.

En el presente capítulo abarcaremos antecedentes y situaciones actuales que ocasionan problemas en el cultivo de la quinua.



**a) *Insectos plaga del cultivo de la quinua***

“Durante todo su periodo vegetativo, el cultivo de la quinua es afectado por una amplia gama de insectos, de los cuales fueron identificadas alrededor de 17 especies que concurren al cultivo de la quinua. Entre las plagas de mayor importancia económica se encuentran la polilla de la quinua (*Eurysacca melanocampta* Meyrick) y el complejo ticonas (*Copitarsia turbata*, *Feltia* sp, *Heliothis títicaquensis*, *Spodoptera* sp)” (Saravia y Quispe, 2005).

Las pérdidas ocasionadas por estas plagas pueden oscilar entre un 5 a 67%, con un promedio de 33.37 % en el Altiplano Sur y entre 6 a 45% en el Altiplano Centro, con un promedio de 21.31%.

**b) *Aves plaga***

Sólo en la región Puno, durante la última campaña agrícola (2011 – 2012), las “aves plagas” ocasionaron pérdidas en la producción de quinua de hasta 30%.

En promedio las pérdidas por hectárea ocasionadas por “aves plaga” superaron los 2 mil nuevos soles y en toda la región Puno bordearon los 66 millones de soles en la campaña 2011 – 2012. (Pérdidas de S/. 2,172.00 por ha y pérdidas para la Región Puno de S/. 65 892 732.86).

A través del proyecto “Aves plaga del cultivo de la quinua y alternativas de control”, liderado por el investigador del INIA Pedro Delgado Mamani, se viene evaluando cuatro alternativas de protección del cultivo como el enmallado de los campos, instalación de equipos que emitan sonidos de aves depredadoras o la implementación cintas vibradoras y banderines brillantes.

### *c) Humedad*

“La quinua es uno de los cultivos considerados como delicados en cuanto a manejo y cuidados de la cosecha. La cosecha de quinua debe realizarse con la debida oportunidad para evitar no solo las pérdidas por efectos adversos del clima y ataque de aves sino, el deterioro de la calidad del grano. Si a la madurez del cultivo hay un período de humedad ambiental alta (superior al 70%), se produce la germinación de los granos en la panoja, con la consiguiente pérdida de la cosecha o por lo menos se produce una oxidación o cambio de color de los granos, con la consiguiente pérdida de la calidad de la cosecha.” (Nieto y Vimos 1992).

“La quinua debe ser cosechada cuando las plantas se hayan defoliado y presenten un color amarillo pálido o los granos hayan adquirido una consistencia tal que resistan a la presión con las uñas. En áreas húmedas como la Sierra Norte de Ecuador y Sur de Colombia, es aconsejable hacer coincidir la cosecha de la quinua con la época seca del año (de junio a septiembre), para evitar pérdidas o deterioro de los granos por efecto de la humedad ambiental.” (Nieto, Castillo y Peralta, 1986). Este motivo puede llegar a afectar hasta el 33%.

### *d) Temperatura*

Para Ulises Osorio, docente de la Universidad Agraria “La Molina” e investigador del impacto del Fenómeno de El Niño en temas agrícolas, la producción de quinua en la costa caerá en 70% debido al incremento de la temperatura (que se estima subirá 1,5°C por encima del promedio).

e) *Heladas*

Heladas Ocurre cuando hay descensos extremos de temperaturas por debajo de menos 4C, bajo estas condiciones se producen alteraciones fisiológicas en las células de las plantas, rupturas del plasma por la presencia de cristales de hielo en los espacios intercelulares. Normalmente ocurren heladas en los meses de junio, julio, agosto cuando el cielo está despejado, no hay nubes; pero puede ocurrir durante la campaña agrícola en determinados momentos. Sin duda este es uno de las causas más fuertes ya que pueden llegar a perder el cultivo completo.

En la tabla n° 2 se resume los porcentajes de perdidas según el motivo:

*Tabla n° 2: Porcentaje de pérdidas en el cultivo de quinua según el motivo.*

Resumen de perdidas	
Motivo	Porcentaje
Insectos plaga	5 – 67%
Aves plaga	30%
Humedad	33%
Temperatura	70%
Heladas	100%

*Fuente: INIA*

Otro factor que puede intervenir es quizás menos nocivo pero que también perjudica son las malezas, las malezas o malas hierbas es un conjunto de plantas que crecen entre los cultivos y los perjudican porque se apropian de los elementos nutritivos del suelo y los privan de luz, aire, humedad y espacio.

### 2.3.2 Cuidados en el cultivo de la quinua

Respecto a los cuidados del cultivo, dependiendo del problema y metodología del agricultor, estos cuidados pueden ser empíricos y avanzados con la tecnología y demás ayuda, pero siempre existen también los cuidados constantes como el riego, abonado, etc.

#### a) *Riego*

El riego es una labor que permite distribuir artificialmente el agua sobre la superficie de la tierra, los sistemas de riego más utilizados son por gravedad y por aspersión.

Consiste en proporcionar agua al suelo para que las plantas se desarrollen normalmente. Los cultivos hortícolas deben recibir abundante cantidad de agua no sólo durante la germinación sino también en el crecimiento. En los semilleros el riego se hace por aspersión usando regaderas o mangueras con boquillas esparcidoras. El riego se hace diariamente durante las primeras horas de la mañana o las últimas de la tarde.

El riego por gravedad o surcos permite tomar el agua de la parte alta del terreno, donde se encuentra la fuente y distribuirla a través de surcos que van a irrigar el suelo. Este sistema tiene la ventaja de que no se necesita equipo especial y se puede aplicar en terrenos inclinados sin peligro de erosión.

El riego por aspersión consiste en proyectar el agua en forma de lluvia. Este sistema se aplica por medio de un sistema de tubos de aluminios conectados a la fuente de agua para que el líquido sea distribuido a presión por medio de molinetes hidráulicos.

**b) Control de plagas.**

La plagas en el semillero se combaten aplicando insecticidas, que son productos químicos que eliminan a los insectos perjudiciales.

**c) Control de enfermedades.**

Las enfermedades causadas por hongos se combaten con sustancias químicas llamadas fungicidas. Los fungicidas son productos a base de azufre y cobre.

**d) Control de heladas****i. Riego**

Aspersión o nebulización: Se trata de utilizar un sistema de riego por aspersión para “aislar” las plantas contra las temperaturas bajas. Básicamente se aprovecha el calor latente en el agua al solidificarse. En algunos casos provoca la modificación calórica del suelo. El agua, al caer sobre la planta se congela liberando una determinada cantidad de calor latente que recibirá el cultivo en el lugar donde la gota de agua cae.

El uso del agua para contrarrestar las bajas temperaturas es un método muy antiguo debido la capacidad del agua de absorber y liberar energía calórica.

Para que este método funcione es importante estar pendiente de seguir con la aspersión de agua hasta que la temperatura suba a menos 1oC, lo que se puede confirmar con un termómetro calibrado colocado en el punto de elevación más bajo y se descongele el hielo en las plantas. Una desventaja de este método es que se consume mucha agua.

El calor latente que se libera cuando el agua pasa de estado líquido a sólido se utiliza como medio de protección ante la presencia de una helada. De este modo sistemas de riego controlados de forma automática mediante la presencia de sensores pueden ser útiles como medio de defensa de heladas.

Cuando la temperatura desciende bajo los  $0^{\circ}\text{C}$  el agua liberada forma una película de hielo alrededor de las hojas y ramas que proporciona la protección necesaria para mantener la temperatura alrededor de los  $0^{\circ}\text{C}$ .

La cantidad de agua liberada por el sistema debe ser suficiente para compensar las pérdidas de energía por radiación, convección, advección y por evaporación. Además la velocidad de rotación del aspersor debe ocurrir en menos de un minuto.

En presencia de heladas por advección, que además de bajas temperaturas presenta vientos, la utilización de este método es muy arriesgado y deben aplicarse cantidades de agua hasta unos 50% mayores. Por otro lado en días claros y sin viento debe detenerse el sistema cuando haya agua líquida entre el hielo y las ramas, mientras que en días nublados o con viento debe mantenerse funcionando durante el día. Este sistema de riego es el que da mayor protección mostrando una relación eficacia/coste superior al 80%, sin embargo presenta un mayor riesgo de enfermedades fúngicas y daños mecánicos de la planta ante el exceso de peso ejercido sobre las ramas por la presencia de hielo.

Otro tipo de riego es el de aspersión por debajo de la cubierta vegetal que consiste en la presencia de micro aspersores que cumplen este rol. Además del calor latente del agua aplicada este sistema aporta el calor sensible que se genera al vaporizarse agua en el ambiente. Dentro de los beneficios que posee este método están el menor riesgo de enfermedades debido a que el agua no moja las yemas y botones florales, además de que la cantidad de agua que debe aplicarse en la mitad del de aspersión por encima de la cubierta vegetal, en la tabla n° 3 se ve el consumo calórico:

*Tabla n° 3: Consumo calórico en el método del riego.*

Estado de agua	Liberación de energía calórica
Hielo	Agua absorbe 80 calorías/gramo
Agua	Hielo libera 80 calorías/gramo
Agua	Vapor absorbe 600 calorías/gramo
Vapor	Agua libera 600 calorías/gramo

*Fuente: INIA*

## *ii. Calentamiento del aire y las plantas*

Existen numerosos sistemas de calentamiento del aire y plantas que varían según tipos de combustible y tipos de estufas. Hay estufas que proporcionan mayor calor que otras, además de la cantidad de humo emitido que varía según el modelo.

Este sistema consiste en la emisión de radiación desde el aparato y un calentamiento del aire por efectos de conducción y convección. Sin embargo el gran problema que posee es que la radiación térmica emitida es poco específica y posee un radio de acción grande que actúa en todas direcciones. Por ende el efecto de radiación es reducido y no del todo aprovechado por la planta.

Se ha comprobado que se obtiene un mejor rendimiento mediante la utilización de pequeños y numerosos focos de calor frente a pocos focos de calor de gran potencia calorífica.

De todo lo dicho anteriormente, lo ideal es prevenir y no combatir cuando se está dando estos aspectos negativos, para lo cual una de las mejores soluciones sería un sistema de monitoreo que pueda predecir los comportamientos y cambios y así prevenir los males que puedan suceder en el cultivo.

## 2.4 Sistemas de monitoreo

Aunque muchos productores consideran al monitoreo como un costo más de producción, se trata de una inversión. Esta inversión evita costos innecesarios, tanto económicos como ambientales. Uno de los países en Latinoamérica donde se va usando en mayor proporción es Argentina, donde el monitoreo va tomando cada vez mayor relevancia. A principio de los años 2000 apenas se monitoreaba un 5% de la superficie de soja. Hoy en día más del 50% de los productores conocen los beneficios del monitoreo y, con mayor o menor técnica, monitorean el cultivo.

### 2.4.1 Sistema de monitoreo de cultivos de hoja de coca en el Perú

El Perú forma parte del Programa Global de Monitoreo de Cultivos Ilícitos (ICMP) que tiene sede en Viena y a través de UNODC apoya a los Estados Miembros en la generación de información cartográfica y estadística sobre la evolución anual de cultivos ilícitos de coca y amapola. Actualmente tiene establecido Sistemas de Monitoreo en Afganistán, Myanmar, Laos, Marruecos, Colombia, Perú, Bolivia y últimamente en Ecuador. En el Perú inició operaciones en el año 1999, teniendo como contraparte nacional a DEVIDA, entidad del Estado encargada de diseñar, coordinar e implementar políticas y actividades dirigidas al control de drogas.

El sistema implementado por UNODC en el Perú ha sido diseñado y estructurado para cartografiar y medir anualmente a nivel nacional, la extensión ocupada por cultivos de coca en áreas tradicionales y nuevas; determinar la oferta anual de hoja de coca y de clorhidrato de cocaína y registrar mensualmente los precios relativos de la hoja y derivados de coca a nivel de distritos con el fin de definir las tendencias de la actividad cocalera. Complementariamente, tiene la capacidad de estructurar Sistemas de



Información Geográfica (SIG), con el objetivo de zonificar el potencial agropecuario y/o forestal de las áreas intervenidas por coca. Esta información constituye la base técnica indispensable para formular y planificar el desarrollo rural de cuencas con problemas de cultivos ilícitos.

Desde el año 2000, el proyecto ha evolucionado en cuanto al empleo de productos sensoriales y metodologías de procesamiento así como mecanismos de verificación terrestre y aérea. Al inicio se procesó aerofotografías a la escala 1:20,000, material que permitió estructurar una base de datos cartográficos de cultivos de coca con elevado nivel de detalle. Posteriormente, con el fin de actualizar la dinámica evolutiva, se ha venido empleando productos satelitales multiespectrales SPOT4 de 20x20m de resolución espacial, SPOT5 de 10x10m de resolución, FORMOSAT2 de 8x8m, KOMSAT e IKONOS, ambos de 4x4m.

Este mismo sistema se puede replicar para otros productos, sin embargo este tipo de monitoreo se hace de manera masificada y genérica, dependiendo del estado y de un gran presupuesto, sin embargo no existe opciones de monitoreo para los pequeños y medianos productores, de fácil accesibilidad económica y de uso, es por esto que se propone una solución enfocada en aquellos productores medianos y pequeños y puedan tener un mejor cultivo y valor agregado a sus productos.

#### **2.4.2 Aplicaciones móviles**

Otra alternativa en los últimos años es la del uso de aplicaciones móviles una de las más conocidas llamada “SIMA”, sin embargo esta aplicación aún están en desarrollo y aun no tiene algunos producto endémicos en su base y su exactitud aún está en proceso de mejora.

### 2.4.3 Beneficios de la agricultura de precisión

- Monitoreo del clima.
- Automatización de procesos.
- Manejo de información de cultivo
- Reportes en tiempo real
- Alarmas
- Históricos
- Disminución de trabajo
- Optimización de procesos
- Condiciones ideales para producción

### 2.5 Internet del Todo

Internet evolucionó de una manera que jamás hubiéramos imaginado. Al principio, los avances se daban lentamente. Hoy en día, la innovación y la comunicación se producen a gran velocidad.

Desde su humilde comienzo como Advanced Research Projects Agency Network (ARPANET) en 1969, que interconectaba unos pocos sitios, hoy se predice que Internet interconectará 50 000 millones de objetos para el año 2020. En la actualidad, Internet proporciona conexiones globales que hacen posible que exista la navegación web, los medios sociales y los dispositivos móviles inteligentes.

### 2.5.1 Evolución de Internet

La evolución de Internet experimentó cuatro fases distintivas. Cada fase tiene un efecto más profundo en los negocios y en la sociedad que la fase anterior.

#### *a) Fase 1*

La primera fase comenzó hace 20 años y se la denomina “conectividad”. El correo electrónico, la navegación web y la búsqueda de contenido fue solo el principio.

Conectividad

Digitalizar el acceso a la información

- Correo electrónico
- Navegador web
- Búsqueda

#### *b) Fase 2*

La segunda fase comenzó a fines de la década de los noventa y fue la fase de la “economía interconectada”. Ese fue el comienzo del comercio electrónico y de las cadenas de suministro conectadas digitalmente. Cambió la forma en que hacemos compras y en que las empresas llegan a nuevos mercados.

Economía interconectada

Digitalizar el proceso empresarial

- Comercio electrónico
- Cadena de suministro digital
- Colaboración

*c) Fase 3*

La tercera fase comenzó a principios de la década de 2000 y se conoce como la fase de las “experiencias cooperativas”. Esta fase se rige por el amplio uso de los medios sociales, la movilidad, los servicios de video y la computación en la nube. Esta fase transformó por completo el mundo laboral.

Experiencias cooperativas

Digitalizar las interacciones (empresariales y sociales)

- Social
- Movilidad
- Nube
- Video

*d) Fase 4*

La fase actual se denomina “Internet de todo” (IdT). En esta fase, se conectan personas, procesos, datos y objetos, lo que transforma la información en acciones que crean nuevas capacidades, experiencias más valiosas y oportunidades sin precedentes.

Internet de Todo

Digitalizar el mundo mediante la conexión lo siguiente:

- Personas
- Procesos
- Datos
- Objetos

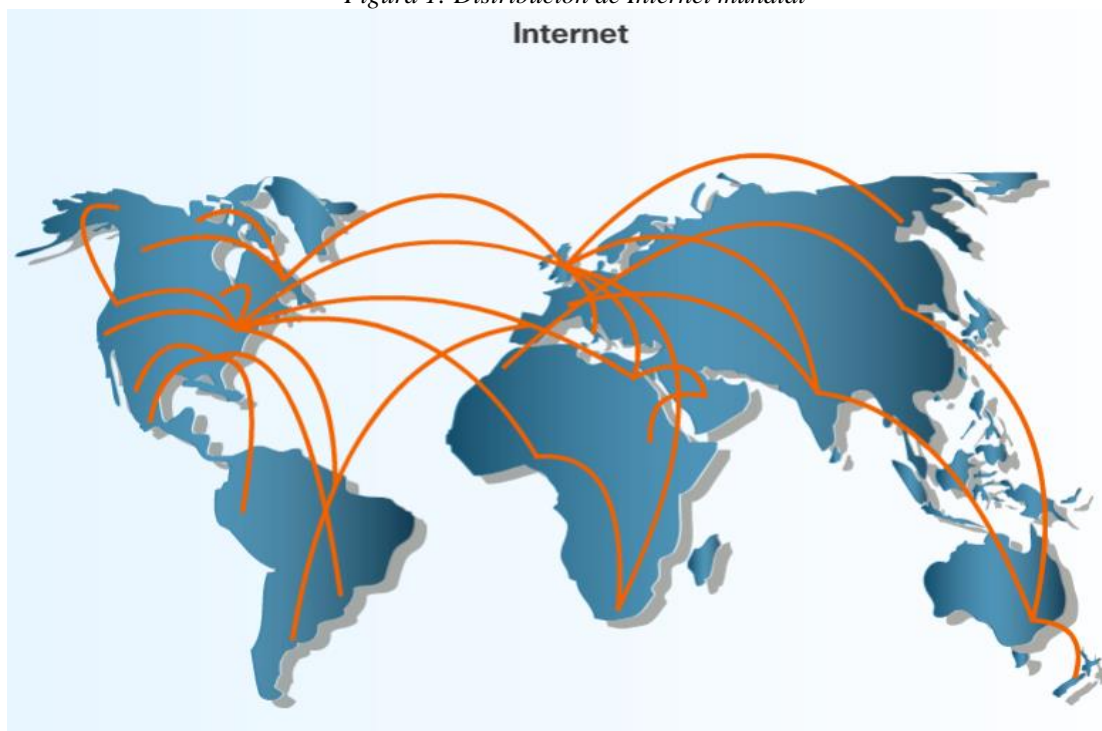
Generalmente, cuando las personas utilizan el término Internet, no se refieren a las conexiones físicas en el mundo real. En realidad, la suelen imaginar como una colección amorfa de conexiones. Es el “lugar” al que se acude para buscar o compartir información. Es la biblioteca, la tienda de videos y el álbum de fotografías personal del siglo XXI.

En la práctica, Internet es, básicamente, una red de redes.

Cada uno de nosotros se conecta a Internet mediante un cable físico o medios inalámbricos. Bajo esta red de redes, yace una verdadera red troncal de conexiones que acercan el mundo a nuestros dispositivos informáticos personales.

En la figura n°1 se observa aunque de manera simplificada las conexiones entre países y continentes, se entiende que existe un gran trabajo de ingeniería para lograrlo.

*Figura 1: Distribución de Internet mundial*



*Fuente: <http://www.cibersociedad.net/congreso/comms/g22solis.htm> (2017)*

### 2.5.2. Transición a IdT

- En la actualidad, más del 99% de los objetos que se encuentran en el mundo físico aún no están conectados a internet
- Sin embargo, un fenómeno denominado “Internet de Todo” activara todo lo que pueda imaginarse
- Internet de Todo conecta el mundo físico a Internet
- Mediante el uso de microsensores en la red, los objetos cotidianos se conectaran y volverán inteligentes.
- Para el año 2020, habrá 50 000 millones de objetos conectados a Internet.

En muy poco tiempo, Internet modificó radicalmente la forma en que trabajamos, vivimos, jugamos y aprendemos. Sin embargo, esto recién es el comienzo. Mediante tecnologías existentes y nuevas, conectamos el mundo físico a Internet. La transición de Internet a Internet de todo se da cuando se conecta aquello que no está conectado.

### 2.5.3. Pilares de IdT

La idea de conectar objetos no es nueva. De hecho, Internet de las cosas (IdC) es un término ampliamente aceptado desde finales de la década de los noventa. “IdC” hace referencia a la red de objetos físicos a los que se puede acceder mediante Internet.

No todos los objetos que se conectan a IdC son dispositivos informáticos, pero muchos sí. Entonces, ¿qué es un dispositivo informático? Si bien quizá sea fácil identificar una computadora de escritorio o portátil, puede ser difícil distinguir qué constituye una computadora y qué no. ¿Los automóviles son dispositivos informáticos? ¿Y un reloj o un televisor?

Los primeros dispositivos informáticos (computadoras) eran máquinas inmensas del tamaño de una habitación, cuyo armado, administración y mantenimiento requerían de varias personas. En la actualidad, son exponencialmente más rápidos y mucho más pequeños que sus predecesores. A efectos de este curso, un dispositivo informático es una máquina electrónica que realiza cálculos a partir de un conjunto de instrucciones y que consta de tres componentes principales: una unidad central de procesamiento (CPU), memoria y una unidad de entrada y salida (E/S).

De acuerdo con la definición anterior, un reloj digital es un dispositivo informático, pero un reloj analógico no lo es. El reloj digital tiene una CPU para ejecutar el programa, tiene memoria para almacenar el programa y otra información, y tiene un dispositivo de entrada y salida (E/S) para permitir la interacción con el usuario (pantalla, cuadrante, botones, alertas sonoras, etc.). Si bien el reloj analógico tiene el componente de E/S, no tiene CPU ni memoria

### ***Objetos, Datos, Personas Y Procesos***

IdT incorpora cuatro pilares para lograr que las conexiones en red tengan más importancia y valor que nunca: personas, procesos, datos y objetos. La información de estas conexiones da lugar a decisiones y acciones que crean nuevas capacidades, experiencias más valiosas y una oportunidad económica sin precedentes para personas, empresas y países.

#### ***2.5.3.1. Objetos***

En la actualidad, el pilar de los objetos, se compone principalmente de varios tipos de computadoras y dispositivos informáticos tradicionales, como equipos de escritorio, computadoras portátiles, smartphones, tablet PC, grandes equipos y clústeres de computadoras. Sin embargo, IdC incluye todos los tipos de objetos, aun los objetos y los

dispositivos que no se conectaban tradicionalmente. De hecho, Cisco calcula que, en algún momento del futuro, el 99% de los objetos físicos tendrán conexión.

Estos objetos contienen tecnología integrada para interactuar con servidores internos y con el entorno externo. Además, tienen capacidad de conexión a red y pueden comunicarse mediante una plataforma de red disponible, confiable y segura. Sin embargo, IdC hace referencia a una sola transición tecnológica: la capacidad de conectar objetos que antes no se conectaban, de manera que puedan comunicarse a través de la red.

Cuando los objetos tienen capacidades de detección y comunicación, la disponibilidad de datos puede cambiar la forma y el lugar donde se toman las decisiones, quienes las toman y los procesos que las personas y las empresas usan para tomar esas decisiones. IdT se basa en las conexiones entre las personas, los procesos, los datos y los objetos, estos son los cuatro pilares de IdT, sin embargo, IdT no tiene que ver con el conjunto de estas cuatro dimensiones de forma aislada; cada una amplifica las capacidades de las otras tres. El verdadero poder de IdT surge en la intersección de todos estos elementos.

En pocas palabras, los objetos son objetos físicos conectados a Internet y entre sí. Estos dispositivos detectan y recolectan más datos, lo que los hace sensibles al contexto y hace que proporcionen información más empírica para ayudar a las personas y a las máquinas.



## *Conexiones informáticas comunes*

### *a) Dispositivos comunes*

Internet conecta varios dispositivos informáticos además de las computadoras de escritorio y portátiles. A tu alrededor, hay dispositivos con los que quizá interactúes todos los días y que también están conectados a Internet.

Por ejemplo, día a día las personas utilizan cada vez más los dispositivos móviles para comunicarse y realizar tareas cotidianas, como revisar el pronóstico del tiempo o realizar operaciones bancarias en línea.

En el futuro, es posible que muchos de los objetos en su hogar también tengan conexión a Internet para que se puedan controlar y configurar de manera remota, así como también fuera de su casa, en el mundo exterior, también hay muchos dispositivos conectados que proporcionan comodidad e información útil e, incluso, fundamental.

### *b) Conexión de dispositivos*

Para que IdT funcione, todos los dispositivos que forman parte de la solución deseada de IdT deben conectarse entre sí a fin de que puedan comunicarse. Hay dos formas de conectar dispositivos: con cables o en forma inalámbrica.

En la mayoría de los casos, la conexión de dispositivos entre sí mediante cables es demasiado costosa o engorrosa para ser práctica. Por este motivo, la mayoría de los dispositivos deben poder enviar y recibir datos de forma inalámbrica.

Existen muchos tipos diferentes de comunicación inalámbrica. Los tipos más comunes de comunicación inalámbrica son Wi-Fi, redes de telefonía móvil, Bluetooth y transmisión de datos en proximidad (NFC). Algunos dispositivos, como los smartphones

y las tablet PC, utilizan una combinación de métodos de comunicación inalámbrica para conectarse a diferentes dispositivos.

### *c) Conexión a objetos informáticos no tradicionales*

Según la página web Internet World Stats, hasta junio de 2012, según las estadísticas había aproximadamente 2400 millones de usuarios de Internet. Esto es solo el 34% de la población mundial total.

En 2012, la cantidad de dispositivos conectados a Internet superó a la población mundial. Esto incluye dispositivos informáticos tradicionales y dispositivos móviles, así como también nuevos dispositivos industriales y de consumo que consideramos “objetos”.

Aunque puede parecer que hay demasiados dispositivos conectados a Internet, esto representa menos del 1% de los objetos que podrían conectarse. Entre los dispositivos que actualmente no están conectados, se encuentran los microondas, los despertadores y los sistemas de iluminación.

### *d) Sensores*

Los sensores son una forma de obtener datos de dispositivos que no son computadoras. Convierten los aspectos físicos de nuestro entorno en señales eléctricas que las computadoras pueden procesar. Algunos ejemplos de esto son los sensores de humedad del suelo, los sensores de temperatura del aire, los sensores de radiación y los sensores de movimiento. Todos los tipos de sensores desempeñan una función importante en la conexión de dispositivos que, tradicionalmente, no estaban conectados a IdT, en la Fig. n° 2 se observa un sensor de oxígeno.

Figura 2: Sensor de oxígeno para autos.



Fuente: <http://www.atraccion360.com/que-es-el-sensor-de-oxigeno> (2017)

#### e) **RFID**

Existe un tipo popular de sensor que utiliza identificación por radiofrecuencia (RFID). La RFID utiliza los campos electromagnéticos de radiofrecuencia para comunicar información entre pequeñas etiquetas codificadas (etiquetas RFID) y un lector de RFID. En general, las etiquetas RFID se utilizan para identificar al portador, como una mascota, y hacerle un seguimiento. Debido a que las etiquetas son pequeñas, pueden fijarse a prácticamente cualquier elemento, incluidos ropa y dinero. Algunas etiquetas RFID no utilizan baterías. La energía que la etiqueta necesita para transmitir la información se obtiene de señales electromagnéticas que envía el lector de etiquetas RFID. La etiqueta recibe esta señal y utiliza parte de la energía en ella para enviar la respuesta, como se ve en la fig. n° 3.

A diferencia del código de barras, la RFID depende de la radiofrecuencia; por lo tanto, no requiere una línea de vista para funcionar.

Debido a su flexibilidad y sus bajos requisitos de alimentación, las etiquetas RFID son una excelente manera de conectar un dispositivo que no es una computadora a una solución de IdT mediante la provisión de información a un dispositivo lector de RFID. Por ejemplo, hoy en día es común que las fábricas de automóviles coloquen etiquetas RFID en las carrocerías. Esto permite un mejor seguimiento de cada vehículo en la línea de montaje.

La primera generación de etiquetas RFID fue diseñada para “una sola escritura y muchas lecturas”. Esto significa que pueden programarse en la fábrica una sola vez, pero no pueden modificarse fuera de ella. Las etiquetas RFID más nuevas están diseñadas para “muchas escrituras y muchas lecturas”, y tienen circuitos integrados que pueden durar entre 40 y 50 años, y que pueden escribirse más de 100 000 veces. Estas etiquetas pueden almacenar eficazmente el historial completo del elemento al que están conectadas, como la fecha de fabricación, el historial de seguimiento de ubicaciones, el ciclo de diversos servicios y el propietario.

Figura 3: Conexión RFID.



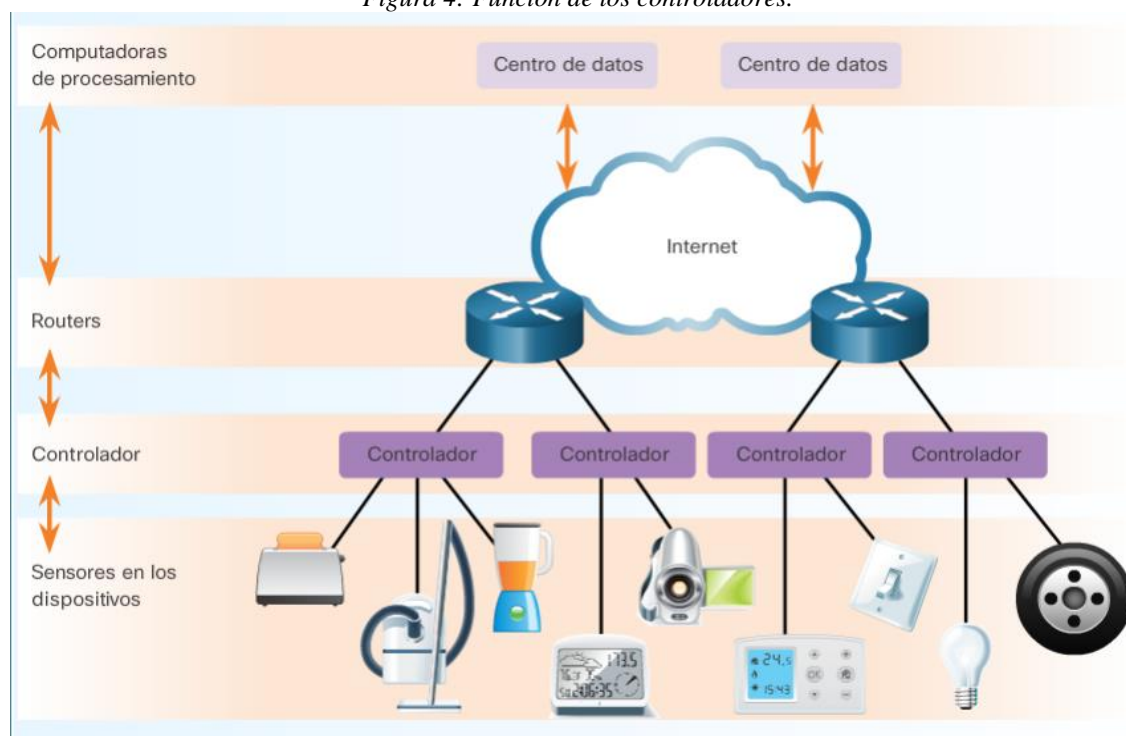
Fuente: <https://ansidotorg.blogspot.com/2015/10/recycling-and-rfid-tags.html> (2017)

### f) Controladores

Los sensores pueden programarse para que tomen mediciones, traduzcan esos datos en señales y después los envíen a un dispositivo principal denominado “controlador”. El controlador es responsable de obtener los datos de los sensores y proporciona una conexión a Internet. Los controladores pueden tener la capacidad de tomar decisiones inmediatas o pueden enviar datos a una computadora más potente para su análisis. Esta computadora más potente puede estar en la misma LAN que el controlador, o bien puede ser accesible únicamente por medio de una conexión a Internet.

Para acceder a Internet y luego a las computadoras más potentes en el centro de datos que se muestra en la figura, el controlador primero envía los datos a un router local. Este router comunica la red local con Internet y puede reenviar datos entre ambas, como se muestra en la fig. n° 4, aquí es donde hacemos la transición de IdC a IdT.

Figura 4: Función de los controladores.



Fuente: [www.cisco.com](http://www.cisco.com) (2017)

### 2.5.3.2 Datos

Los datos son un valor asignado a todo lo que nos rodea; están en todas partes. Sin embargo, por sí solos, los datos no tienen sentido. Los datos se vuelven más útiles al interpretarlos, por ejemplo, mediante la correlación o la comparación. Esos datos útiles ahora son información. Cuando se la aplica o se la comprende, esa información se convierte en conocimiento.

En la comunicación electrónica, los datos se representan como unos y ceros. Estos elementos diferenciados se conocen como bits (o dígitos binarios). Todos los datos electrónicos se almacenan en este formato binario digital. Mientras que los seres humanos interpretan palabras e imágenes, las computadoras interpretan patrones de bits.

La ventaja de utilizar una codificación digital es que los datos se pueden almacenar de manera más eficaz y pueden transmitirse por grandes distancias sin que se deteriore la calidad.

En conclusión los datos representan la información que generan las personas y los objetos. Cuando se combinan estos datos con el análisis, se proporciona información útil a las personas y a las máquinas. Se toman mejores decisiones y se obtienen mejores resultados.

#### *a) Administración de datos*

Por lo general, las computadoras no tienen la conciencia contextual y la intuición de los seres humanos. Por lo tanto, es importante considerar los dos estados de datos siguientes: estructurados y no estructurados.

**b) Datos estructurados**

Los datos estructurados son aquellos que se introducen y se mantienen en campos fijos dentro de un archivo o un registro. Las computadoras introducen, clasifican, consultan y analizan datos estructurados con facilidad. Por ejemplo, cuando envía su nombre, dirección y datos de facturación a un sitio web, crea datos estructurados. La estructura obliga al uso de cierto formato para la introducción de los datos, a fin de minimizar los errores y hacer que sea más fácil para la computadora interpretarlos.

**c) Datos no estructurados**

Los datos no estructurados carecen de la organización de los datos estructurados; son datos sin procesar. No tienen la estructura que identifica el valor de los datos. No hay un método fijo para introducir o agrupar los datos no estructurados y, luego, analizarlos. Algunos ejemplos de datos no estructurados incluyen el contenido de fotos y archivos de audio y de video.

Los datos estructurados y no estructurados son recursos valiosos para las personas, las organizaciones, las industrias y los gobiernos. Al igual que otros recursos, la información recolectada a partir de datos estructurados y no estructurados tiene un valor medible. Sin embargo, el valor de esos datos puede aumentar o disminuir, según la forma en que se administran. Incluso los mejores datos pierden valor con el tiempo.

Es importante que las organizaciones tomen en cuenta todos los tipos de datos (estructurados, no estructurados y semiestructurados) y determinen cómo darles formato para que puedan administrarse y analizarse.

Para comprender la administración de datos, es importante entender conceptos como el almacenamiento de datos y el transporte de datos.

#### *d) Almacenamiento de datos*

Al referirnos al espacio de almacenamiento, utilizamos el término bytes (B). Un solo byte es una combinación de 8 bits.

Con los años, la cantidad de espacio de almacenamiento disponible aumentó exponencialmente. Por ejemplo, hasta no hace mucho tiempo, el espacio de almacenamiento de los discos duros solía medirse en megabytes. En la actualidad, los discos duros de un terabyte son comunes.

Existen tres tipos principales de almacenamiento de datos:

*Datos locales:* datos a los que se accede en forma directa mediante dispositivos locales. Los discos duros, las unidades flash USB y los CD o DVD son ejemplos de almacenamiento local de datos.

*Datos centralizados:* datos que se almacenan en un único servidor centralizado y se comparten desde allí. Diversos dispositivos pueden acceder a la información de forma remota mediante la red o Internet. Usar un servidor de datos centralizados puede tener como resultado cuellos de botella e ineficiencia, y puede convertirse en un único punto de error.

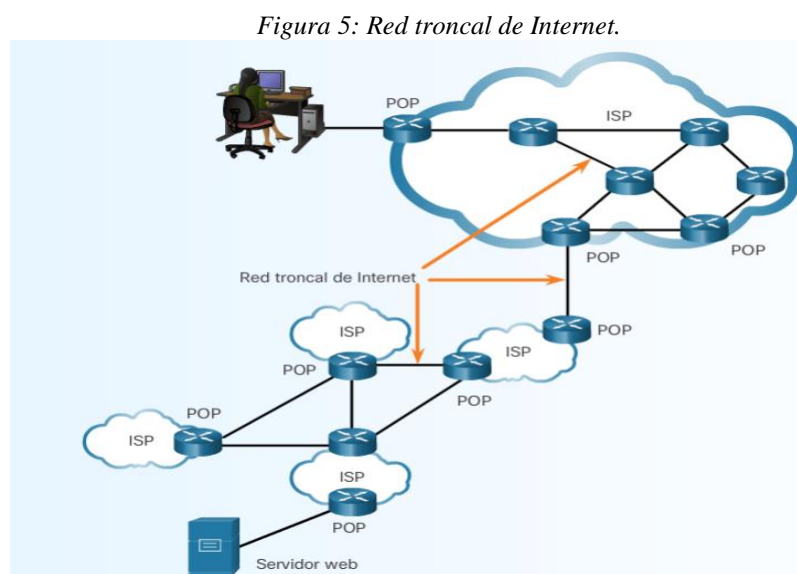
*Datos distribuidos:* datos administrados por un sistema de administración de base de datos central (DBMS). Los datos distribuidos son datos que se reproducen y almacenan en varias ubicaciones. Esto permite un uso compartido de datos eficaz y sencillo. Para obtener acceso a los datos distribuidos, se utilizan aplicaciones locales y globales. Con un sistema distribuido, no hay un único origen de error. Si hay un corte de energía en uno de los sitios, los usuarios pueden seguir teniendo acceso a los datos de los otros sitios.



### e) Transporte de datos

#### i. Proveedores de servicios de Internet

En los entornos de almacenamiento centralizado y distribuido de datos, estos deben transportarse a través de la red o Internet. Los dispositivos que reenvían datos a través de Internet deben utilizar un proveedor de servicios de Internet (ISP). Los ISP proporcionan las conexiones para darles acceso a Internet a personas y empresas, y también pueden interconectarse a otros ISP. Las redes se conectan a un ISP en un punto de presencia (POP). En un ISP, una red de routers y switches de alta velocidad transmite los datos entre los distintos POP. Varios enlaces interconectan los POP para proporcionar rutas alternativas para los datos por si acaso se produce un error en un enlace o se sobrecarga de tráfico. Para enviar información más allá de los límites de una red ISP, los paquetes se reenvían a otros ISP. Como se muestra en la figura n° 5, Internet consta de enlaces de datos de alta velocidad que interconectan varios ISP. Estas interconexiones forman parte de una red muy grande, de gran capacidad, conocida como “red troncal de Internet”.



Fuente: [www.cisco.com](http://www.cisco.com) (2017)

## *ii. Direccionamiento IP*

Los paquetes que atraviesan Internet deben ser paquetes de protocolo de Internet (IP). Cada paquete IP debe contener direcciones IP de origen y de destino válidas. Si no hay información de dirección válida, los paquetes no llegan al host de destino, y los paquetes devueltos no regresan al origen inicial. El protocolo IP define la estructura de las direcciones IP de origen y de destino. Especifica la forma en que estas direcciones se utilizan en el routing de paquetes de un host a otro o de una red a otra.

Actualmente, Internet utiliza IPv4 (versión 4 de IP), pero se está haciendo la transición a IPv6 (versión 6 de IP). IPv6 permite un mayor acceso y escalabilidad con más direcciones IP disponibles y otras funciones.

La dirección IP es similar a la dirección postal de una persona. Se conoce como “dirección lógica” porque se asigna de manera lógica según la ubicación del host. Este proceso es similar a la asignación de las direcciones por parte del gobierno local según la descripción lógica de la ciudad, el pueblo o el barrio. No sería posible recordar todas las direcciones IP de todos los servidores que prestan servicios de hospedaje en Internet. Por eso, existe una manera más sencilla de ubicar servidores mediante la asociación de un nombre con una dirección IP.

## *iii. Paquetes IP*

Cuando se juega un videojuego en Internet, se chatea con un amigo, se envía un correo electrónico o se navega por la Web, los datos enviados y recibidos se transmiten en forma de paquetes IP. Antes de enviarse por Internet, los datos se dividen en paquetes IP. Para redes Ethernet, el tamaño del paquete es de entre 64 y 1500 bytes. Descargar una sola canción de 3 MB requiere más de 2000 paquetes de 1500 bytes cada uno.

En las redes, cada byte de datos se transmite de a un bit por vez. El ancho de banda de red, o la velocidad de transferencia de datos, se expresa en bits por segundo. Por ejemplo, una conexión de un megabit (1 000 000 bits) significa que, en teoría, los datos pueden transmitirse a un megabit por segundo (1 Mb/s).

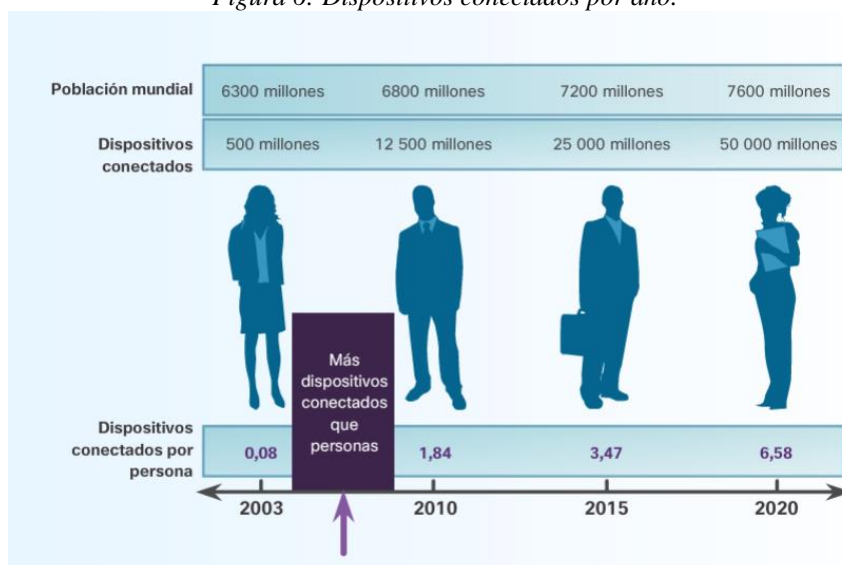
#### *iv. Administración de direcciones IP*

En Internet, cada dirección IP debe ser única. La Autoridad de números asignados de Internet (IANA) es responsable de controlar la distribución de direcciones IP de modo que no haya duplicaciones. La IANA asigna bloques de direcciones IP a cada uno de los cinco registros regionales de Internet (RIR). Los ISP obtienen bloques de direcciones IP del RIR en su región geográfica. Es responsabilidad del ISP administrar esas direcciones y asignarlas a las redes de sus clientes y a los dispositivos y redes de los usuarios finales.

El ISP determina adónde debe reenviar el tráfico. Los paquetes pasan de un router a otro, a través de varias redes ISP hasta que llegan a su destino final. Los routers en cada uno de los ISP utilizan la dirección de destino de los paquetes IP para elegir la mejor ruta a través de Internet. El switching de paquetes es transparente para el usuario, que solo ve qué se envió y qué se recibió.

El volumen de datos que, hace una década, se producía en un año, ahora se produce en una semana. Eso equivale a la producción de más de 20 exabytes de datos por semana. A medida que más objetos no conectados se conectan, los datos siguen creciendo exponencialmente, como referencia en la fig. n° 6.

Figura 6: Dispositivos conectados por año.

Fuente: [www.cisco.com](http://www.cisco.com) (2017)

#### v. Datos en movimiento

En general, los datos se consideran información que se recolectó con el tiempo, estos datos almacenados de manera histórica se denominan “datos en reposo”, sin embargo, a medida que continúa el crecimiento acelerado de grandes cantidades de datos, una gran parte del valor de estos se pierde casi con la misma rapidez con la que se crea, a esta generación de nuevos datos de manera continua se denomina “datos en movimiento”.

Esta nueva forma de ver a los datos, permite aportar nuevas soluciones a nuestro mundo, en diversos ámbitos como la salud, la educación, etc., desde este enfoque Internet de Todo será quien se encargue de estos “datos en movimiento”.

#### vi. Administración de datos masivos

Uno de los factores que impulsan el crecimiento de la información es la cantidad de dispositivos conectados a Internet y la cantidad de conexiones entre esos dispositivos. Pero esto es solo el principio.

Nuevos dispositivos se conectan a Internet a diario y generan una gran cantidad de contenido nuevo. Con esta cantidad de información, las organizaciones deben aprender cómo administrar los datos y también cómo administrar los “datos masivos”.

Los datos masivos tienen tres dimensiones principales que es necesario explicar: volumen, variedad y velocidad. El volumen describe la cantidad de datos que se transporta y se almacena, la variedad describe el tipo de datos, y la velocidad describe la rapidez a la que estos datos se transmiten. Los datos no pueden moverse sin una infraestructura. La velocidad de la infraestructura (entrada/salida, ancho de banda y latencia) y la capacidad de habilitar rápidamente recursos óptimos (red, CPU, memoria y almacenamiento) afecta en forma directa la velocidad de los datos.

#### *vii. Análisis de datos masivos*

Las aplicaciones de datos masivos reciben información de una gran diversidad de orígenes de datos, entre ellos, computadoras, smartphones, tablet PC, máquinas, sensores, medios sociales y aplicaciones multimedia, de esta manera el análisis de datos masivos pueden ayudar a identificar tendencias, predecir comportamientos y dar poder a los responsables de la toma de decisiones y tomar en cuenta lo siguiente:

- La cantidad de datos que se generan
- La forma en que se identifican y administran estos datos como un activo de la organización
- La forma en que estos datos se convierten en información utilizable
- La forma en que las organizaciones utilizan estos datos para tomar decisiones

Las aplicaciones de datos masivos deben poder recolectar estos datos y estructurarlos de forma que puedan generar valor para las organizaciones. Por ejemplo, las aplicaciones de datos masivos deben poder adecuarse a tendencias y orígenes de datos cambiantes, como las siguientes:

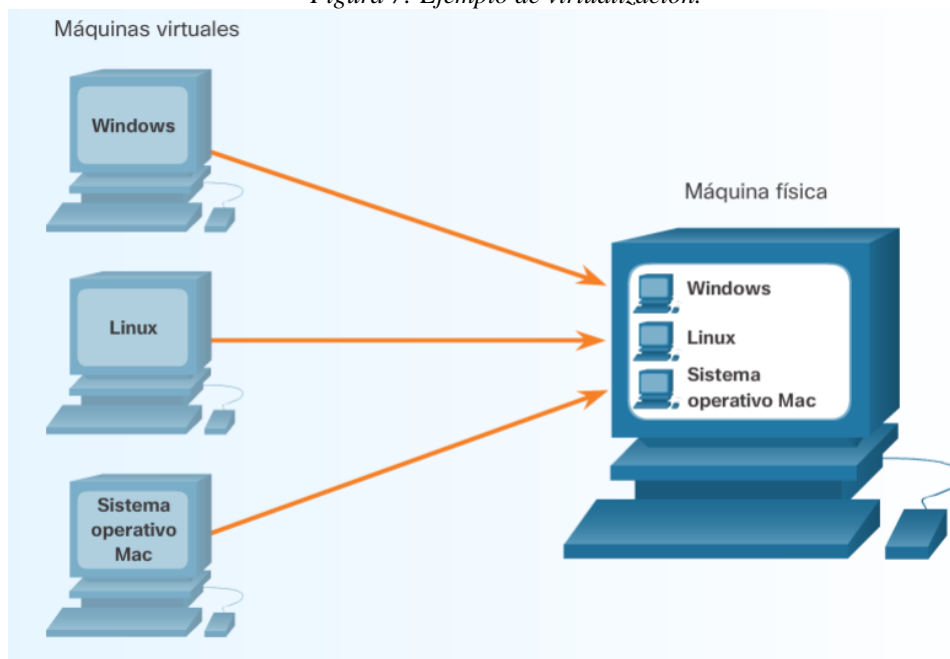
- **Movilidad:** dispositivos móviles, eventos, uso compartido e integración de sensores
- **Acceso a los datos y consumo de datos:** Internet, sistemas interconectados, redes sociales y modelos de acceso
- **Capacidades del ecosistema:** cambios importantes en el modelo de procesamiento de información y la disponibilidad de un marco de código abierto

Como resultado, el costo y la complejidad de estos modelos aumentaron, lo que produjo cambios en la manera en que se almacenan y analizan los datos masivos, dando paso a la virtualización y computación en la nube.

### *viii. Virtualización*

Técnicamente hablando la virtualización es la emulación de equipos virtuales en una sola computadora física, de manera que se tiene varias infraestructuras virtuales con su propio sistema operativo en una misma infraestructura física, como se ve en la fig. n° 7, en el mundo empresarial esto se traduce en reducción de costos operativos y administrativos, así como las necesidades de energía y refrigeración, también sirve para poder realizar pruebas, si ocurre algún problema simplemente se elimina la máquina virtual.

Figura 7: Ejemplo de virtualización.



Fuente: [www.cisco.com](http://www.cisco.com) (2017)

#### ***ix. Computación en la nube***

La computación en la nube es otra forma de administrar y almacenar datos y obtener acceso a ellos, permitiendo al usuario su acceso en cualquier momento y cualquier lugar.

La computación en la nube implica una gran cantidad de computadoras conectadas a través de una red. Los proveedores de computación en la nube dependen en gran medida de la virtualización para ofrecer sus servicios. También puede reducir los costos operativos mediante un uso más eficiente de los recursos.

### **2.5.3.3 Personas**

En la actualidad, la mayoría de las personas se conectan en forma social a través de los dispositivos con conexión a Internet. A medida que IdT evolucione, nos conectaremos de maneras nuevas y valiosas. La ropa y los dispositivos que se llevan puestos ya están cambiando la forma en que nos conectamos.

Los datos, por sí solos, no sirven de nada. Una gran cantidad de datos a la que nadie puede tener acceso no resulta útil. La organización de estos datos y su transformación en información útil permite que las personas tomen decisiones con buenos fundamentos e implementen las medidas adecuadas, permitiendo generar un valor económico basado en Internet de Todo, de esta manera a las personas de les considera unos de los 4 pilares.

“Esto [IdT] no tiene que ver con la tecnología, sino con la manera en que cambiamos la vida de las personas”. (John Chambers, director ejecutivo de Cisco Systems, 2015)

### **2.5.3.4 Procesos**

Los procesos desempeñan una función fundamental en la manera en que los otros pilares operan juntos para ofrecer valor en el mundo conectado de IdT, con el proceso adecuado, las conexiones adquieren pertinencia y agregan valor, dado que se entrega la información correcta a la persona indicada en el momento adecuado y de la manera apropiada.

## **2.5.4 Conectando los pilares**

### **2.5.4.1 Conexiones M2M**

Las conexiones de máquina a máquina (M2M) tienen lugar cuando se transfieren datos de una máquina u “objeto” a otro a través de una red. Las máquinas incluyen sensores,



robots, computadoras y dispositivos móviles. Estas conexiones M2M a menudo se denominan “Internet de las cosas”.

Un ejemplo de M2M es un automóvil conectado que emite una señal para informar que un conductor ya casi llega a casa, lo que le indica a la red doméstica que ajuste la temperatura y la iluminación del hogar.

#### **2.5.4.2 Conexiones M2P**

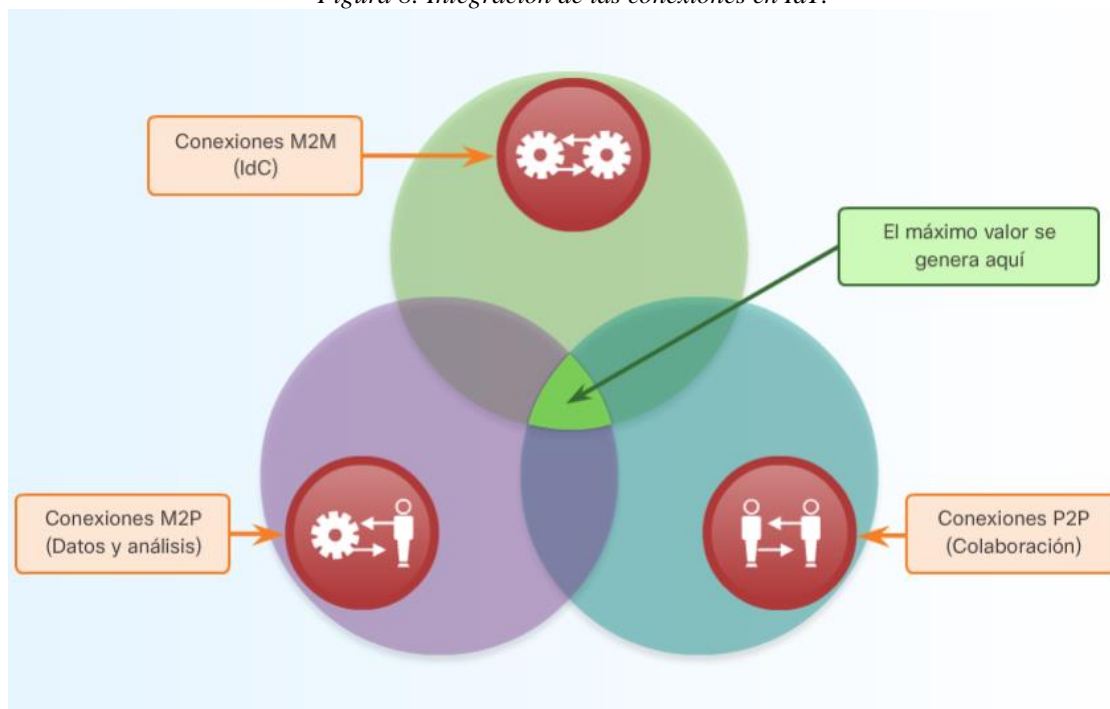
Las conexiones de máquina a persona (M2P) tienen lugar cuando la información se transfiere entre una máquina (como una computadora, un dispositivo móvil o un letrero digital) y una persona. Cuando una persona obtiene información de una base de datos o realiza un análisis complejo, tiene lugar una conexión M2P. Estas conexiones M2P facilitan el movimiento, la manipulación y la información de datos de máquinas para ayudar a las personas a que tomen decisiones fundadas. Las acciones que las personas realizan según sus razonamientos fundados completan un ciclo de realimentación de IdT.

#### **2.5.4.3 Conexiones P2P**

Las conexiones de persona a persona (P2P) tienen lugar cuando la información se transfiere de una persona a otra. Las conexiones P2P se producen cada vez más a través de video, dispositivos móviles y redes sociales. Con frecuencia, estas conexiones P2P se denominan “colaboración”.

Como se muestra en la fig. n° 8, el valor más alto de IdT se obtiene cuando el proceso facilita la integración de las conexiones M2M, M2P y P2P.

Figura 8: Integración de las conexiones en IdT.



Fuente: [www.cisco.com](http://www.cisco.com) (2017)

#### 2.5.4.4 Interacciones de IdT

Las interacciones entre los elementos de los cuatro pilares crean una riqueza de nueva información. Los pilares interactúan de forma tal que establecen tres conexiones principales en el entorno de IdT: personas que se comunican con personas (P2P), máquinas que se comunican con personas (M2P) y máquinas que se comunican con máquinas (M2M).

Estas interacciones en IdT nos dan los siguientes atributos:

- Hiperconciencia: los sensores pueden captar datos en tiempo real sobre productos.
- Capacidad para predecir: los nuevos tipos de herramientas de análisis de datos permiten que una organización estime las tendencias y comportamientos futuros.

- Agilidad: las predicciones, que son cada vez más precisas, permiten que las organizaciones sean más receptivas y flexibles a las tendencias de mercado emergentes.

Si se combinan estos tres atributos, los agricultores, organizaciones y/o empresas pueden crear, comunicar y entregar sus ofertas de mejor forma, como se muestra en la tabla n° 4.

*Tabla n° 4: Atributos de las interacciones de IdT.*

Hiperconsciente	Predictivo	Ágil
<ul style="list-style-type: none"> <li>• Detectar la ubicación, el estado y el contacto de los clientes y recursos de la empresa.</li> <li>• Supervisar la conducta y las opiniones de los clientes en tiempo real.</li> <li>• Identificar los cambios competitivos y de mercado.</li> </ul>	<ul style="list-style-type: none"> <li>• Anticiparse a las transiciones del mercado.</li> <li>• Optimizar el rendimiento de los recursos y las operaciones.</li> <li>• Prever y abordar de manera proactiva las amenazas de seguridad emergentes.</li> </ul>	<ul style="list-style-type: none"> <li>• Lograr la diferenciación competitiva con una respuesta más rápida que la de los rivales.</li> <li>• Promover la innovación mediante la creación de “plataformas” para lograr una ventaja sostenible.</li> <li>• Responder a las amenazas de rápida evolución</li> </ul>

*Fuente: CISCO*

### 2.5.5 La red

Las redes componen la base de IdT. Hay redes de todo tamaño. Pueden ir desde redes simples, compuestas por dos computadoras, hasta redes que conectan millones de dispositivos.

Las redes en definición sencilla nos permiten conectarnos en diferentes niveles, permitiéndonos el acceso a datos, almacenamiento, procesos, etc.

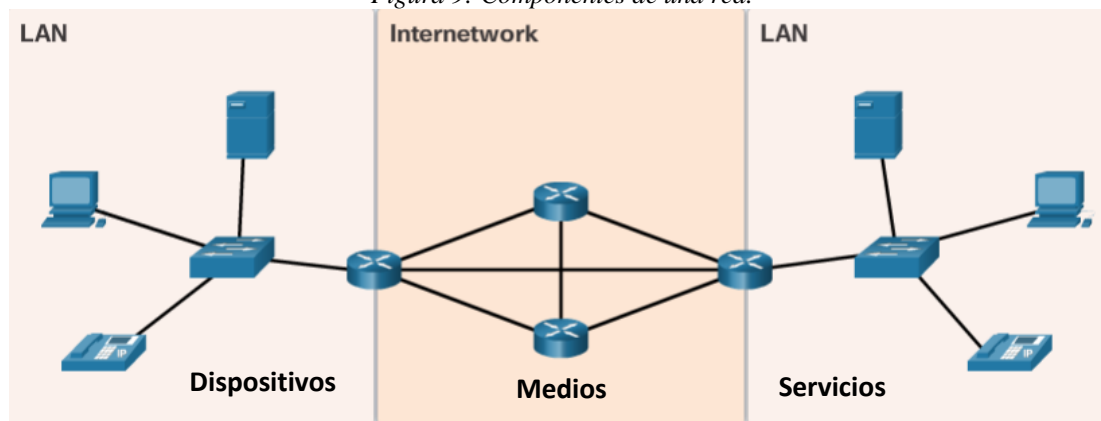
Internet es la red más extensa que existe. De hecho, el término Internet significa “red de redes”. Internet es, literalmente, una colección de redes privadas y públicas interconectadas. Por lo general, las redes de empresas, de oficinas pequeñas e incluso las redes domésticas proporcionan una conexión a Internet compartida.

#### *a) Componentes de las redes*

La ruta que toma un mensaje de origen a destino puede ser tan sencilla como un solo cable que conecta una computadora a otra o tan compleja como una red que literalmente abarca el mundo. Esta infraestructura de red es la plataforma que da soporte a la red. Proporciona el canal estable y confiable por el cual se producen las comunicaciones, como se muestra en la fig. n° 9.

Los dispositivos y los medios son los elementos físicos, o el hardware, de la red. Por lo general, el hardware está compuesto por los componentes visibles de la plataforma de red, como una computadora portátil, una PC, un switch, un router, un punto de acceso inalámbrico o el cableado que se utiliza para conectar esos dispositivos. A veces, puede que algunos componentes no sean visibles. En el caso de los medios inalámbricos, los mensajes se transmiten a través del aire mediante radio frecuencias invisibles u ondas infrarrojas.

Figura 9: Componentes de una red.



Elaboración: propia.

### b) *Dispositivos finales*

Los dispositivos de red con los que la gente está más familiarizada se denominan “dispositivos finales”. Todas las computadoras conectadas a una red que participan directamente en las comunicaciones de la red se clasifican como hosts. Estos dispositivos forman la interfaz entre los usuarios y la red de comunicación subyacente.

Los que siguen son algunos ejemplos de dispositivos finales:

- Computadoras (estaciones de trabajo, computadoras portátiles, servidores de archivos y servidores web)
- Impresoras de red
- Teléfonos VoIP
- Cámaras de seguridad
- Dispositivos portátiles móviles (smartphones, tablet PC, PDA y lectores inalámbricos de tarjetas de débito y crédito, y escáneres de códigos de barras)
- Sensores, como termómetros, balanzas y otros dispositivos que se conectarán a IdT

Los dispositivos finales son el origen o el destino de los datos transmitidos a través de la red. Para distinguir un dispositivo final de otro, cada dispositivo final en la red se identifica por una dirección. Cuando un dispositivo final inicia una comunicación, utiliza la dirección del dispositivo final de destino para especificar adónde se debe enviar el mensaje.

*c) Dispositivos de red intermediarios*

Los dispositivos intermediarios interconectan dispositivos finales. Estos dispositivos proporcionan conectividad y operan detrás de escena para asegurar que los datos fluyan a través de la red. Los dispositivos intermediarios conectan los hosts individuales a la red y pueden conectar varias redes individuales para formar una internetwork. Los siguientes son ejemplos de dispositivos de red intermediarios:

- Switches y puntos de acceso inalámbrico (acceso a la red)
- Routers (internetworking)
- Firewalls (seguridad)

La administración de datos a medida que fluyen en la red es también una función de los dispositivos intermediarios, otras funciones son las siguientes:

- Volver a generar y transmitir las señales de datos.
- Conservar información acerca de las rutas que existen a través de la red y de internetwork.
- Notificar a otros dispositivos los errores y las fallas de comunicación.
- Dirigir los datos por rutas alternativas cuando hay una falla en el enlace.
- Clasificar y dirigir los mensajes según las prioridades de calidad de servicio (QoS).
- Permitir o denegar el flujo de datos de acuerdo con la configuración de seguridad.

#### *d) Medios de red*

La comunicación a través de la red se transmite por un medio, como un cable o por aire. El medio facilita la comunicación de origen a destino.

Las redes modernas utilizan principalmente tres tipos de medios para interconectar los dispositivos y proporcionar la ruta por la cual pueden transmitirse los datos, estos medios son los siguientes:

- Hilos metálicos dentro de cables
- Fibras de vidrio o plástico (cable de fibra óptica)
- Transmisión inalámbrica

La codificación de la señal que se debe realizar para que se transmita el mensaje es diferente para cada tipo de medio. En los hilos metálicos, los datos se codifican en impulsos eléctricos que coinciden con patrones específicos. Las transmisiones por fibra óptica dependen de impulsos de luz, en intervalos de luz visible o infrarroja. En las transmisiones inalámbricas, los patrones de ondas electromagnéticas muestran los distintos valores de bits.

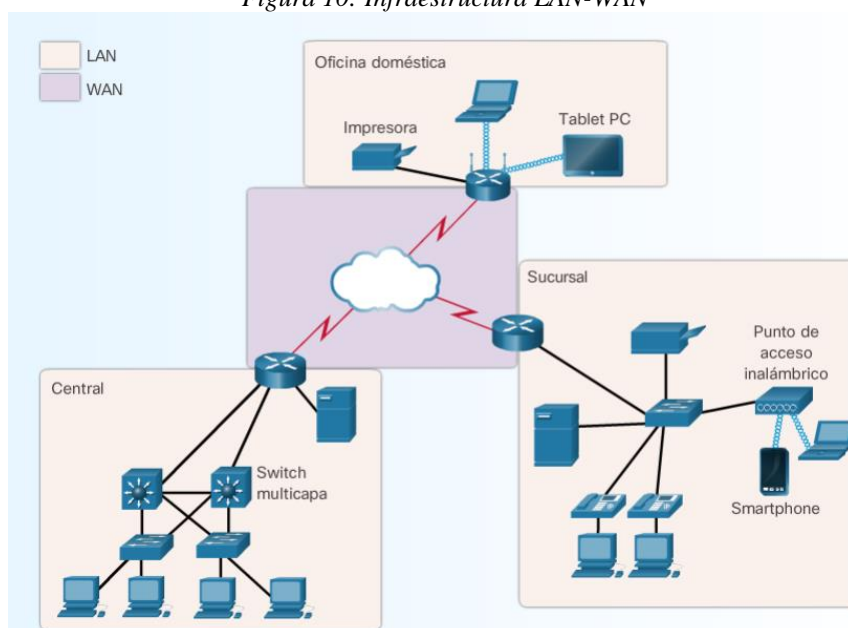
#### *e) Tipos de redes*

En la fig. n° 10 se muestran dos de los tipos de infraestructuras de red más comunes:

- **Red de área local (LAN):** una infraestructura de red que proporciona acceso a usuarios y dispositivos finales en un área limitada, como un hogar, una escuela, un edificio de oficinas o un campus. Proporciona un ancho de banda de alta velocidad a los dispositivos finales internos y a los dispositivos intermediarios.

- **Red de área extensa (WAN):** una infraestructura de red que interconecta las LAN a través de áreas geográficas amplias, como ciudades, estados, provincias, países y continentes. Las WAN suelen pertenecer a una organización autónoma, como una empresa o un gobierno. Generalmente, las WAN proporcionan velocidades de enlace entre redes LAN que son más bajas que las velocidades de enlace dentro de una LAN.

Figura 10: Infraestructura LAN-WAN



Fuente: [www.cisco.com](http://www.cisco.com) (2017)

### f) Internet

Gracias a Internet nos podemos conectar al mundo, ya que si bien utilizar una LAN o una WAN trae ventajas, la mayoría de las personas necesitan comunicarse con un recurso en otra red. Es posible que esta red esté fuera de la red local. Esta comunicación se logra mediante Internet.

Internet no pertenece a una persona o un grupo. Internet es una colección mundial de redes interconectadas.



### ***g) La red convergente***

Las redes modernas están en constante evolución para satisfacer las demandas de los usuarios. Las primeras redes de datos estaban limitadas a intercambiar información basada en caracteres entre sistemas informáticos conectados. Las redes tradicionales de teléfono, radio y televisión se mantenían separadas de las redes de datos, cada servicio tenía su propio conjunto de reglas y estándares para asegurar una comunicación satisfactoria.

Entonces se puede decir de manera concisa que las redes antiguas estaban separadas mientras que las redes convergentes actuales transmiten todo mediante un mismo canal, esta convergencia presenta la primera fase en el armado de la red de información inteligente que admitirá IdT.

### ***h) Programación***

Ya se mencionó a los actuadores, pero estos necesitan recolectar la información correcta y también transmitir esa información, es entonces donde entra la programación, ya que este define un conjunto de instrucciones para recibir esos datos y decidir que se procesa y que se transmite a otro dispositivo.

### ***i) Definición***

Un programa informático es un conjunto de instrucciones que se le da a una computadora para que se ejecuten en un orden específico, esto se realiza mediante un lenguaje de programación, estos lenguajes permiten que los seres humanos escriban instrucciones de manera que las computadoras puedan entenderlas. Si bien existen varios lenguajes informáticos diferentes, todos ellos se basan en estructuras lógicas.

Las estructuras lógicas más comunes que se encuentran en los lenguajes de programación:

- **IF** *se cumple la condición*, **THEN** *seguir las instrucciones* (If/Then)
- **FOR** *la expresión* **DO** *seguir las instrucciones* (For/Do).
- **WHILE** *se cumple la condición*, **DO** *seguir las instrucciones* (While/Do)

Las condiciones lógicas como estas son los componentes básicos de los programas informáticos.

#### *j) Tipos de programas*

Los distintos programas realizan diferentes tareas. Por ejemplo, hay programas para medir e informar la temperatura, programas que controlan los semáforos y programas que nos permiten interactuar con computadoras y dispositivos.

En ocasiones, una categoría de programa es tan común que recibe su propio nombre. Algunas categorías incluyen lo siguiente:

- **Firmware:** el firmware contiene las instrucciones que el dispositivo sigue durante el arranque.
- **Sistemas operativos:** son programas que se escriben para permitir que las personas interactúen con una computadora.
- **Aplicaciones:** son programas diseñados y escritos para realizar una tarea o un servicio específico.

### ***k) Transición a IdT***

IdC se centra en conectar lo desconectado, principalmente los “objetos” de IdT, este proceso requiere una convergencia entre los sistemas de tecnología operativa (TO) y de tecnología de la información (TI) que se implementan.

La TO se define como el control industrial y la infraestructura de automatización de una organización. Esto incluye el hardware (como sensores y dispositivos finales) y el software utilizado para controlar y supervisar los equipos y los procesos de fabricación. La mayor parte de la comunicación en la TO se lleva a cabo entre máquinas.

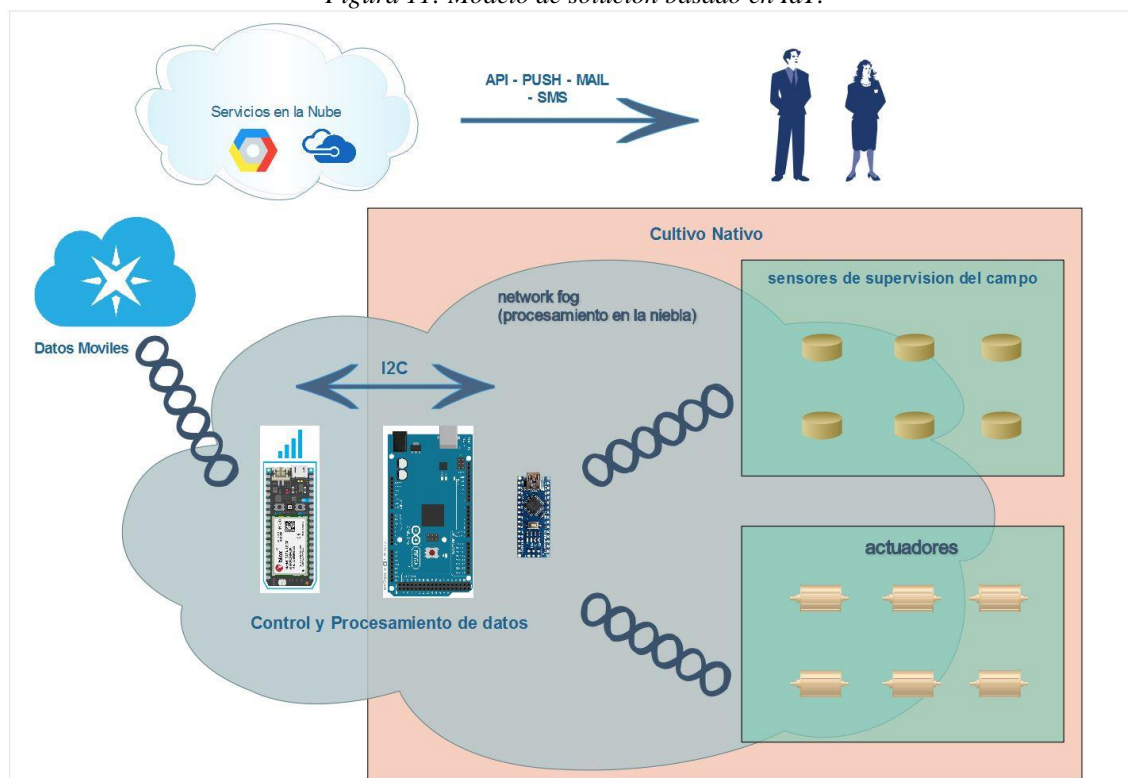
Los sistemas de TI se refieren a la infraestructura de red, las telecomunicaciones y las aplicaciones de software utilizadas para procesar la información y permitir el intercambio de esa información entre las personas.

Para realizar una transición a IdT, es necesario comprender adecuadamente las interacciones entre los pilares, es decir: M2M, M2P, P2P, y de esta manera los sistemas, empresas, organizaciones, etc., podrán hacer una buena transición y así también entender lo necesario para dicha transición.

### ***l) Creación de un modelo para una solución mediante IdT***

Para poder realizar el uso de la transición de IdT, es necesario crear un modelo de solución basado en IdT de tal manera que satisfaga las necesidades del problema, en el presente proyecto basado en cultivos nativos (en el caso de nuestra región tomando el ejemplo de la quinua), se plantea el siguiente modelo como se muestra en la fig. n° 11.

Figura 11: Modelo de solución basado en IdT.



*Elaboración: propia.*

## 2.6 Redes Fog

### 2.6.1 Concepto

Redes Fog en su traducción del inglés también llamadas redes en la niebla o computación en la niebla.

La niebla no deja de ser una extensión de lo que conocemos como la nube, solo que con algunas características que la hacen diferente. En este sentido, es bastante acertada la comparación meteorológica. Así, la nube es algo lejano para el usuario, que está en el cielo y muchas veces no alcanza. En computación serían esos centros de datos con miles de servidores que no sabemos muy bien dónde están. Por el contrario, el Fog computing, la niebla es algo que nos envuelve, nos rodea, que incluso podemos tocar con las manos.

El Internet de las cosas (IoT) es en parte responsable de este cambio en la tendencia de la computación en la nube. Tenemos que pensar que los miles de millones de sensores encargados de transmitir los datos provenientes de cada vez más dispositivos provocarán una necesidad cada vez mayor de ancho de banda, pero también de capacidad de cómputo de los centros de datos, que tienen que manejar esa cantidad ingente de información. Por eso, muchos ven el fog computing el futuro de la tecnología en la nube, como una forma más racional y eficiente de trabajar con todos estos datos generados, bajo este mismo concepto también se evitan latencias innecesarias.

Y no solo debemos asociar a este tipo de computación en un entorno doméstico, sino que es una tecnología que se puede utilizar en la recogida de datos e información de una ciudad inteligente, la creación de redes de datos para vehículos conectados, que no dependan en exclusiva de la nube, ya que un fallo de comunicaciones en ambos casos podría producir un importante colapso. Si lo pensamos detenidamente, el IdT no ha hecho más que dar sus primeros pasos. El aumento de la información generada no hará sino crecer de forma exponencial en los próximos años.

### **2.6.2. Algunos inconvenientes del Fog Computing**

Si uno de los puntos fuertes de la nube es la seguridad y un mayor control sobre los datos, en el fog computing, tener el procesamiento y almacenamiento de parte de ellos en el punto de origen genera algunas dudas, como la posibilidad de poder manipular el código ya que estos dispositivos son más cercanos a las personas.

Otro aspecto a tener en cuenta es que solo se transmite un resumen de la información, por lo tanto se reduce la visibilidad de lo que realmente sucede. Sería como si de una encuesta solo tenemos el resumen final, pero no los datos en bruto, no podemos

saber si el proceso de cálculo ha sido correcto o no. Con todo el modelo tal y como está planteado, parece la única manera de procesar y transmitir la cantidad de datos que se van a generar en los próximos años.

### **2.6.3. Controladores en la niebla**

Los sensores obtienen datos y reenvían esa información a los controladores. El controlador puede reenviar la información reunida de los sensores a otros dispositivos en la niebla.

Tomando como ejemplo un sistema de semáforos inteligentes. Los sensores detectan e informan la actividad al controlador. El controlador puede procesar estos datos de manera local y determinar los patrones de tráfico óptimos. Con esta información, el controlador envía señales a los actuadores en los semáforos para ajustar los flujos de tráfico.

Este es un ejemplo de la comunicación M2M. En esta situación, los sensores, los actuadores y el controlador coexisten en la niebla. Es decir, la información no se reenvía más allá de la red local de los dispositivos finales.

## CAPITULO III

### MATERIALES Y METODOS

#### 3.1 El Electron

##### 3.1.1 Configurando el Electron

Antes de hacer todas las configuraciones respectivas, debemos crear una cuenta en <https://login.particle.io/setup> para poder acceder a todos los recursos, como se muestra en la fig. n° 12.

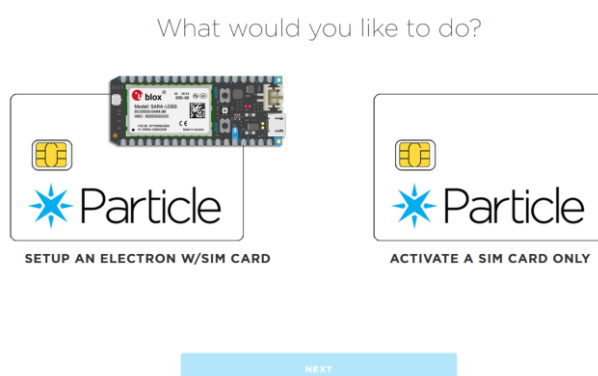
*Figura 12: Login en Particle*



*Fuente: www.particle.io (2017)*

Los primeros pasos para empezar a desarrollar el prototipo, será activar una de las partes principales, el Electron, para lo cual activaremos el chip que trae consigo, el chip es compatible con redes 2G/3G de américa, este chip usara un determinado operador local, en nuestro caso utilizara la red de Movistar, a continuación se mostrara el procedimientos de activación del chip para que se registre en la Particle Cloud, para el presente proyecto por ser la primera vez utilizando el dispositivo se utilizara la primera opción “setup an Electron w/sim card”, debido a que tenemos que configurar el Electron y activar el chip, de tener ya un dispositivo se puede comprar más chips y usar la segunda opción, luego de presionar la primera opción le daremos a “next”, como se muestra en la fig. n° 13.

*Figura 13: Activación del Electron.*



*Fuente: [www.particle.io](http://www.particle.io) (2017)*

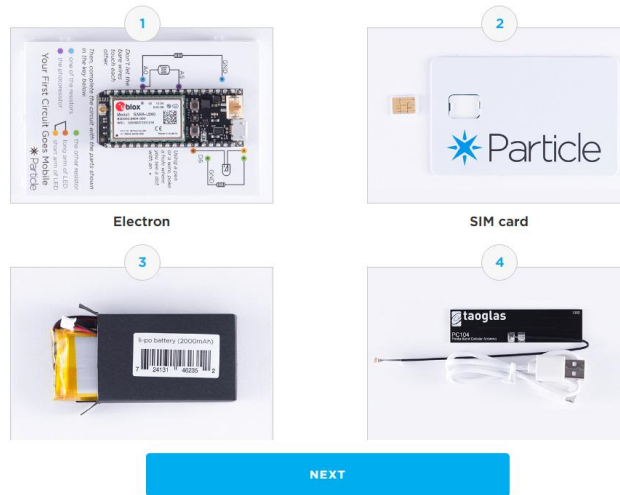
A continuación nos mostrara la siguiente pantalla donde nos indica que debemos de tener en el Electron para su primera configuración, le daremos a next, mostrado en la fig. n°14.



Figura 14: Set up del Electron.

Let's set up your **Electron!**

Before going any further, please make sure you have the following items readily available:



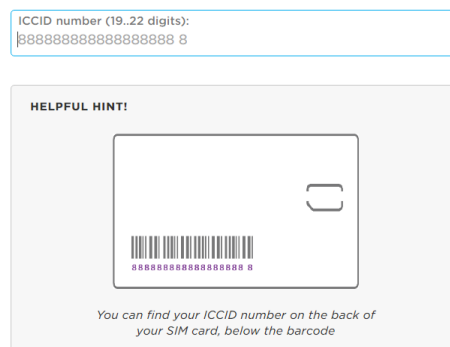
Fuente: [www.particle.io](http://www.particle.io) (2017)

Ahora colocaremos el ICCID que está en la tarjeta del chip y nos dirán que esperemos un momento y el chip estará registrado, véase fig. n° 15.

Figura 15: Número de ICCID.

Enter your SIM's **ICCID number**

Providing the ICCID number will allow you to activate your Particle SIM card



Fuente: [www.particle.io](http://www.particle.io) (2017)

### 3.1.2 Conectando vía USB

Vale la pena señalar aquí que actualmente no se puede configurar un Electron a partir de la línea de comandos (CLI) porque necesitamos que se ingrese un número de tarjeta de crédito, pero el CLI será extremadamente útil para otras cosas.

Para todos los métodos de conexión por USB, el dispositivo debe estar en modo “listening”, donde el LED RGB parpadea en azul, para lo cual debemos mantener presionado el botón MODE.

Se puede realizar la instalación tanto en Windows como en Mac, variando algunos pasos sencillos, en el presente caso usaron una computadora con Windows.

Con la actualización del servicio ahora es posible instalar la CLI de Particle directamente con un instalador para obviar los pasos a continuación.

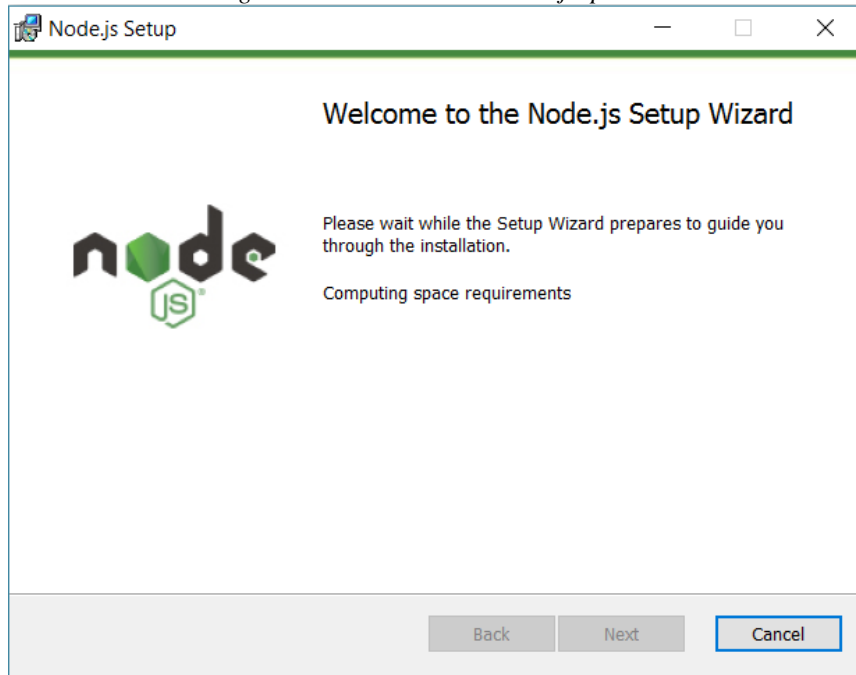
### 3.1.3 Instalando Node.Js

La CLI de Particle se ejecuta con Node.js. Descargamos la última versión del sitio web de Node.js

Iniciamos el instalador y seguimos las instrucciones para instalar node.js, como se muestra en las figuras n° 16, 17 y 18.

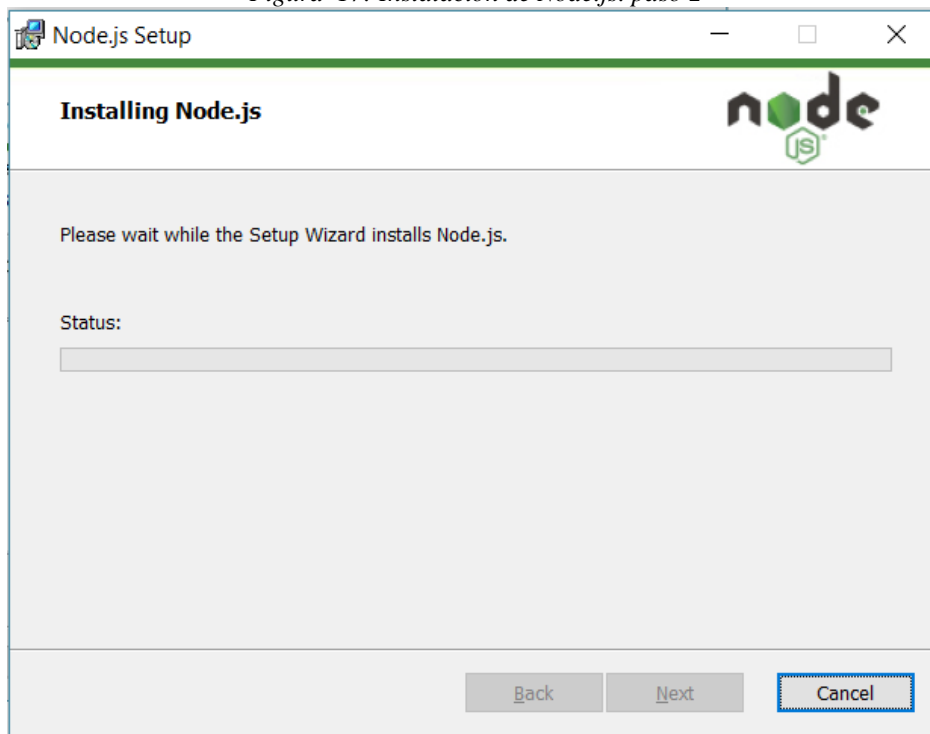
A continuación abrimos el terminal de Microsoft o el programa de terminal preferido.

Figura 16: Instalación de Node.js. paso 1



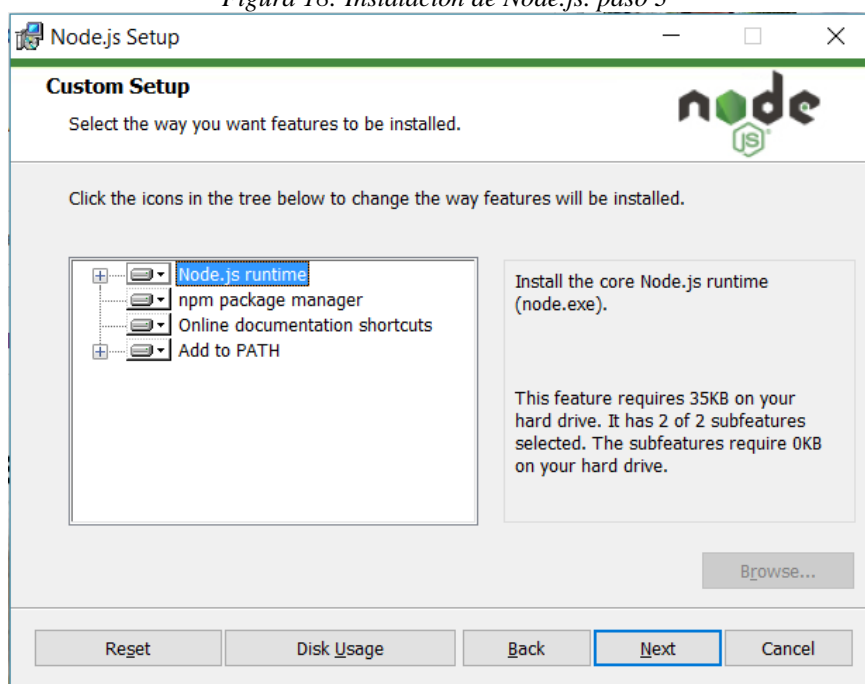
Elaboración: Propia.

Figura 17: Instalación de Node.js. paso 2



Elaboración: Propia.

Figura 18: Instalación de Node.js. paso 3

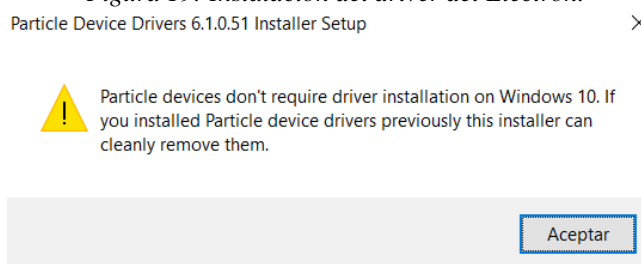


Elaboración: Propia.

### 3.1.4 Instalando el driver de Particle

Descargamos directamente de la página y procedemos a instalar, como se muestra en la fig. n° 19.

Figura 19: Instalación del driver del Electron.

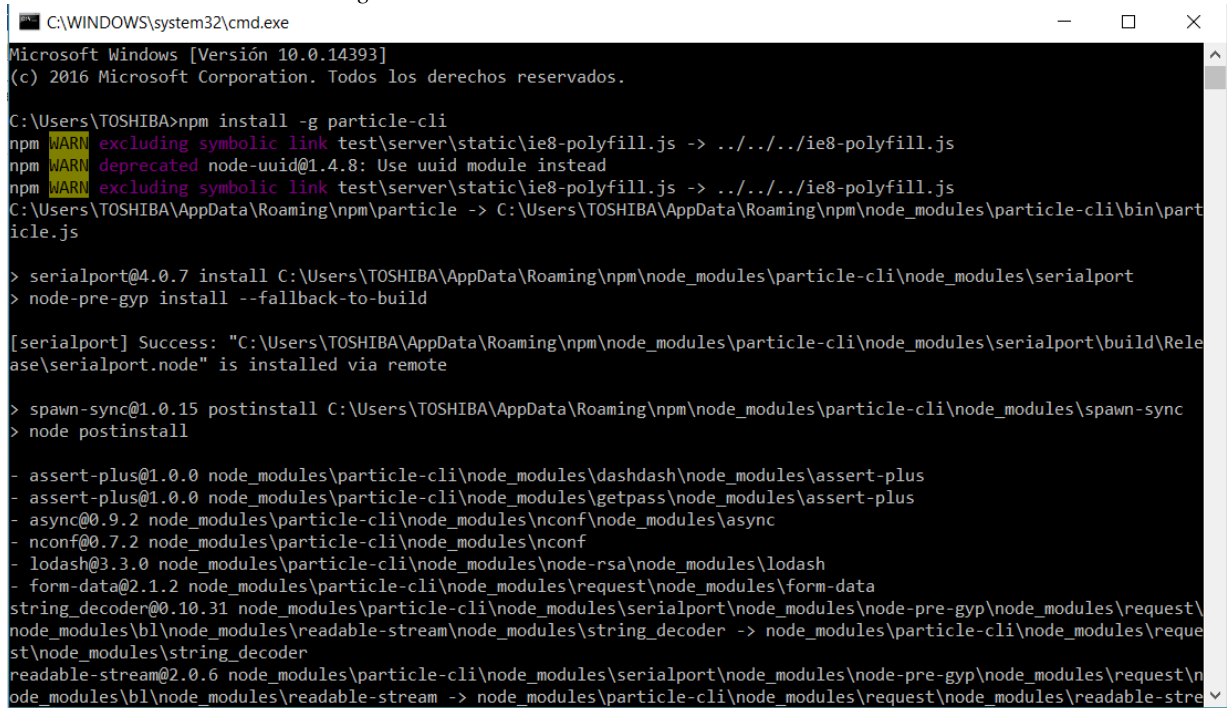


Elaboración: Propia.

Posteriormente para instalar la CLI de Particle abriremos un terminal, pudiendo ser el de Windows o el de Node.js que se instaló. Y colocaremos el comando (véase fig. n° 20)

```
npm install -g particle-cli
```

Figura 20: Instalación de la CLI de Particle.



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.14393]
(c) 2016 Microsoft Corporation. Todos los derechos reservados.

C:\Users\TOSHIBA>npm install -g particle-cli
npm WARN excluding symbolic link test\server\static\ie8-polyfill.js -> ../../../../ie8-polyfill.js
npm WARN deprecated node-uuid@1.4.8: Use uuid module instead
npm WARN excluding symbolic link test\server\static\ie8-polyfill.js -> ../../../../ie8-polyfill.js
C:\Users\TOSHIBA\AppData\Roaming\npm\particle -> C:\Users\TOSHIBA\AppData\Roaming\npm\node_modules\particle-cli\bin\particle.js

> serialport@4.0.7 install C:\Users\TOSHIBA\AppData\Roaming\npm\node_modules\particle-cli\node_modules\serialport
> node-pre-gyp install --fallback-to-build

[serialport] Success: "C:\Users\TOSHIBA\AppData\Roaming\npm\node_modules\particle-cli\node_modules\serialport\build\Release\serialport.node" is installed via remote

> spawn-sync@1.0.15 postinstall C:\Users\TOSHIBA\AppData\Roaming\npm\node_modules\particle-cli\node_modules\spawn-sync
> node postinstall

- assert-plus@1.0.0 node_modules\particle-cli\node_modules\dashdash\node_modules\assert-plus
- assert-plus@1.0.0 node_modules\particle-cli\node_modules\getpass\node_modules\assert-plus
- async@0.9.2 node_modules\particle-cli\node_modules\nconf\node_modules\async
- nconf@0.7.2 node_modules\particle-cli\node_modules\nconf
- lodash@3.3.0 node_modules\particle-cli\node_modules\node-rsa\node_modules\lodash
- form-data@2.1.2 node_modules\particle-cli\node_modules\request\node_modules\form-data
string_decoder@0.10.31 node_modules\particle-cli\node_modules\serialport\node_modules\node-pre-gyp\node_modules\request\node_modules\bl\node_modules\readable-stream\node_modules\string_decoder -> node_modules\particle-cli\node_modules\request\node_modules\string_decoder
readable-stream@2.0.6 node_modules\particle-cli\node_modules\serialport\node_modules\node-pre-gyp\node_modules\request\node_modules\bl\node_modules\readable-stream -> node_modules\particle-cli\node_modules\request\node_modules\readable-stream

```

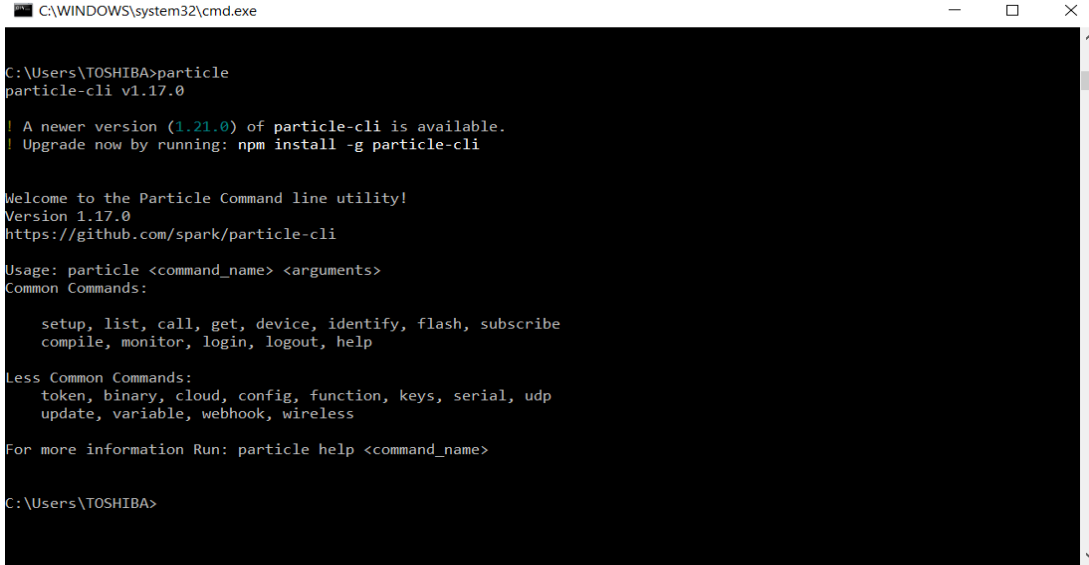
*Elaboración: Propia.*

### 3.1.5 Conectando nuestro dispositivo

En nuestro caso estamos usando el Electron, como se mencionó antes el Electron se configura desde la página web de particle, mientras que los otros dispositivos como el Photon se pueden configurar mediante CLI.

Para lo cual para el resto de configuración usamos la interfaz de línea de comandos que más nos agrade. Abrimos un símbolo del sistema, colocamos el comando Particle (véase fig. n° 21) y nos saldrá lo siguiente:

Figura 21: Comandos de Particle.



```

C:\WINDOWS\system32\cmd.exe
C:\Users\TOSHIBA>particle
particle-cli v1.17.0

! A newer version (1.21.0) of particle-cli is available.
! Upgrade now by running: npm install -g particle-cli

Welcome to the Particle Command line utility!
Version 1.17.0
https://github.com/spark/particle-cli

Usage: particle <command_name> <arguments>
Common Commands:
  setup, list, call, get, device, identify, flash, subscribe
  compile, monitor, login, logout, help

Less Common Commands:
  token, binary, cloud, config, function, keys, serial, udp
  update, variable, webhook, wireless

For more information Run: particle help <command_name>


C:\Users\TOSHIBA>

```

Elaboración: Propia.

Ahora podremos logear nuestra cuenta de desde la CLI, como muestra la fig. n°22.

Figura 22: Login de cuenta mediante la CLI.



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.14393]
(c) 2016 Microsoft Corporation. Todos los derechos reservados.

C:\Users\TOSHIBA>particle login
particle-cli v1.17.0

! A newer version (1.21.0) of particle-cli is available.
! Upgrade now by running: npm install -g particle-cli

? Please enter your email address gianmarcopalacios@gmail.com
? Please enter your password *****
> Successfully completed login!

C:\Users\TOSHIBA>

```

Elaboración: Propia.

### 3.1.6 Modos del dispositivo

Es necesario detenernos un momento para conocer lo diversos modos de trabajo del dispositivo para poder trabajar con fluidez a la hora de hacer los proyectos.

El dispositivo cuenta con los siguientes modos:

a) *Modos normales*

Resumiremos los modos en la siguiente tabla n° 5.

*Tabla n° 5: Modos de funcionamiento del Electron.*

Modo	Característica
Conectado	Color: cian Estado: conectado a internet
OTA Firmware Update	Color: magenta Estado: cargando código o actualizando firmware
Looking for Internet	Color: verde parpadeando Estado: intentando conectarse a Internet.
Cellular off	Color: blanco Estado: el modo celular se apaga
Listening	Color: azul parpadeando Estado: conectándose a una aplicación móvil o por USB
Safe	Color: cian parpadeando Estado: se conecta a la nube pero no ejecuta ningún firmware
DFU (Device Firmware Update)	Color: amarillo Estado: programar un firmware personalizado vía USB

*Fuente: www.particle.io (2017)*

**b) Firmware Reset**

El Electron puede almacenar una copia de seguridad de cualquier firmware de usuario deseado en memoria flash en la dirección 0x080A0000, separado de la memoria flash del usuario que se encuentra en 0x08080000. Esta copia de seguridad del firmware se puede restaurar en la memoria del usuario con una secuencia de botones que sólo está disponible cuando la memoria flash de copia de seguridad contiene una imagen de firmware válida. Para programar el Electron con una copia de seguridad del firmware del usuario a través de USB, necesitaremos poner el Electron en modo DFU y ejecutar el siguiente comando:

```
particle flash --factory user-backup-firmware.bin
```

**c) Factory Reset**

El restablecimiento de fábrica no está disponible en el electrón, pero eso no es un problema, si se experimenta problemas con el firmware de la aplicación, se puede utilizar el Modo a prueba de errores para recuperarse.

**d) Tinker**

Tinker es la aplicación móvil de Particle que hace posible la conexión con el Electron sin escribir ningún código. Es ideal para el desarrollo temprano, el aprendizaje y la creación de prototipos.

Por otro lado Tinker no solo es la aplicación, sino también el código de compilación, con lo cual nosotros podríamos crear una aplicación en base a ese código según nuestras necesidades.



### 3.1.7 Flashear Apps con Particle Build

Particle Build, es una sección donde podremos crear nuestros códigos de compilación, agregarle librerías, guardarlos, etc.

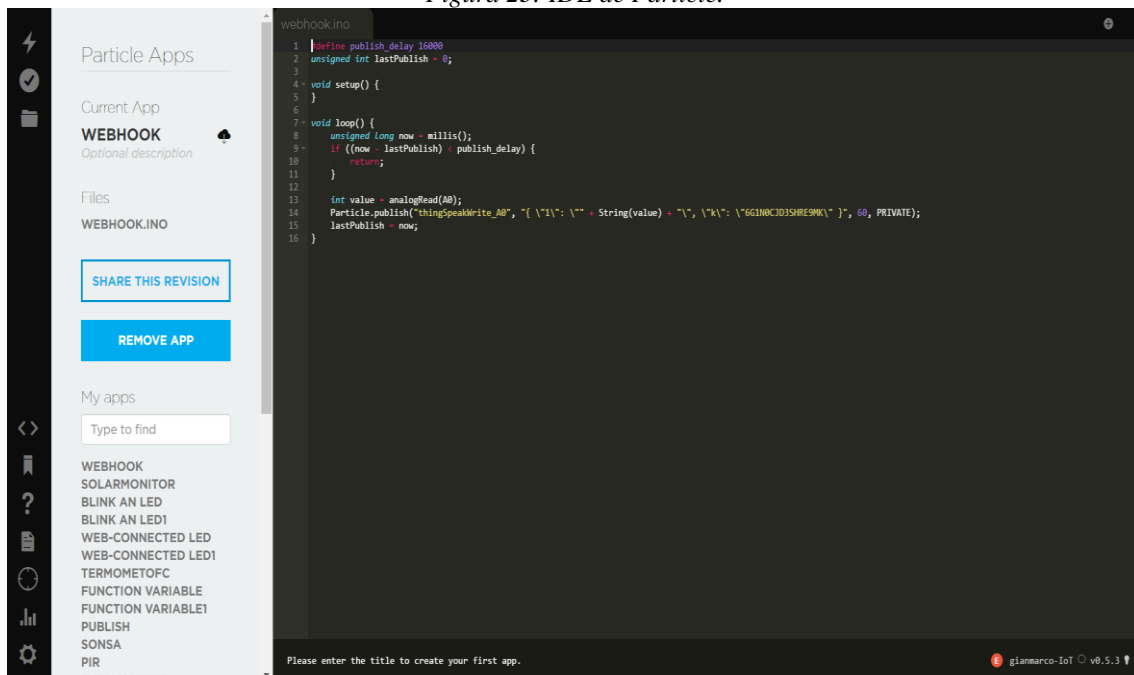
Para lo cual debemos primero logearnos en la página de Particle.

Luego de eso debemos dirigirnos a la sección de programación, al IDE, se debe resaltar que el IDE se hace de forma online, es decir mediante la misma página web de Particle, aunque también existe una versión beta de un programa ejecutable que es el “Particle Dev”, posteriormente se hablara de la herramienta.

#### 3.1.7.1 Web IDE

Particle Build es un Entorno de Desarrollo Integrado, o IDE; Lo que significa que se puede hacer el desarrollo de software de forma sencilla mediante una que se ejecuta en el navegador web (véase fig. n° 23)

Figura 23: IDE de Particle.



Elaboración: Propia.

Particle Build tiene una barra de navegación a la izquierda en la parte superior, donde hay tres botones, que tienen las funciones más importantes:

- **Flash:** Flashea o inserta el código actual en el dispositivo. Esto inicia una actualización de firmware over-the-air o por aire y carga el nuevo software en el dispositivo. Es importante tener en cuenta que hacer una actualización OTA en el electrón usa datos y que en todo caso sería más recomendable hacerlo de manera serial.
- **Verify:** Esto compila el código sin flashear al dispositivo; si hay algún error en el código, se mostrarán en la consola de depuración que se encuentra en la parte inferior de la pantalla.
- **Save:** Guarda los cambios realizados en el código.

En la parte inferior, hay cuatro botones más para navegar por el IDE:

- **Code:** Muestra una lista de las aplicaciones de firmware y te permite seleccionar cuál editar o flashear.
- **Library:** Explorar las librerías creadas por otros usuarios y también desarrollar propias librerías.
- **Docs:** Te dirige a la documentación de Particle.
- **Devices:** Muestra una lista de los dispositivos, para que se pueda elegir cual flashear, y también obtener más información sobre cada dispositivo.
- **Settings:** Cambie la contraseña, cierre la sesión u obtenga el token de acceso para las llamadas API.

### 3.1.8 Aplicaciones y librerías – Particle

El núcleo de Particle Build es la sección "Particle Apps", que muestra el nombre de la aplicación actual en el editor, así como una lista de las otras aplicaciones y también aplicaciones de ejemplo compatibles con la comunidad.

La aplicación que se tiene abierta en el editor se muestra bajo el encabezado "Aplicación actual".

Desde este panel, se dispone de una gran cantidad de botones y acciones que pueden ayudar a crecer y gestionar la biblioteca de aplicaciones.

#### *a) Agregando archivos a las aplicaciones*

A medida que el código base crece, naturalmente debemos crear bibliotecas para administrar mejor el desarrollo del firmware. Para agregar un archivo a la aplicación, debemos pulsar el botón "+" situado en la esquina superior derecha.

Esto creará dos nuevas pestañas, una con .h y otra con la extensión .cpp.

#### *b) Usando librerías*

Las librerías de firmware son una parte importante para conectar el dispositivo a sensores y actuadores. Facilitan la reutilización de código a través de múltiples proyectos de Particle, o para aprovechar el código escrito por otras personas en la comunidad de Particle. Por ejemplo, las librerías de firmware facilitan la obtención de datos del sensor de temperatura DS18B20 sin escribir el código por su cuenta.

Las librerías de Particle se alojan en GitHub, y se puede acceder fácilmente a través de todas las herramientas de desarrollo de Particle, incluyendo el Web IDE.

Para incluir librerías, se debe abrir el editor de librerías, buscar la librería correspondiente para el sensor o actuador y hacer clic en el botón Incluir en el proyecto. Incluir una librería en el proyecto agregará la dependencia de la librería al archivo `project.properties` que se compilará con el proyecto cuando se verifique o se muestre al dispositivo de destino.

Al darle clic en la librería podremos ver su código fuente y todos sus detalles, agregar librerías también aumenta el tamaño de nuestra aplicación.

Finalmente luego de hacer el código y agregar librerías está todo listo para poder flashear nuestra aplicaciones en el dispositivo, simplemente damos clic el botón flash de la esquina superior izquierda, no debemos olvidar que este flasheo se hace mediante OTA para lo cual debemos tener cuidado con los datos a ser usados, ya que mientras más grande el código mayor será su tamaño, en la fig. n° 24 observamos los datos de un código, para evitar eso podemos hacerlo mediante forma serial o en todo caso tomar en cuenta su tamaño, cuando el código se compila correctamente el parte inferior nos aparecerá un aviso donde al darle clic nos dará detalles de la aplicación.

*Figura 24: Metadatos de un código del Electron.*

```
Output of arm-none-eabi-size:
text      data      bss      dec      hex
12804     8         708      13520    34

In a nutshell:
Flash used   12812 / 110592  11.6 %
RAM used     716 / 20480    3.5 %
```

*Elaboración: Propia.*

### *c) El Firmware*

Un sistema integrado como el Core, Photon o Electron no tiene un sistema operativo como un ordenador tradicional. En su lugar, ejecuta una sola aplicación, a menudo llamada firmware, que se ejecuta siempre que el sistema está alimentado.

Firmware es así llamado porque es más concreto que el software y más suave que el hardware. El hardware es fijo durante la fabricación, y no cambia. El software se puede actualizar en cualquier momento, por lo que es muy flexible. El firmware está en algún punto intermedio; Las empresas de hardware emiten actualizaciones de firmware, pero tienden a ser muy poco frecuentes, porque actualizar el firmware puede ser difícil.

Cuando flasheamos un código en el dispositivo, se está haciendo una actualización de firmware over-the-air. Esta actualización de firmware sobrescribe casi todo el software del dispositivo; la única parte que no ha sido tocada es el gestor de arranque, que gestiona el proceso de carga del nuevo firmware y garantiza que siempre se puede actualizar el firmware a través de USB o mediante un restablecimiento de fábrica.

#### **3.1.9 El servicio de facturación de Particle**

El Electron necesita datos para estar en línea, así que Particle tiene su propia SIM y el servicio mismo. Los planes de Particle funcionan de manera muy similar a los planes pre-pagados para teléfonos celulares. Al principio de un período mensual de facturación se cobra por la "tarifa base", que es el costo de tener la tarjeta SIM en vivo en la red. La tasa base incluye el primer megabyte (MB). Si supera el primer MB, se agregara otro MB al plan de servicio a un costo menor que el de base. En los EE.UU., por ejemplo, la tasa base es de \$ 2.99 y MB adicionales son \$ 0.99. Estos MB adicionales se agregan a la factura del mes siguiente.

Este precio se está casi usando en toda América, por lo que en Perú también se cobrara lo mismo.

Este servicio se adhiere a una operadora móvil, en el presente caso el SIM de Particle se adhiere a la red de Movistar.

Para ver el uso de la SIM, límite de datos, pagos y demás, existe la sección Console en la web de Particle.

### ***3.1.9.1 Periodo de facturación***

El primer día que se configura el primer SIM se convierte en el día mensual.

Si se agrega más SIMs a la cuenta, también se sincronizarán hasta el día del SIM original.

### **3.1.10 Optimizando el uso de datos**

#### ***a) Cambiado a UDP desde TCP***

El mensaje no cifrado para un evento pequeño publicado es de 10 bytes. Agregando 20 bytes de cabecera de IPv4, ya se tiene 30 bytes de datos comunes a las comunicaciones TCP del Photon y a las comunicaciones UDP del Electrón.

En el Photon, el modo de cifrado AES es CBC, que siempre redondea hasta un tamaño de bloque de 16 bytes. Ese mensaje de 10 bytes sólo tiene 6 bytes más grande, además de añadir una longitud de 2 bytes a nuestros mensajes cifrados para que el receptor sepa cuántos bytes para retirar desde la conexión. La trama TCP es de 20 bytes, más todos los mensajes se reconocen en el nivel TCP añadiendo otros 20 (TCP) + 20 (IP) para el ACK. Por lo tanto, incluso si no reconocemos en la capa de aplicación, un pequeño evento no confirmado, publicado de un Photon resultados en al menos 98 bytes de comunicación.

En el Electrón, el modo de encriptación AES es CCM, que no añade ninguna sobrecarga, sin embargo, el túnel DTLS tiene 29 bytes de sobrecarga por datagrama. La trama UDP es de 8 bytes, mucho más pequeña que la de TCP 20. Además, la capa UDP no tiene confirmación incorporada. Un pequeño evento no confirmado, publicado de un Electrón es sólo 67 bytes.

#### ***b) Reducción de la frecuencia de ping***

Para mantener la conexión TCP viva en el Photon tenemos que enviar un ping cada 15 segundos, incluso si su aplicación no se está comunicando en absoluto. Los pings terminan siendo del mismo tamaño que el publicado anteriormente, además de reconocerlos en el nivel de aplicación. El ping inicial da como resultado 98 bytes, más entonces el reconocimiento de la aplicación es otro 98 bytes. Eso es 196 bytes cada 15 segundos, o casi 46 kilobytes por hora, apenas en pings.

En el Electron, usando UDP, no hay conexión para mantener vivo. La única razón por la que enviamos pings es mantener el NAT abierto en el proveedor de celulares. Un electrón que está parpadeando cian y haciendo nada, hace pings cada 23 minutos. El ping y el acuse de recibo de la nube son cada uno de 61 bytes. Que promedia a 318 bytes por hora, una reducción dramática.

#### ***c) Menos handshakes y sesiones más largas***

Cuando un dispositivo se conecta inicialmente a la nube de Particle, tiene que realizar un handshake o apretón de manos para configurar el cifrado. El handshake es siempre grande en comparación con los mensajes de flujo, kilobytes múltiples para el Photon y el Electron. Sin embargo, podemos controlar la frecuencia de los handshakes.

Con TCP en el Photon, un handshake debe ocurrir para cada conexión recién abierta. Un Photon que se duerme durante una hora, se despierta para leer sus sensores

cada hora, y publica los datos acumulados cada 6 horas tendrá que darle un apretón de manos cada 6 horas cuando se conecte a la nube.

En el Electron usamos UDP, que es sin conexión. Un electrón puede irse a dormir, despertarse algún tiempo más tarde y hablar con la nube de Particle sin realizar un handshake, continuando con las credenciales de sesión DTLS establecidas previamente. Actualmente el Electron tiene que hacer un handshake al menos una vez por semana, y esto está sujeto a cambios. El uso de datos handshake se reduce dramáticamente de centenares de KB por semana para los Photones a algunos KB por semana para los electrones.

#### *d) La estructura de la transmisión de datos*

Cuando los datos se envían a o desde el dispositivo, viene con una sobrecarga fija. Esto incluye el encabezado UDP (User Datagram Protocol), el encabezado DTLS (Datagram Transport Layer Security) y un mensaje de confirmación (ACK). El encabezado UDP es necesario porque especifica de dónde vino el paquete y dónde va, y que agrega 26 bytes a un mensaje. Sin el encabezado de DTLS no podríamos cifrar sus datos, y eso es 27 bytes adicionales. Por último, el ACK es un mensaje muy corto de la nube con sus propias cabeceras UDP y DTLS que actúa como un recibo y confirma que un mensaje fue recibido, y añade alrededor de 61 bytes. Para un Particle.publish muy corto, está buscando un total de casi 128 bytes, en su mayoría compuesto de infraestructura.

#### *e) ¿Qué consumen los datos?*

Cada vez que el Electron se comunica con Internet o viceversa, los datos viajan a través de la red celular y se miden. Donde también hay una inevitable sobrecarga en todas las comunicaciones debido a la estructura de Internet y la necesidad de seguridad y



fiabilidad. Muchas de estas comunicaciones son intencionales y obvias, como llamar a Particle.publish () o un código intermitente en el aire. Hay otras comunicaciones de las que tal vez ni siquiera sepan. Por ejemplo, cuando un Electron se enciende o se restablece tiene que registrar con las torres celulares y la nube de Particle, y este sistema de mensajes puede utilizar tanto como 6KB. Los siguientes datos de uso:

**i. Funciones de firmware o registradas por el firmware:**

- Publicar (comunicación más pequeña)
- Llamadas de funciones desde la API
- Llamadas variables desde la API
- Suscribirse a eventos o respuesta Webhook
- Conexiones de red directas a través de TCPServer / TCPClient / UDP

**ii. Conectividad de fondo:**

- Pings (una vez cada 23 minutos si no hay otras comunicaciones y el dispositivo está despierto 122 bytes)
- Handshakes (siempre que el dispositivo esté conectado, y una vez por semana hasta 6 KB)

**iii. Flashear un código:**

- Flashear un código over-the-air

**iv. Tinker App**

- Todas las funciones de Tinker (E / S digital, E / S analógicas), utiliza las API para comunicarse con el Electron.

**v. Bibliotecas: HTTPClient, MQTT, etc.:**

Siempre que se utilice una biblioteca que hable sobre la red, usará datos. Una gran cantidad de datos. No es recomendable utilizar estos con el Electron, ya que utilizarán datos mucho más rápido que las publicaciones y funciones de Particle.

### *f) Formas de reducir el uso de datos*

Hay muchos métodos para poder reducir el uso de datos, se tocarán los métodos de forma ascendente en su complejidad.

- **Usar nombres más cortos**

Esto se aplica a `Particle.publish`, `.variable` y `.function`. Esos nombres más largos tienen que ser enviados a y desde la nube, por lo que es mucho mejor usar `Particle.publish("x")` que `Particle.publish("x_proyecto_de_tesis_final")`.

- **Utilizar `Serial()` para el desarrollo**

Cuando se está probando y depurando el código por primera vez, se puede evitar escenarios de publicación de datos desenfrenados enviando lecturas de sensores, alertas, etc. a través de un cable USB. Comentamos la línea `Particle.publish()`, y se agrega `Serial.begin(9600)` a `Setup()` y en su lugar utilizar `Serial.println("datos_aqui")` para registrar datos en un terminal serial. Se puede usar `Particle Dev`, el IDE de Arduino o cualquier otro programa de terminal.

- **Publicación impulsada por eventos**

Una de las estructuras muy comunes que vemos en los códigos es un `loop()` con una lectura de sensor, luego una publicación y una demora (`delay`). Esto llama a la publicación (y utiliza los datos) en algún intervalo regular, pero los datos que se informaron pueden no haber cambiado, como ejemplo: imaginemos un sensor de temperatura en el patio: es poco probable que la temperatura haya cambiado mucho después de 1 segundo, 1 minuto o incluso 10 minutos. Lo más eficiente es guardar la temperatura que publicó por última vez y comparar la lectura actual con la anterior. Si es más que algunos grados diferentes, entonces se publica el nuevo

dato. Esta forma de ahorro está directamente relacionada con el concepto de redes fog o redes en la niebla, ya que estos datos se encuentran en el borde del terminal, no es necesario pasar todos los datos a la nube, si no que se mantienen en el borde hasta que un dato necesario pueda ser requerido, aplicando así el concepto de redes fog. Ejemplo aplicando a un código:

```
If(abs(temperatura-lastTemperature)>5){
Particle.publish("T", temperature);
}
```

Fuente: [www.particle.io](http://www.particle.io) (2017)

- **Combinar publicaciones**

Es posible que se esté enviando datos que nos con críticos en el tiempo, entonces se puede considerar combinar muchos puntos de datos en una sola publicación, en lugar de utilizar los 100 bytes de sobrecarga para cada punto de datos, sólo se utilizará una vez y se podrá ahorrar mucho, muestreando los datos basados en el cambio o la frecuencia, se puede almacenar en una cadena o una matriz y enviarlos como una única publicación cuando se haya recogido tanto como se desea (o llenado en el campo de los datos de 255 bytes de un Particle.publish). Se puede agregar comas entre puntos de datos para que sean fáciles de separar.

Ejemplo aplicando a un código:

```
Particle.publish("T", String::format("%d,%d,%d,%d,%d",
temperatures[0], temperatures[1], temperatures[2], temperatures[3],
temperatures[4]));
```

Fuente: [www.particle.io](http://www.particle.io) (2017)

- **On-Device Logic**

La mayoría de los comportamientos que nos gustaría implementar pueden vivir directamente en el Electron. Por ejemplo, si se tiene un invernadero con una ventilación motorizada y se desea que la ventilación se encienda si la temperatura excede un umbral establecido por el usuario, entonces hay dos opciones. Se podría enviar la temperatura de lectura a la nube de vez en cuando, comparar la lectura con el umbral, y tener la nube de control de la ventilación a través de una función si la temperatura está por encima del umbral. Esto es menos eficiente respecto a los datos, ya que se podría usar una función para establecer el umbral en el dispositivo mismo cuando sea necesario, y de otro modo mantener la lógica y el control por completo en el dispositivo. Esto tiene la ventaja añadida de ser más confiable en el caso de una interrupción de la conectividad.

En este caso también nos adentramos al concepto de redes en la niebla, ya que los datos se mantienen en el borde y solo los requeriremos cuando sean necesarios.

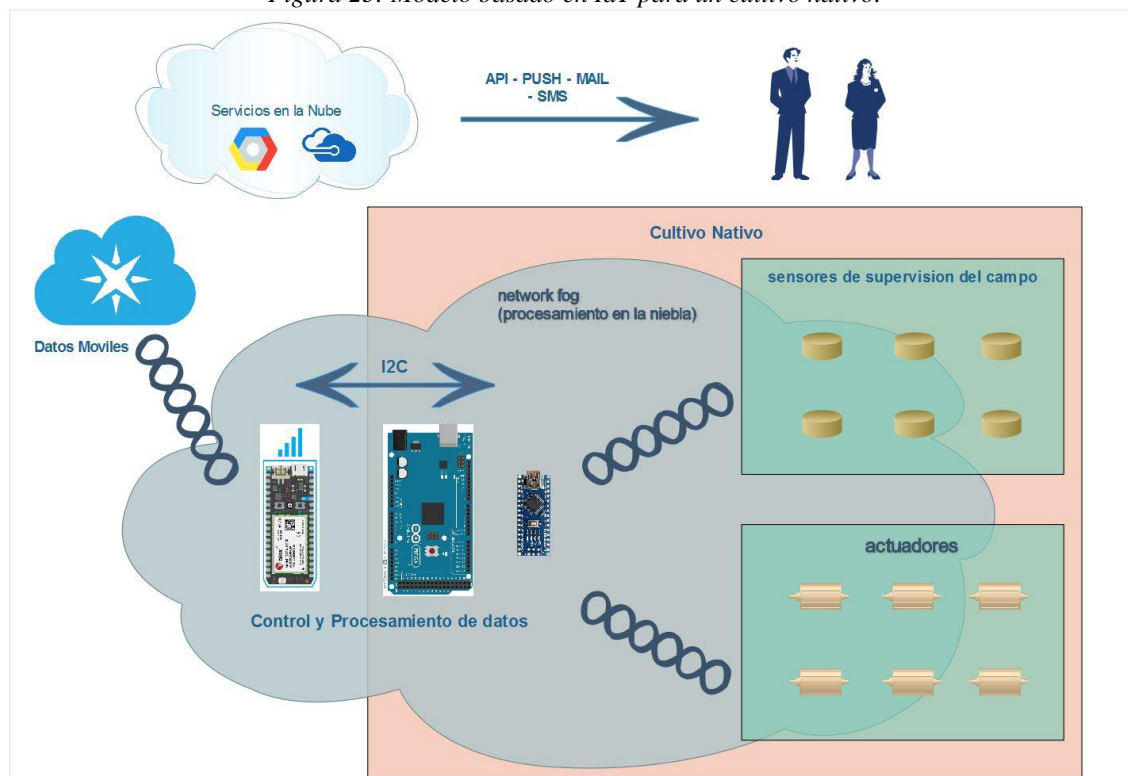
- **Estado "dormido"**

Dentro de la programación nos encontramos con esta función “System.sleep ()”, cuando el Electron está en este estado no usará pings y el consumo de energía será mucho menor, también es posible probar con los modos del mismo.

### 3.2 Creación del modelo como solución IdT

En la siguiente fig. n° 25 (figura ya mencionada) se presenta, la creación del modelo para nuestro caso, como ya sabemos este se adapta según el producto del cultivo.

Figura 25: Modelo basado en IdT para un cultivo nativo.

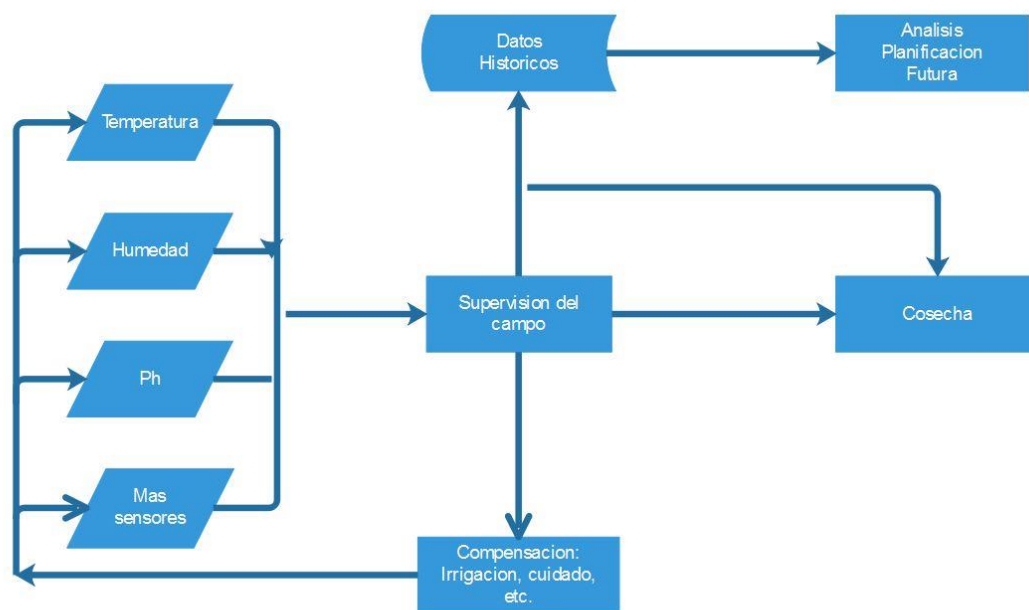


Elaboración: propia.

### 3.3. Diagrama de flujo del modelo

Los sensores en el cultivo proporcionan supervisión para lograr condiciones de cultivo óptimas., según los umbrales determinados, el sistema de irrigación y protección se puede activar para mantener las condiciones del campo dentro de los rangos ideales para el cultivo. En la figura n° 26 podemos ver un ciclo de realimentación, además, los datos para la temporada de cultivo y la cosecha resultante se pueden capturar y almacenar para un análisis posterior. El análisis puede ayudar a identificar las oportunidades y las mejoras en las operaciones generales del cultivo para futuras temporadas.

Figura 26: Diagrama de flujo del modelo de monitoreo.



*Elaboración: propia.*

### 3.4 Adquisición de los datos

Como se dijo en el inicio, para poder obtener datos, necesitamos dispositivos que nos de tal información, para eso están los diversos sensores para poder adquirir los datos necesarios para que el proceso de cultivo sea de la manera más óptima, estos sensores deben ser programados en las placas de desarrollo para poder obtener los resultados analógicos y que sean procesados de manera digital, los sensores utilizados son los siguientes:

- Sensor de PH, este sensor es un electrodo que permitirá obtener el PH del agua con el que se riega, este sensor es necesario calibrarlo, y se estabiliza entre 1 – 2 minutos.
- Sensor de temperatura ambiente, nos permitirá obtener la temperatura de manera digital, y mediante programación hacer el cambio a grado °C o °F.

- Sensor de humedad ambiental, este sensor funciona de la misma manera que el de temperatura ambiente, ambos sensores, nos permitirán prevenir el clima externo.
- Sensor de temperatura de suelo.
- Sensor de humedad de suelo, junto al de temperatura de suelo, nos permiten obtener los datos para la activación del riego y protección.
- Sensor de movimiento PIR, nos permite ver presencia externa como aves dentro de la cosecha o almacenamiento.

### 3.5. Usando las redes en la niebla (Fog Network)

En esta parte pondremos en uso el concepto de Fog Network o como ya se dijo redes en la niebla, ya que aquí se hará un primer procesado de los datos y como el concepto lo dice, estas redes se encuentran en los perímetros y nos ayudan a optimizar el uso de datos, para lo cual dentro este concepto se tomó como opción el uso de la comunicación I2C.

El bus I2C es la abreviatura de Inter-IC (inter integrated circuits), un tipo de bus diseñado por Philips Semiconductors a principios de los 80s, que se utiliza para conectar circuitos integrados (ICs) sus primeras aplicaciones se dieron en video sin embargo con los años se amplió su uso, de tal manera que se está convirtiendo en un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de "inteligencia", sólo requiere de dos líneas de señal y un común o masa, es decir que de esta manera podemos conectar diversos dispositivos de diversos fabricante, en el presente caso conectaremos el Electron de Particle con Arduino.

La metodología de comunicación de datos del bus I2C es en serie y sincrónica.

Una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos.

Las señales que se usan son:

SCL (System Clock) es la línea de los pulsos de reloj que sincronizan el sistema.

SDA (System Data) es la línea por la que se mueven los datos entre los dispositivos.

GND (Masa) común de la interconexión entre todos los dispositivos "enganchados" al bus.

Las líneas SDA y SCL son del tipo drenaje abierto, es decir, un estado similar al de colector abierto, pero asociadas a un transistor de efecto de campo (o FET). Se deben polarizar en estado alto (conectando a la alimentación por medio de resistores "pull-up") lo que define una estructura de bus que permite conectar en paralelo múltiples entradas y salidas, de esta manera podemos tener diversos dispositivos conectados en una misma línea y cada dispositivo funcionara según la línea de donde sea llamada con la siguiente función:

```
Wire.begin() // función de inicio de la conexión I2C
```

```
Wire.beginTransmission() // función donde indicamos la dirección que usaremos
```

Dentro del concepto de Internet del Todo, se debe cumplir con la interacción de los pilares (M2M, M2P, P2P) para que este exista, por lo tanto en este paso se está cumpliendo con la interacción M2M, debido a la conexión entre dos dispositivos y a su vez permitiendo que los datos se procesen entre ellos y no salgan de la niebla, excepto cuando sea necesario.



Los datos que serán procesados en este paso, serán los del cultivo en sí, es decir los datos adquiridos por el sensor humedad y temperatura del suelo serán procesados entre ellos, es decir al borde de la red, la activación también del riego y los motores se darán de manera autónoma solo cuando estos datos pasen los valores mínimos.

### **3.6 Inicio del Internet del Todo**

Para poder salir de la red necesitamos conectarnos a Internet, para lo cual tenemos dos opciones que son de manera alámbrica o inalámbrica, tomando en cuenta que el presente proyecto está enfocado para zonas agrarias donde no existe este acceso a internet, pues sabemos que no existe un cableado para un internet alámbrico, por tanto la opción a elegir será de manera inalámbrica, en este punto se podría pensar que sería mediante una red wlan, sin embargo las redes wlan igualmente necesitan una conexión alámbrica del router, por tanto la opción final es el uso de redes móviles, como bien se sabe cada año las operadoras móviles van aumentando su cobertura, por lo tanto no es descabellado decir que nuestros campos puedan comunicarse mediante telefonía móvil, puesto que esta cobertura ya casi cubre la mayoría de lugares donde también se hace el agro, otro punto a favor es que no necesitamos un conexión de alta velocidad ya que la transmisión de datos se dará en kb/s por lo que con una red 2G (gprs o edge) será suficiente.

Es aquí donde podemos decir que el núcleo que nos permitirá esto será el dispositivo llamado Electron de la compañía Particle, este dispositivo se conecta mediante redes móviles, para el caso de nuestro país se conecta mediante la red de Movistar, este dispositivo cuenta con un chip propio, este chip no es tarifado ni contralado por la compañía móvil sino más bien por la misma empresa Particle, de esta forma no necesitamos contratos ya que el pago se hace directo con la empresa.

Otro beneficio de tomar esta alternativa y no la de hacer un diseño para llevar internet inalámbrico Wi-Fi, es que es económico, el dispositivo en si es más económico que implementar toda una red de acceso y por otro lado el uso de los datos son de igual forma, ya que no es específicamente un plan si no que se tarifa el consumo, el precio mínimo de este es de 2.99\$ por el primer Mb de uso y 0.99\$ por cada Mb adicional, teniendo un tope de 5 Mb por mes costo más que suficiente, como ya se comentó el uso de los datos se da en Kb por lo que este será más que suficiente dependiendo de nuestro diseño, más adelante se detallare la cantidad de datos que se usa, pero no solo por los datos, sino que también se optimizara el diseño para que el uso de los datos sea el mínimo posible.

Viendo desde el punto de vista económico y factible, no solo es una opción para lugares que no tienen acceso a Internet sino que también, es una opción para poder reemplazar en lugares donde ya tienen acceso a Internet ya que el costo de instalación y mantenimiento es mucho menor, logrando así una mejora en la producción y la economía del agricultor.

El uso de redes en la niebla reduce drásticamente la cantidad de data, por lo que el Electron solo procesara la información más relevante, y que nos permita hacer la prevención y predicción correcta, por lo que en esta etapa el Electron procesara los datos ambientales, ya que estos parámetros son quienes ocasionan las mayores pérdidas en los cultivos, tomando en cuenta que los aspectos climáticos no siempre son acertados por páginas web, aplicaciones móviles, etc., por lo que es muy necesario tener los parámetros del sitio para no correr riesgos y mejorar el cultivo del producto.

Con nuestro núcleo de procesamiento recolectaremos la segunda parte de los datos como son los sensores de: PH, humedad y temperatura ambiente, y de movimiento PIR, para

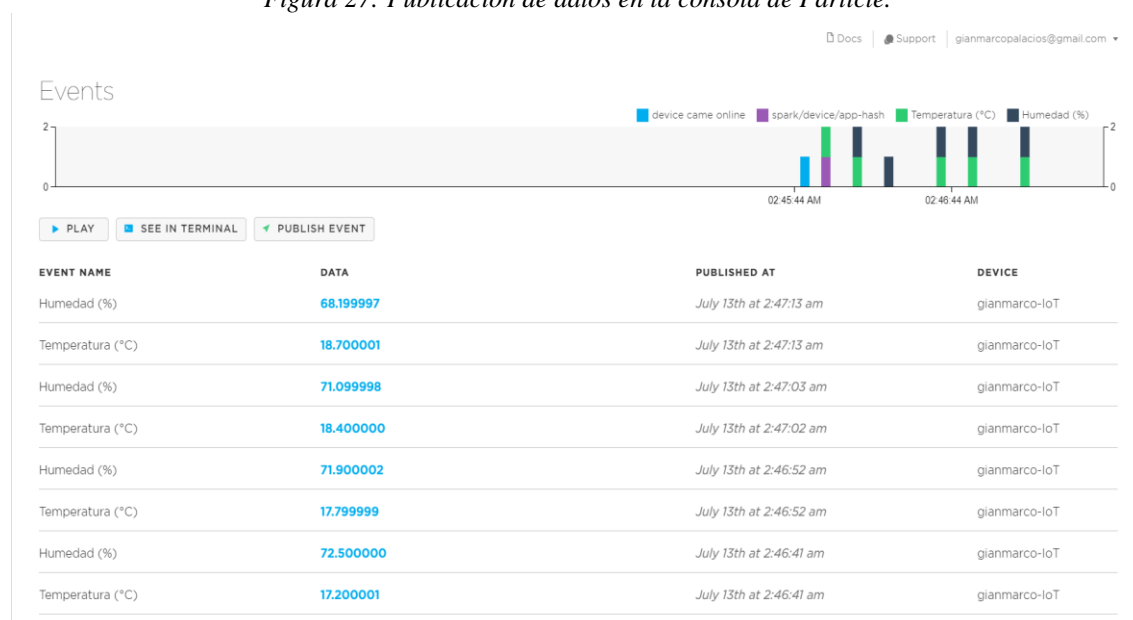
después mediante su programación permitir que se nos abra la puerta hacia Internet, para lo cual por una parte se programara el control de los sensores, también se programara la salida de estos datos, esto es posible mediante el uso de los comando:

```
Particle.publish(const char *eventName);
```

```
Particle.publish(String eventName);
```

Como podemos ver en el comando podemos usar un nombre que se haya designado o el nombre del evento en sí. En la fig. n° 27 podemos observar el funcionamiento del comando y como esta se conectó a internet, ya que para acceder y visualizar los datos se debe dirigir a un navegador web e ingresar al apartado **console** de Particle, vemos como la temperatura y humedad fluctúan, para este paso se usó fuego para modificar los parámetros.

Figura 27: Publicación de datos en la consola de Particle.



Elaboración: Propia.

Esta sería la forma más práctica de poder usar nuestros datos sin embargo necesitamos visualizarlos de manera gráfica y también disminuir el tráfico de ellos para lo cual se optó por utilizar ThingSpeak.

### 3.6 ThingSpeak

Antes de pasar al uso de ThingSpeak, debemos de hablar de las API (Application Programming Interfaces), por su traducción al castellano como Interfaces de Programación de Aplicaciones, una API es una especificación formal sobre cómo un módulo de un software se comunica o interactúa con otro, es decir es un conjunto de comandos, funciones y protocolos informáticos, que permiten comunicarse con un software, esto es muy útil ya que no se necesita trabajar el código desde cero, si no con las instrucciones predefinidas del software, desde mi criterio propio una API es una puerta.

Tomando en cuenta la definición de una API, se optó por utilizar la plataforma ThingSpeak, que se trata de una aplicación Open Source para la interacción con el IdC e IdT, esta plataforma utiliza una API para almacenar y recuperar datos de los objetos usando HTTP, de esta forma uniendo los conceptos se utilizara el API de ThingSpeak del cual recopilara la información enviada por el Electron, de esta forma mediante su interfaz gráfica podremos ver nuestras variables en el tiempo.

Dentro de la aplicación web se creara una cuenta y posteriormente un canal, con nombre Monitoreo Campo y dentro los parámetros de Temperatura y Humedad, como se muestra en la figura n° 28.

Figura 28: Creación de un canal en ThingSpeak.

## New Channel

**Name**

**Description**

**Field 1**

**Field 2**

**Field 3**

*Elaboración: Propia.*

Posteriormente nos dirigimos a la pestaña de API donde podremos observar las API keys de lectura y escritura como se muestra en la figura n° 29.

*Figura 29: API Keys.*

### Monitoreo Campo

Channel ID: 302114  
Author: gianma123  
Access: Private

[Private View](#) [Public View](#) [Channel Settings](#) [Sharing](#) **[API Keys](#)** [Data](#)

#### Write API Key

Key

[Generate New Write API Key](#)

#### Read API Keys

Key

*Elaboración: Propia.*

Luego nos dirigimos a la IDE de Particle para la programación, dentro se debe insertar las librerías correspondientes de los sensores a usar, este caso un DHT22 y también la librería de ThingSpeak, en este paso procedemos a colocar el ID del canal y también la API de escritura, como se muestra en la figura n° 30.

Figura 30: Programación del API key y channel ID.

```

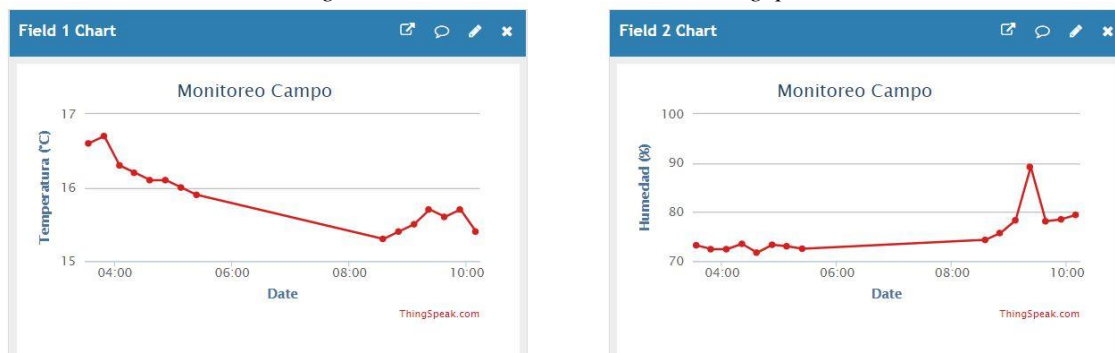
20 /* Thingspeak */
21 TCPCClient client;
22 unsigned Long myChannelNumber = 302114;
23 const char * myWriteAPIKey = "NKSWD8KC8YOE6MHW";
24

```

Elaboración: Propia.

Procedemos a compilar el código y veremos cómo nuestros datos empiezan a aparecer en ThingSpeak como se muestra a continuación en la figura n° 31.

Figura 31: Visualización de datos en ThingSpeak.



Elaboración: Propia.

En el gráfico anterior podemos ver la fluctuación de manera visual de nuestros datos, la comunicación se dio mediante la API de ThingSpeak, sin embargo para mejorar la comunicación y uso de datos usaremos Webhooks.

### 3.7 Webhooks

Los Webhooks son una herramienta que nos permite hacer notificaciones mediante HTTP, por lo tanto nos permite el intercambio de información entre aplicaciones sin modificar sus infraestructuras, lo cual se les denomina “Callbacks HTTP de usuario”, en el presente proyecto será un puente entre Particle y ThingSpeak, tomando la decisión de usar Webhooks ya que de esta manera los datos pasaran de un lado a otro de manera más segura, puesto que en IdT, uno de los mayores problemas es la seguridad.

El siguiente Webhook es creado mediante la integración de Particle, de manera anterior sea crea un canal en ThingSpeak, donde igualmente debemos tener el ID del canal y la API de escritura, estos datos colocaremos en las integraciones de Particle, en este paso cambiaremos los nombres de los eventos a sus abreviaciones para el ahorro de data. A continuación el siguiente código para crear un Webhook:

```
{
  "event": "nombre_del_evento",
  "url": "https://api.thingspeak.com/update",
  "requestType": "POST",
  "form": {
    "api_key": "{{k}}",
    "field1": "{{1}}",
    "field2": "{{2}}",
    "field3": "{{3}}",
    "field4": "{{4}}",
    "field5": "{{5}}",
    "field6": "{{6}}",
    "field7": "{{7}}",
    "field8": "{{8}}",
    "lat": "{{a}}",
    "long": "{{o}}",
    "elevation": "{{e}}",
    "status": "{{s}}"
  },
  "mydevices": true,
  "noDefaults": true
}
```

Elaboración: propia

### 3.8 Matlab Visualizations

Dentro de ThingSpeak tenemos varios recursos, uno de ellos es el de poder usar visualizaciones basadas en Matlab de los datos obtenidos, de esta manera podemos interpretar y hacer un acercamiento de los resultados futuros, para debemos introducir el siguiente código, que nos dará como resultado el siguiente grafico en la fig. 32.

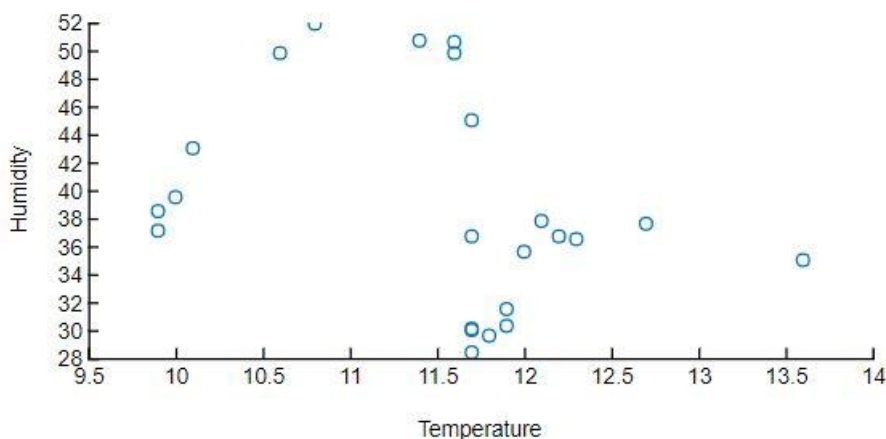
```

readChannelID = debemos colocar el id del canal;
fieldID1 = 1;
readAPIKey = 'la API de lectura';
%% Lectura de Data %%
[data, time] = thingSpeakRead(readChannelID, 'Field', fieldID1, 'NumPoints', 10,
'ReadKey', readAPIKey);
%% Visualizacion de Data %%
thingSpeakPlot(time, data);

```

Elaboración: propia

Figura 32: Visualización de la humedad frente la temperatura mediante programación Matlab.



Elaboración: Propia.

### 3.9 Pushbullet

En este punto del Proyecto, casi en el final de la cadena, es necesario que exista la interacción entre el sistema de monitoreo y el usuario, para lo cual tomando en cuenta las interacciones actuales entre persona y máquina, pues están se dan en su mayoría a través de su teléfono móvil, y si estamos hablando de data, pues esta llega de manera de alertas o mensajes push, por lo tanto entre varias opciones para llegar al usuario se optó por usar la aplicación Pushbullet, que es una aplicación que mediante Webhooks, podrá llegar la información de nuestros datos de manera rápida y sencilla.



Es necesario acceder a la versión web, ya que para hacer login se hace mediante Facebook o google, por tanto al activar en los dos mediante la misma cuenta se sincronizaran, se crea un canal de comunicación y en el Webhook colocamos lo siguiente:

```
{
  "eventName": "pushbulletHOME",
  "url": "https://api.pushbullet.com/v2/pushes",
  "requestType": "POST",
  "headers": {
    "Authorization": "Bearer o.JmFYUJxQzxQHldz4nEaFyAGJqaqNwPAG",
    "Content-Type": "application/json"
  },
  "json": {
    "type": "note",
    "title": "{{SPARK_EVENT_VALUE}}",
    "body": ""
  },
  "mydevices": true
}
```

Elaboración: propia

Posteriormente en la línea de comandos colocamos el siguiente comando para activar el Webhook en el Electron:

```
particle webhook create webhook.json
```

Esto nos dará una respuesta y significa que fue configurado correctamente, luego de eso cada vez que haya una publicación en Particle, este también será comunicado por el mismo canal enviando de esta forma el mensaje push con la información a nuestros celulares tal como se muestra en la fig. n° 33.

*Figura 33: Notificación push en un celular mediante Pushbullet.*



*Elaboración: Propia.*

### 3.10 Integración con servicios adicionales

El concepto de IdT se ha venido desarrollando con más fuerza en estos últimos años, de tal manera que grades empresas están brindando servicios para soluciones en IdT, tales como Google con su plataforma Google Cloud, Microsoft con su plataforma Azure, y muchas más, en el presente proyecta se aplicara a integración con Google Cloud como parte final de la cadena de desarrollo del modelo de IdT, ya que esta plataforma permite a los desarrolladores crear, probar e implementar aplicaciones en la infraestructura altamente escalable y confiable de Google.

La transmisión de datos desde dispositivos de Particle a Google Cloud Platform crea innumerables oportunidades para agregar inteligencia y robustez a los productos en IoT.

Por ejemplo podríamos hacer lo siguiente:

- Almacenar los datos de una flota de dispositivos en una base de datos duradera, de larga duración alojada
- Combinar las fuentes de datos de toda una flota en flujos de datos unificados para proporcionar inteligencia de negocios de todo el producto
- Tomar medidas en el mundo físico basado en los resultados del análisis predictivo en un conjunto de datos históricos.

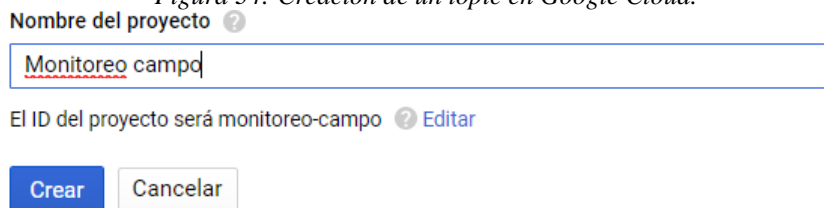
Dentro de algunas opciones del producto se tiene:

- Cloud Datastore: una base de datos NoSQL altamente escalable para aplicaciones web y móviles
- BigQuery: Un almacén de datos rápido, económico y totalmente administrado para análisis de datos a gran escala
- SQL: Un servicio de base de datos MySQL completamente administrado
- Cloud Dataflow: Un servicio en la nube completamente administrado y un modelo de programación para procesamiento de datos por lotes y en tiempo real
- API de predicción: aprender a analizar los datos y hacer predicciones

Para empezar a desarrollar se debe entender que a la hora de hacer el uso del comando *publish*, este al publicar un evento, también podemos subscribirnos a dicho evento, de esta manera es como la integración funciona.

Para empezar la integración debemos crear una cuenta en Google Cloud y posteriormente crear el proyecto (fig. n° 34), dentro de la página podemos ver que existen diversas alternativas para poder desarrollar nuestras aplicaciones en IdT.

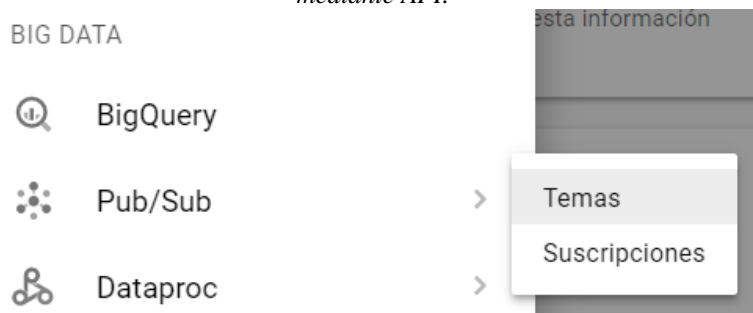
Figura 34: Creación de un topic en Google Cloud.



Elaboración: Propia.

Luego se crea un tema de “pub/sub” (fig. n° 35), para poder acceder al evento así como también habilitamos la API de “pub/sub”

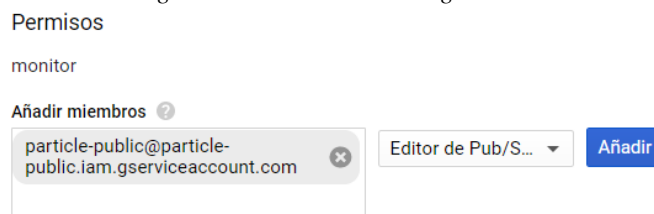
Figura 35: Creación de un SUB/PUB en Google Cloud mediante API.



Elaboración: Propia.

Procedemos a dar permisos para poder publicar en el tema creado de nombre “monitoreo”, colocando el siguiente comando como se muestra en la figura n° 36:

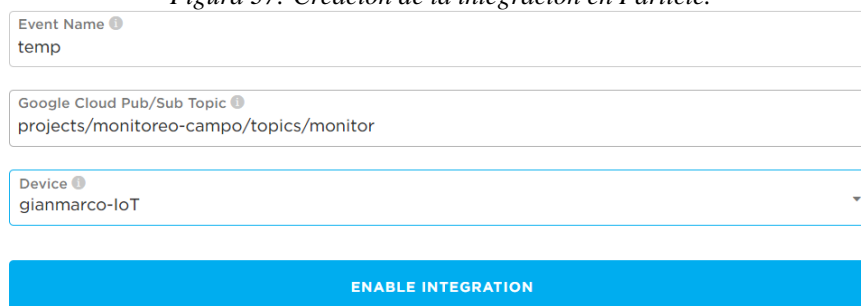
Figura 36: Permisos en Google Cloud.



Elaboracion: Propia.

Para poder habilitar la integración, nos dirigimos a la consola de Particle, donde crearemos una integración con los siguientes datos, como se muestra en la figura n° 37, recordar que en event name, debe ir el nombre del evento del comando Particle.publish()

Figura 37: Creación de la integración en Particle.



Elaboración: Propia.

Le damos a habilitar integración y posteriormente en test, para verificar que haya comunicación extremo a extremo, en el Electron colocamos el mismo firmware con el que se estuvo trabajando en todo lo demás, para tener la integración completa de todo el modelo de IdT.

Para poder verificar que están llegando los datos, accedemos al CLI de Google Cloud.

En donde crearemos la suscripción, para poder ver nuestros datos con la siguiente línea de comandos (fig. n° 38):

Figura 38: Creación de la suscripción mediante el CLI de Google Cloud.

```
gianmarco_iot@monitoreo-campo:~$ gcloud beta pubsub subscriptions create test_sub --topic monitor
Created subscription [projects/monitoreo-campo/subscriptions/test_sub].
gianmarco_iot@monitoreo-campo:~$
```

Elaboración: Propia.

Se procede a configurar la lectura de los mensajes

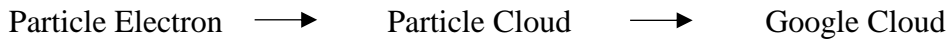
Figura 39: Obtención de los datos visualizados mediante el CLI de Google Cloud.

```
gianmarco_iot@monitoreo-campo:~$ gcloud beta pubsub subscriptions pull test_sub --auto-ack --max-messages 100
```

DATA	MESSAGE_ID	ATTRIBUTES
test-event	73057301200654	device_id=api event=temp published_at=2017-07-18T16:55:54.051Z
test-event	73057516503379	device_id=api event=temp published_at=2017-07-18T16:59:07.463Z

Elaboración: Propia.

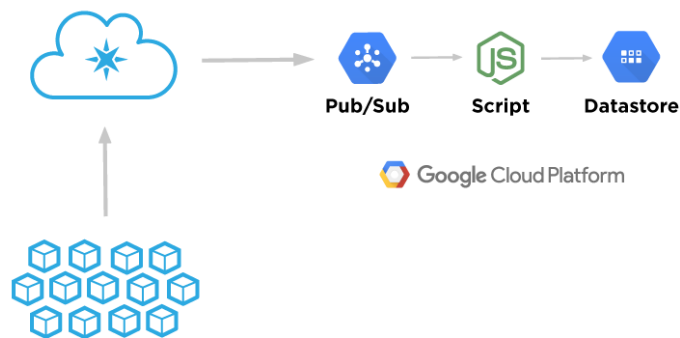
Aquí podemos ver que los mensajes están llegando correctamente desde el Electron (fig. n° 39), cumpliendo la cadena.



Finalmente se procederá a guardar nuestra data, dentro de las alternativas de Google Cloud podemos almacenar nuestra data a largo plazo, tener nuestros datos nos permitirá, poder acceder a ellos y hacer su análisis en el momento que deseemos, o cuando sea necesario saber un evento determinado de una fecha determinada.

Para este último proceso se creara un script que será intermediario en la suscripción y pasar los datos a la datastore, en la figura n° 40 se observa la cadena del transporte de los datos.

Figura 40: Cadena de transporte de datos hacia el Datastore de Google Cloud.



Elaboración: Propia.

Igualmente es necesario pedir permisos en el tema de sub/pub, creando una clave privada que servirá para autenticar el script (fig. n° 41), por lo que nos dirigiremos a los credenciales de Google Cloud.

*Figura 41: Configuración y permisos en el topic de Google Cloud.*

Cuenta de servicio

Nueva cuenta de servicio

Nombre de cuenta de servicio ?

Función ?

ID de cuenta de servicio

@gianmaiot-152605.iam.gserviceaccount.com

Tipo de clave

Descarga un archivo que contiene la clave privada. Guárdalo en un lugar seguro porque no podrás recuperar la clave si se pierde.

JSON  
Recomendado

P12  
Para compatibilidad inversa con código en formato P12

*Elaboración: Propia.*

De esta forma se descargara el script para la autenticación. Clonamos el repo de Google Cloud desde GitHub (fig. n° 42), y se configura y modificara los archivos para poder realizar la comunicación, finalmente corremos el script y dentro de la opciones de Google Cloud nos dirigimos a datastore, habilitamos nuestro servidor donde se guardara nuestros archivos y empezar a recibir nuestros datos (fig. n° 43).

Figura 42: Clonación del repo de Google Cloud desde HitGub

```

Welcome to Cloud Shell! Type "help" to get started.
gianmarcopalacios@gianmaiot-152605:~$ git clone https://github.com/spark/google-cloud-datastore-tutorial.git
Cloning into 'google-cloud-datastore-tutorial'...
remote: Counting objects: 61, done.
remote: Total 61 (delta 0), reused 0 (delta 0), pack-reused 61
Unpacking objects: 100% (61/61), done.

```

*Elaboración: Propia.*

Figura 43: Datastore creado en Google Cloud.

Consultar por tipo    Consultar por GQL

---

Tipo  
        Número de columnas que se pueden mostrar

---

Nombre/ID

id=5629499534213120

*Elaboración: Propia.*

De esta manera cerramos el ciclo del modelo, posteriormente esta información servirá para retro alimentar nuestro sistema y poder prevenir y tomar decisiones en base a la información obtenida, en este punto obtenemos la interacción de M2P y también P2P, ya que con estos datos el agricultor podrá defender frente a los compradores y consumidores que su producto se encuentra en excelente estado ya que fue producido con los valores óptimos del producto, así entonces se cierra el ciclo del modelo para volver a empezar dependiendo de los parámetros que pueden variar cada cierta temporada.

### 3.11 Resumen del proceso de diseño e implementación

- Elección del producto nativo
- Elección del tipo de comunicación (Wi-Fi, red móvil, etc.)
- Elección de dispositivos a usar (procesador, sensores, etc.)
- Diseño de la solución de IdT
- Implementación de códigos necesarios
- Implementación de Fog Network



- Implementación de la conexión a internet
- Implementación de la seguridad (Webhooks)
- Implementación de la visualización de los datos
- Almacenamiento de datos
- Análisis de datos (analytics)
- Mensajería instantánea (push)

## CAPITULO IV

### RESULTADOS Y DISCUSIÓN

#### 4.1. El Electron como medio de transmisión

Elegir el medio de transmisión fue una de las tareas más complicadas porque actualmente en el mercado existen muchas opciones, de manera que era necesario por una que se adecue a nuestras necesidades que principalmente eran: economía, amplia conectividad, facilidad de uso y transporte y escalabilidad.

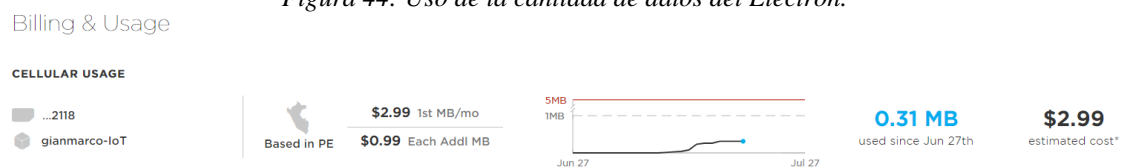
Por lo tanto tomando en cuenta nuestros factores el principal era conectividad, como bien lo dice su nombre mismo Internet del Todo, nuestro modelo necesita estar conectado a Internet, es decir es necesario que salgamos de la red, por lo que en primera instancia se depuro opciones que solo permitan conexiones LAN o WLAN, por lo que descartando estas alternativas, el tipo de conexión que nos quedaba era o por Internet o por Redes móviles, se pensó en implementar con ZigBee, ya que ZigBee tiene gran versatilidad y conexión de kilómetros sin embargo para que salga a Internet necesita de un router, por lo que es aquí donde se descarta opciones que necesiten de Router ya que

esto nos llevaría a la necesidad de tener que implementar una infraestructura adicional por lo que no es lo más conveniente, de esta forma se llegó a la conclusión de que la mejor opción son las redes móviles, ya que las redes móviles actualmente tienen mayor cobertura y llegan a lugares cada vez más lejanos.

Entonces investigando e indagando, es que nos encontramos en que la Empresa Particle desarrollaba un prototipo que se conectaría mediante redes móviles, de manera que sería perfecto frente a nuestras necesidades, para ese entonces no tenía competencia, pero actualmente más empresas (Arduino, Microchip), están desarrollando también sus placas de desarrollo.

El Electron cumple con su objetivo que es conectarse a Internet, pudiendo comprobar que siempre y cuando haya una conexión a la red móvil, este se conectara sin ningún problema, y debido a su empaquetamiento de datos nos permite poder hacer uso de la información en la escala de los Kb como se muestra en la figura n° 44.

Figura 44: Uso de la cantidad de datos del Electron.



Elaboración: Propia.

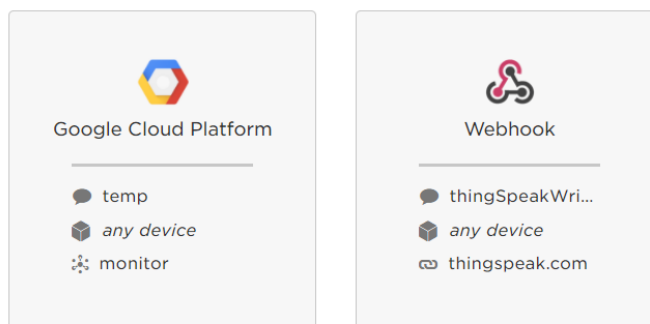
En la imagen podemos ver que nuestro dispositivo está registrado en Perú, y que se tiene un consumo de 0.31MB aproximadamente en una semana y media, por lo que podemos llegar al mes a la cantidad de 1MB siendo el costo de \$2.99, sin embargo podemos consumir más datos dependiendo de nuestras necesidades.

## 4.2 Ventajas del Electron frente a otros dispositivos

La principal ventaja es definitivamente su conexión directa a las redes móviles, aunque ya nuevas empresas estén desarrollando, este ya tiene un tiempo en el mercado por lo que podemos encontrar un mejor soporte.

Sin embargo una de las mayores ventajas, es su integración la Particle Cloud, ya que esta nube nos dará mucha versatilidad, como el poder observar nuestras datos en cualquier parte donde haya una conexión, el estado del uso nuestros datos, la interacción con otros servicios, la creación de integraciones (Webhooks, Google Cloud, etc.), y también el desarrollo de nuestro producto y su posible escalabilidad, en la siguiente figura n°45 se ve las integraciones usadas.

*Figura 45: Integraciones realizadas en Particle.*



*Elaboración: Propia.*

Otra de las ventajas es su IDE tanto de manera web como programa descargable, ya que este tiene interacciones con Github y Atom, apoyando así más su flexibilidad a la hora de realizar proyectos, puesto que los hace de manera más sencilla y rápida, por otro lado al estar basado en C, como muchas plataformas de código abierto, se puede desarrollar aplicaciones basadas en terceros y de manera sencilla también la integración con diversas librerías.

Finalmente, otra ventaja es Tinker, que es su aplicación móvil para Android o IOS, este nos permite configurar nuestro dispositivo de manera gráfica, desde cualquier lugar y acomodarle a nuestras necesidades, y mejor aún el código de la aplicación está disponible para poder usarlo y poder crear nuestras propias aplicaciones basadas en el código y adaptarlas al proyecto de la manera que sea necesario.

#### **4.3 Inconvenientes en el uso del Electron**

Es necesario tener mucho cuidado a la hora de programar nuestras aplicaciones y prestar atención en el uso de datos (en el capítulo 3 se explicó los métodos de uso de reducción de datos), ya que de una manera inapropiada o sin conocimiento preciso, el tráfico de datos podría ser consumido muy rápidamente, llegando a los límites antes de fin de mes.

De la misma manera tener cuidado a la hora de usar los sensores puestos que estos usaran energía directamente de la batería, por lo que se debe implementar de tal manera que no exceda el uso de tal.

#### **4.4. Un sistema de monitoreo económico**

Luego de desarrollar el modelo de solución de IdT, y tomando en cuenta los componentes, si es posible realizar un sistema de monitoreo económico frente a otras opciones, como son estaciones con computadoras y conexión a internet que elevan el precio de tales, así de esta forma se han optado sistemas de monitoreo pero que no están basados en IdT y algunas como en España que se basaron en IdT llegan a ser costosas para cultivos medianos ya que está más orientado a grandes productores, de esta forma podemos decir que respetando el modelo de IdT si podemos realizar un sistema de

monitoreo económico para medianos agricultores y mejorar así la calidad del producto, en la siguiente tabla n° 6 podemos observar una aproximación del costo a invertir:

*Tabla n°6: Presupuesto mínimo para la implementación del sistema de monitoreo.*

Descripción	Unidad de medida	Costo Unitario (S/.)	Cantidad	Costo total (S/.)
Electron		500.00	X1	500.00
Arduino Mega		80.00	X1	80.00
Arduino Nano		22.00	X1	22.00
Plan data	Mes/Mb	9.67	X1	9.67
Sensores	Kit	189.00	X1	189.00
Accesorios		100.00	X1	100.00
Panel solar		150.00	X1	150.00
Analytics	Opcional	Segundo el uso		
Total				1050.67

*Elaboración: Propia*

#### 4.5. La agricultura de precisión

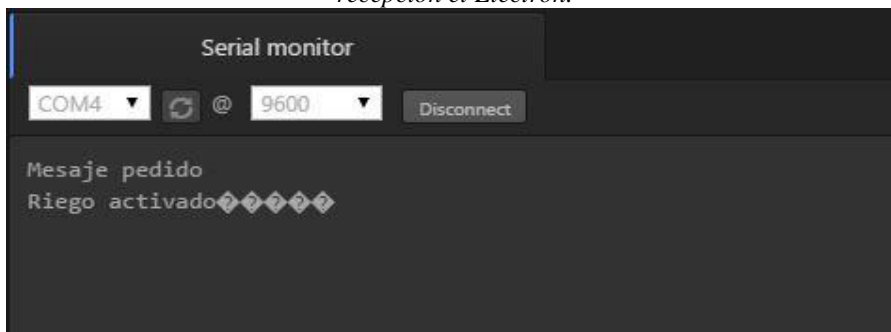
Definitivamente prevenir y predecir sucesos será mejor que atacar el problema cuando este ya se dio, es por eso que es necesario el uso de la agricultura de precisión, que no es nada más que tener los parámetros de cultivos controlados para poder evitar problemas dentro del cultivo, para ello se opta por organizar sensores de tal manera que puedan brindar información de los principales parámetros y causas de las pérdidas y enfermedades, es así que podemos obtener los siguientes resultados sobre las mediciones de nuestros parámetros (fig., n° 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56 y 57), datos que serán procesados posteriormente y tenerlos en la mano.

Figura 46: Comunicación I2C Electron-Arduino/envió de datos del Arduino.



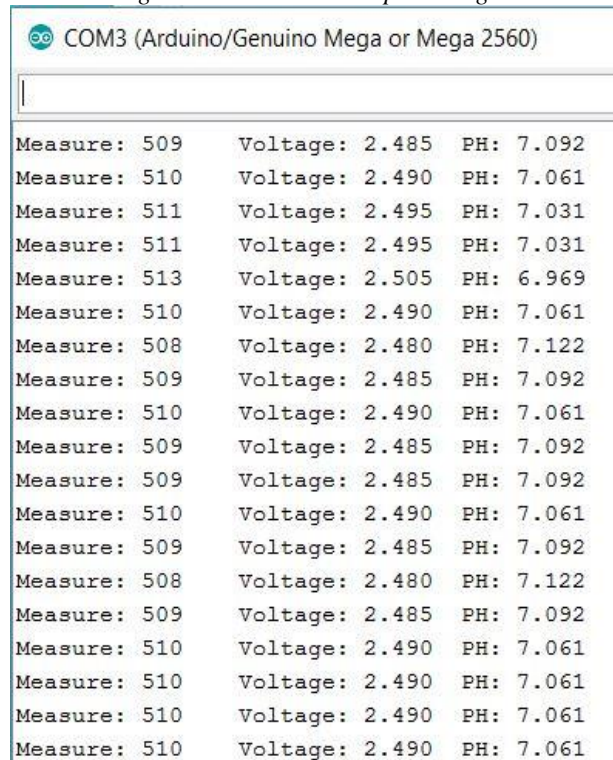
Elaboración: Propia.

Figura 47: Comunicación I2C Electron-Arduino recepción el Electron.



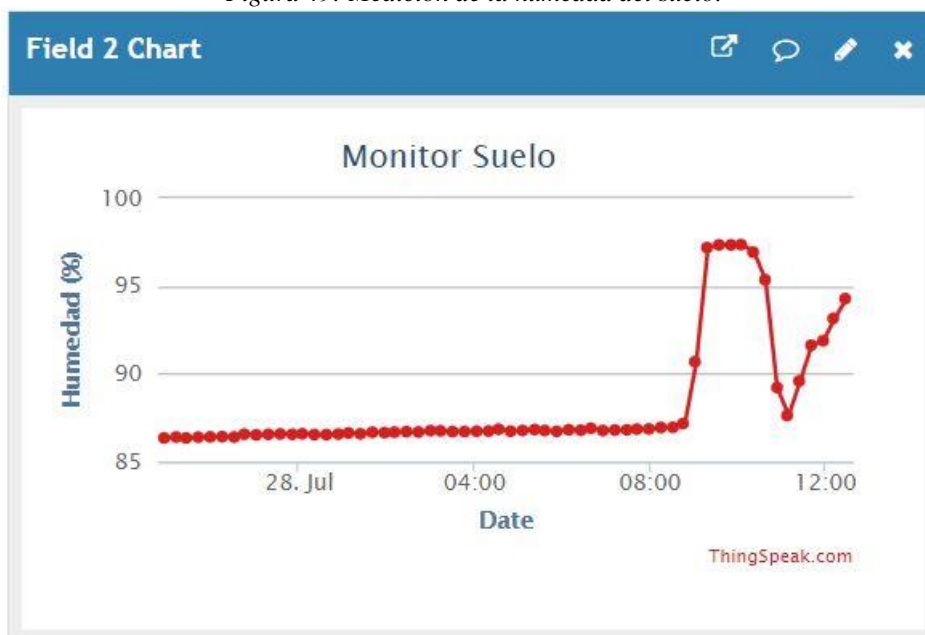
Elaboración: Propia.

Figura 48: Medición del pH del agua.



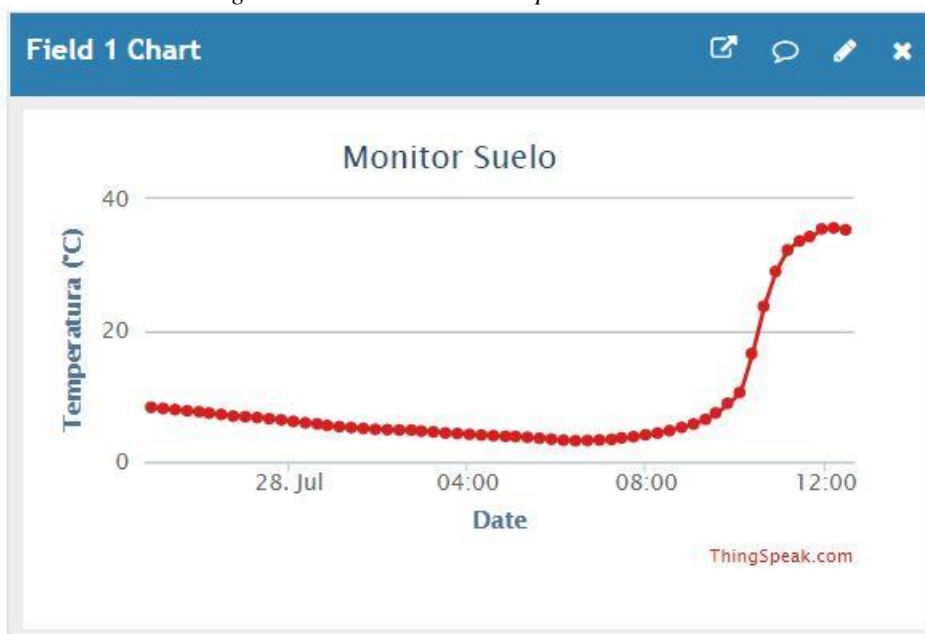
Elaboración: Propia.

Figura 49: Medición de la humedad del suelo.



Elaboración: Propia.

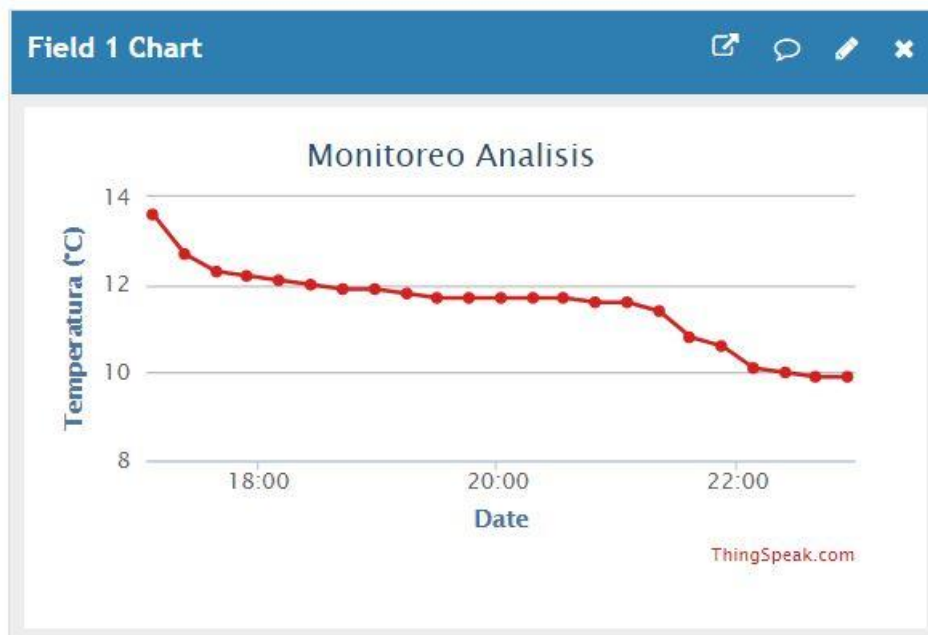
Figura 50: Medición de la temperatura del suelo.



Elaboración: Propia.

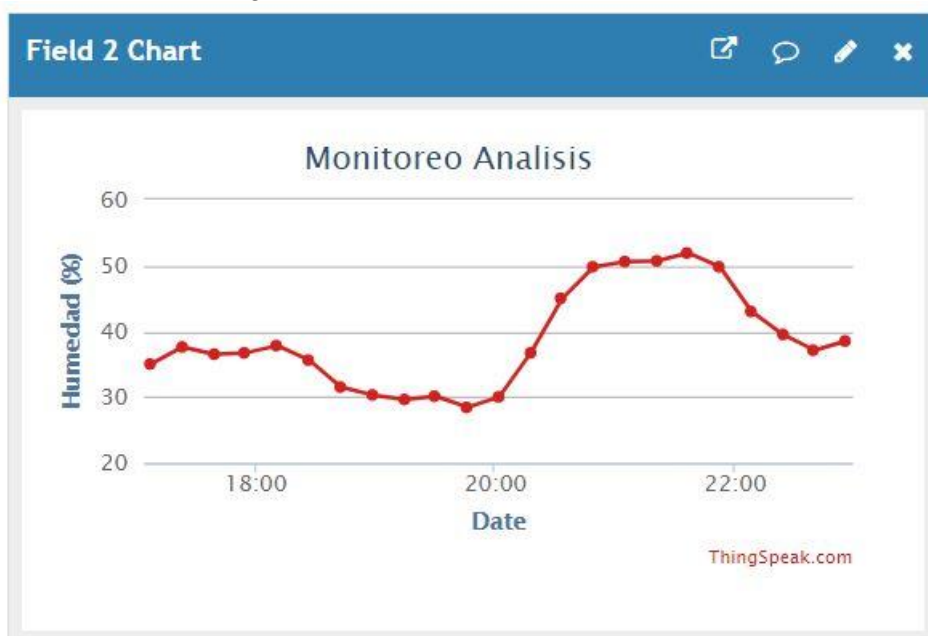


Figura 51: Medición de la temperatura ambiente.



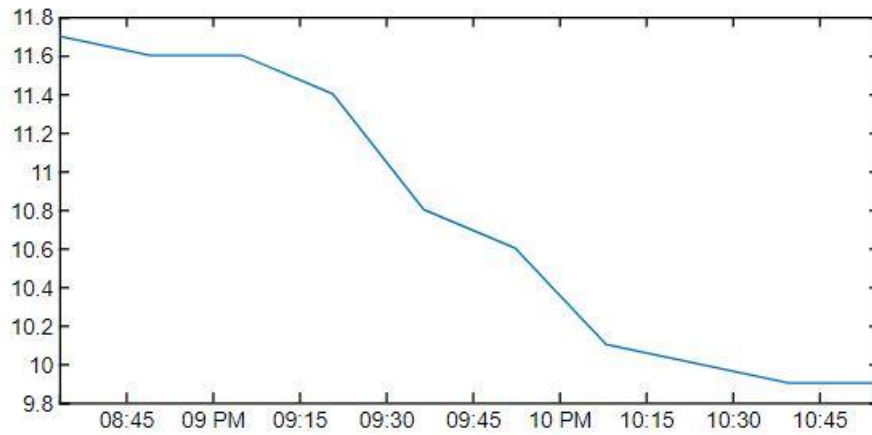
Elaboración: Propia.

Figura 52: Medición de la humedad ambiente.



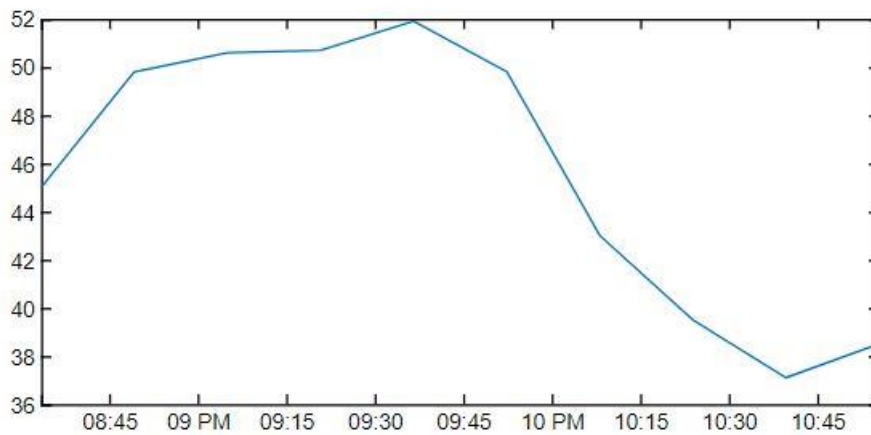
Elaboración: Propia.

Figura 53: Plot de la temperatura ambiente.



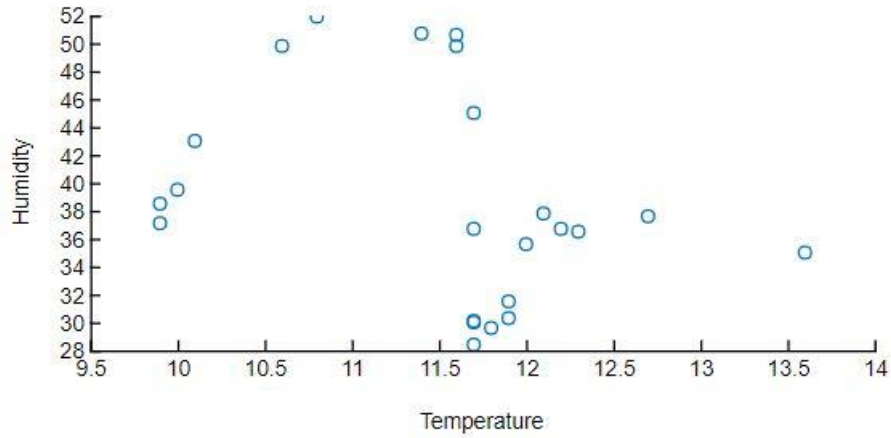
Elaboración: Propia.

Figura 54: Plot de la humedad ambiente.



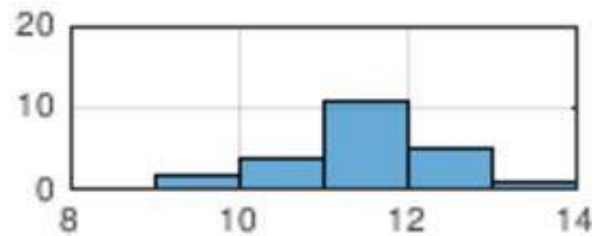
Elaboración: Propia.

Figura 55: Comparación entre humedad y temperatura ambiente.



Elaboración: Propia.

Figura 56: Histograma de la temperatura ambiente.



Elaboración: Propia.

Figura 57: Aviso de detección de movimiento.



Elaboración: Propia.

Con los resultados anteriormente obtenidos, podemos ver que si es posible la obtención de los datos de acuerdo a los parámetros y manejar su control mediante el monitoreo, sin embargo también se debe tener en cuenta el uso de los instrumentos para poder tener mejores resultados, en el caso de los sensores.

#### **4.6. Fog Networks**

Se puede observar que las Fog Network (redes en la niebla), optimizan y ayudan mejor en el procesamiento de los datos, ya que estos son datos que debes estar siendo constantemente monitorizados pero que no necesariamente deben ser visualizados, por lo tanto reducen considerablemente el uso del data, así mismo se tomó como parte de la solución el uso de la comunicación I2C que por el protocolo que maneja, podemos manejar muchos dispositivos conectados mediante la misma línea, siendo su comunicación de manera interna o caso de que se desee visualizar, se haría de manera serial como se mostró en los gráficos.

#### **4.7. Internet del Todo como solución y sus interacciones**

Gracias a que este nuevo concepto nos abre muchas posibilidades en el ámbito de las telecomunicaciones, realmente dentro del modelo planteado cumple como una solución, frente a otras opciones predecesoras donde no se utiliza el concepto, de esta manera es posible la implementación del modelo de monitoreo utilizando este nuevo concepto, sin embargo para cualquier proyecto en general basado en soluciones de IdT, es necesario que este cumpla con la intervención de los 4 pilares (objetos, datos, personas y procesos) y de sus interacciones entre sí (M2M, M2P, P2P), resumidos en la siguiente tabla n° 7:

*Tabla n° 7: Pilares e interacciones del modelo de monitoreo para cultivos.*

Los 4 pilares	
Objetos	Todo lo que deseemos conectar a Internet, en este caso, los parámetros como son el agua, el calor, la tierra, el movimiento externo, etc.
Datos	La recolección de los parámetros como: la temperatura, humedad, movimiento, pH, etc.
Personas	Personas comprometidas con el proceso, en el presente caso: el agricultor/agricola.
Procesos	La interacción entre el sistema mismo, entre el sistema y el agricultor, y futuras conexiones con empresas.
Interacciones	
M2M	Conexiones entre los dispositivos que permiten comunicarse sin necesidad de un tercero.
M2P	Comunicación entre los dispositivos y las personas inmiscuidas.
P2P	Ya que el agricultor se puede desprender del campo, él puede tener más contacto con personas para mejorar sus ventas y también con sus proveedores.

*Elaboración: Propia*

#### 4.8. Convergencia

Convergir dentro del ámbito de las telecomunicaciones, nos lleva a unir todo los caminos en uno solo, así como en la 2 era digital múltiples servicios pasaban por un mismo canal, ahora en esta nueva era de la misma manera, todos los servicios se dan por el mismo medio, es así que en el presente proyecto, se unifica los diversos servicios, para proporcionar calidad y seguridad dentro del modelo planteado, es aquí donde entonces convergen:

- Uso de la programación, como medio para salir de la red hacia Internet.
- Uso de las API, como herramienta para comunicarse con diferentes servicios.
- Uso de los Webhooks para poder dar seguridad de extremo a extremo en la entrega de datos.
- Uso de códigos para visualizar de manera gráfica los procesos.
- Uso de aplicaciones de terceros para una comunicación directa con el usuario como lo es Pushbullet.
- Uso de servicios de grandes empresas como Google Cloud o Microsoft Azure, que nos permiten mejorar y dar escalabilidad a nuestro sistema.

Como se puede ver la convergencia con Internet del Todo nos ayuda a desarrollar mejores soluciones, en comparaciones a años pasados donde las posibilidades eran más limitadas.

## CONCLUSIONES

**PRIMERO:** Es posible diseñar un modelo de un sistema para el monitoreo de cultivos nativos utilizando Internet del Todo y Redes Fog, sin necesidad de recurrir a dispositivos extremadamente costosos y tener una comunicación inalámbrica sin la presencia de una red LAN, utilizando para ello una comunicación móvil (GPRS, HSPA, LTE), procesada por las plataformas de desarrollo Electron y Arduino, logrando entre ambas una Red Fog, para su posterior diseño de solución basado en el concepto de Internet del Todo y la interacción de los pilares que este requiere, logrando adquirir datos históricos que podrá beneficiar de manera económica al agricultor mediante; la optimización del riego, identificación temprana de deficiencias en el suelo, identificación rápida de lugares que necesiten más atención del agricultor, estimación del rendimiento del producto mediante la anticipación de cambios climatológicos con mayor margen de precisión y de esta forma disminuir las pérdidas que pueda tener el campo antes, durante y después del cultivo, mejorando así el producto para el mercado y tener mejores resultados en todos los ámbitos (uso de analytics).

**SEGUNDO:** Es posible disminuir las pérdidas en los cultivos, mediante el control de parámetros estables de cultivos nativos (variantes según la especie), haciendo uso de la agricultura de precisión (histogramas), logrando según nuestros valores porcentuales presentados en la tabla n°2 una reducción estimada de hasta: 67% por insectos plaga, 30% por aves plaga, 33% por cambios de humedad, 70% por cambios de temperatura y 100% por heladas, esta reducción se logra gracias a que al tener nuestros parámetros monitoreados y controlados será más fácil evitar que la causa de la pérdida se haga presente, sin embargo esto también está sujeto a los métodos y productos que el agricultor/agrícola use como forma de solución. De esta manera estos cultivos puedan ser

aceptados según las exigencias y estándares del mercado internacional, y así poder obtener un producto con valor agregado y la venta total de la producción.

**TERCERO:** Es posible realizar una comunicación remota entre el agricultor y el campo sin la necesidad de que el agricultor este presente, ya que los datos recolectados son llevados en primera instancia a la nube de Particle, para pasar posteriormente con la integración de Webhooks que nos dará seguridad en nuestra información para luego ser visualizada mediante el servicio web de ThingSpeak, para que el agricultor pueda observar de manera gráfica el comportamiento de sus cultivos, accediendo a la página web de ThingSpeak mediante cualquier dispositivos que tenga conexión a internet, así también podrá recibir de manera inmediata mensajes push cuando se presente un cambio o peligro que pueda tener el cultivo, y así el agricultor/agrícola podrá intervenir solo cuando sea necesario (tomando decisiones o estar al tanto de alguna falta o falencia del cultivo) o permitir que el sistema se autocontrole, de esta manera él mismo puede aprovechar y distribuir mejor su tiempo para encontrar más oportunidades para su producto e invertirlo en lo que él considere necesario.



## RECOMENDACIONES

**PRIMERO:** Para mejorar el proyecto se recomienda, prestar constante atención en los módulos y sensores, ya que con el paso de los días salen nuevos y mejores productos que podrían mejorar el funcionamiento y la eficacia del proyecto, así mismo, estar en constante actualización de los firmwares de los dispositivos usados.

**SEGUNDO:** Tener en cuenta la escalabilidad del sistema, para poder adaptar los dispositivos a usar, pudiendo cambiar los modelos a usar de Arduino o el mismo Electron, pudiendo variar según la zona y la accesibilidad que tiene a sistemas de comunicación.

**TERCERO:** Estar en constante desarrollo de nuevos métodos de ahorro de datos, ya que para agricultores grandes, este será un factor importante, o tener convenios con operadoras según el sistema que se esté desarrollando.

## REFERENCIAS

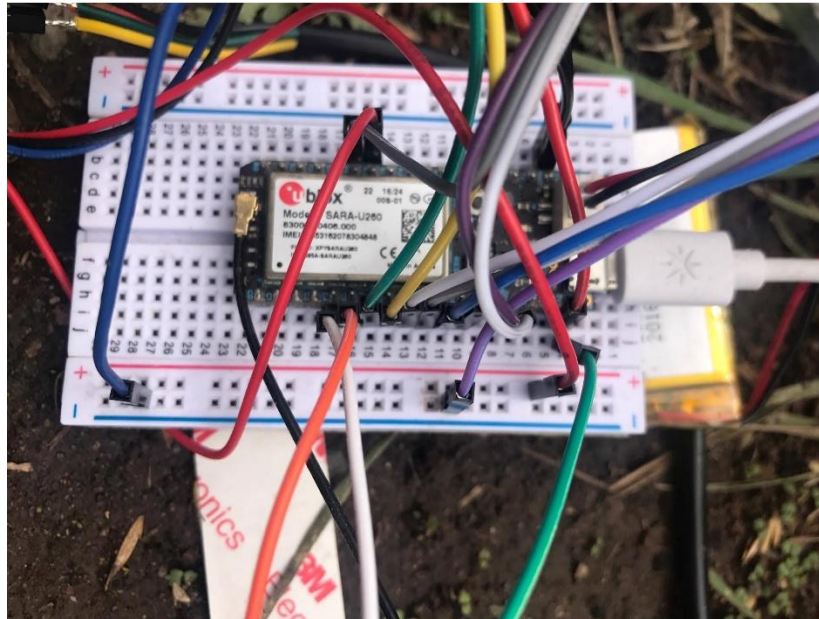
- Abdelwahab, S., Hamdaoui, B., Guizani, M. (2014). Enabling Smart Cloud Services Through Remote Sensing: An Internet of Everything Enabler. IEEEExplore.
- Bacarelli, E., Vinueza, P., Scarpiniti, M. (2017). Fog of Everything: Energy-Efficient Networked Computing Architectures, Research Challenges, and a Case Study. IEEEExplore.
- Delgado, P. (2013). Aves plaga del cultivo de quinua (*Chenopodium quinoa* Willd.) y alternativas de control. Recuperado de Concytec: Contrato de subvención N° 343-2012--CONCYTEC-OAJ.
- Eui-Nam, H., Aazam, M. (2016). Fog Computing: The Cloud-IoT/IoE Middleware Paradigm. IEEEExplore.
- Estrada Jiménez, R., Medina Hinostroza, T., Roldán Chávez, A. (2006). Manual para caracterización in situ de cultivos nativos: conceptos y procedimientos. Perú: Unidad de medio y comunicación – INEIA
- Lavado Mallqui, L. (2011). Metodología de la investigación en ciencia y tecnología Gutenberg en Internet. Lima: Fondo Editorial de la UIGV
- McEwen, A. (2015). Internet de las Cosas – la tecnología revolucionaria que todo lo conecta. España: Anaya Multimedia
- O’Leary, D. (2013). ‘Big Data’, The ‘Internet of Things’ and The ‘Internet of Signs’. Los Angeles: University of Southern California.
- Particle. (2016). Getting Started. Recuperado de <https://docs.particle.io/guide/getting-started/intro/photon/>
- Pejoan, J. (2015). Bits Matemáticos – La teoría de la información y las comunicaciones. España: Estella

- Seng, K., Curtis, M., Chon, T. (2014). Internet of Things: Trends, challenges and applications. IEEEExplore.
- Universidad Católica San Pablo. Peru. Introducción a Internet de Todo. Recuperado de <https://1305435.netacad.com/>
- Villalobos Rebolledo, R. (2010). Determinación de parámetros de riego para el cultivo cebolla de bulbo en el distrito de riego del alto chicamocha. (Tesis de Maestría). Universidad Nacional de Colombia, Bogotá.
- Zanella, A., Bui, N., Castellani, A. (2014). Internet of Things for Smart Cities. IEEEExplore.

# ANEXOS

**Anexo A: componentes del sistema**

**Anexo A1: Electron (Particle)**



Elaboración: propia

**Anexo A2: Arduino (Mega)**



Elaboración: propia



**Anexo A3: Arduino (Nano)**



Elaboración: propia

**Anexo A4: Sensor humedad-temperatura (suelo)**



Elaboración: propia



**Anexo A5: Sensor de Ph (pH electrode)**



Elaboración: propia

**Anexo A6: Modulo del sensor de pH**



Elaboración: propia



**Anexo A7: Sensor de humedad-temperatura (ambiente – DHT22)**



Elaboración: propia

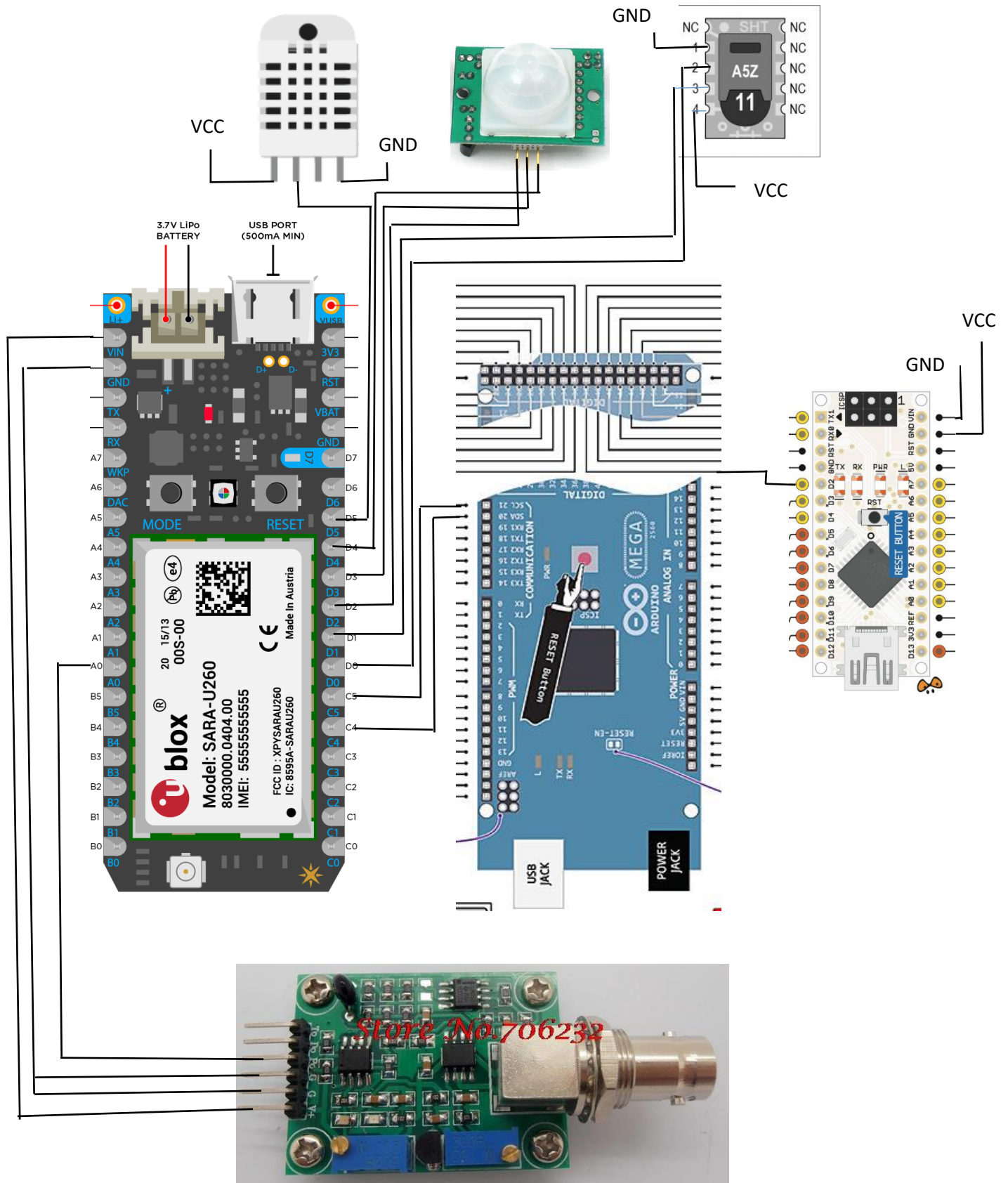
**Anexo A8: Sensor de movimiento (PIR)**



Elaboración: propia



Anexo A9: Diagrama de conexiones del circuito



Elaboración: propia

## Anexo B: Programación de las funciones

### Anexo B1: programación de la comunicación I2C (Electron)

```
char mensaje1;//iniciamos una cadena

void setup()
{
  Wire.begin();//iniciamos el wire (i2c)
  Serial.begin(9600);// iniciamos el serial para mostrar los mensajes
}

void loop()
{
  if(Serial.available())//si hay entrada de datos por serial
  {
    char valor = Serial.read();//leemos la cadena

    if(valor == 'a')//si es "a", podemos colocar otra variable según sea conveniente o un valor
    {
      Wire.requestFrom(1,19);//realizamos una petición al canal 1 de 19 caracteres (bits)
      Serial.println("Mensaje pedido");//imprimimos de que hemos pedido el mensaje al dispositivo en el canal 1(Arduino)
    }
  }
  while(Wire.available())
  {
    int mensaje1 = Wire.read();//guardamos el mensaje en la cadena mensaje 1
    Serial.print(string (mensaje1));//imprimimos el mensaje
    delay(100);
  }
  delay(2500);//esperamos 2 segundos y medio
}
```

Elaboración: propia

**Anexo B2: programación de la comunicación I2C (Arduino)**

```
#include <Wire.h>
void setup()
{
  Wire.begin(1); // Abrimos el canal 1 (0x01) del I2C
  Wire.onRequest(peticion); // Creamos el evento que se realizará cuando el
  Receptor(Electron) llame a el emisor (este Arduino)
  Serial.begin(9600);
}
void loop(){
}
void peticion()
{
  Serial.print("Se pidio el mensaje, Enviando"); // Imprimimos cuando el receptor nos
  pide el mensaje
  Wire.write("Riego activado"); // Imprimimos cuando se realice la petición ese mensaje
  y lo enviaremos por el canal 1
}
Elaboración: propia
```

**Anexo B3: programación PH (Arduino)**

```
int ph_pin = A0; // Pin conectado a Po
float offset = 0.0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  int measure = analogRead(ph_pin); // lectura del sensor
  Serial.print("Measure: ");
  Serial.print(measure);

  double voltage = 5 / 1024.0 * measure; // conversion de valor digital a voltaje
  Serial.print("\tVoltage: ");
  Serial.print(voltage, 3);
  // PH_step = (voltage@PH7 - voltage@PH4) / (PH7 - PH4) // calibracion
  // PH_probe = PH7 - ((voltage@PH7 - voltage@probe) / PH_step)
  float Po = (7 + ((2.5 - voltage) / 0.16)) + offset;
  Serial.print("\tPH: ");
  Serial.print(Po, 3);

  Serial.println("");
  delay(1000);
}
```

Elaboración: propia

**Anexo B4: Programación humedad-temperatura del suelo (Electron)**

```

#include "application.h"
#include "SHT1x.h" // inclusión de la librería del sensor

//SYSTEM_MODE(MANUAL);

//Conexiones:
// SHT_ROJO    ELECTRON_3V3
// SHT_NEGRO    ELECTRON_GND
// SHT_AMARILLO ELECTRON_D1 CLOCK
// SHT_VERDE    ELECTRON_D0 DATA

// Definiendo las conexiones para iniciar el
#define dataPin D0
#define clockPin D1
SHT1x sht1x(dataPin, clockPin);

#define MAX_TEMP    100.0
#define MIN_TEMP    4.0
#define MIN_HUMIDITY 45.0

uint32_t previousMillis = 0;
const uint32_t interval = 1000;
bool state = 0;

int soilPublish(String command);

void setup()
{
  Particle.function("soil", soilPublish); //mandando datos mediante publish
  Serial.begin(9600); //
  Serial.println("Starting up");
  pinMode(D7,OUTPUT);
}

void loop()
{
  float temp_c;
  float temp_f;
  float humidity;

  uint32_t currentMillis = millis();

  // lectura intervalica del sensor
  if (currentMillis - previousMillis >= interval)
  {
    previousMillis = currentMillis;

```

continua ...

```

state =! state;
digitalWrite(D7,state);

// Lectura de los valores del sensor
temp_c = sht1x.readTemperatureC();
temp_f = sht1x.readTemperatureF();
humidity = sht1x.readHumidity();

//verificacion si los valores estan dentro de lo permitido
if(((temp_f > MAX_TEMP) || (temp_f < MIN_TEMP)) ||
(humidity < MIN_HUMIDITY)))
{
  String soil_reading = "{\"temp\": " + String(temp_f,2) + ",\"humidity\": " +
String(humidity,2) + "}";
  //publicacion de los valores
  Particle.publish("alert", soil_reading, 60, PRIVATE);
}

// monitoreo mediante Puerto serial, recomendado antes de publicación por internet
Serial.print("Temperature: ");
Serial.print(temp_c, DEC);
Serial.print("C / ");
Serial.print(temp_f, DEC);
Serial.print("F. Humidity: ");
Serial.print(humidity);
Serial.println("%");
}

}

int soilPublish(String command)
{
  String soil_reading = "{\"temp\": " + String(sht1x.readTemperatureF(),2) +
",\"humidity\": " + String(sht1x.readHumidity(),2) + "}";
  Particle.publish("soil", soil_reading, 60, PRIVATE);
  return 1;
}

```

Elaboración: propia

**Anexo B5: Programación humedad-temperatura ambiente (Electron)**

```

#include <Adafruit_DHT.h> //incluimos la libreria del sensor
#include "math.h"

// Definimos el tipo de sensor o el modelo
#define DHTTYPE DHT22          // DHT 22 (AM2302)

// DHT22 salidas del sensor:
// Pin 1 (izquierda): +3.3V
// Pin 2: output/salida
// Pin 4 (derecha): GROUND
#define DHT_5V_PIN D2
#define DHT_SENSOR_PIN D3
#define DHT_GROUND_PIN D4

DHT dht(DHT_SENSOR_PIN, DHTTYPE);

/* Thingspeak */
//TCPClient client;
//unsigned long myChannelNumber = 184077; //numero de canal
//const char * myWriteAPIKey = "39662E74UQB6RK0D"; // API de escritura

void setup() {
  // Conectando a ThingSpeak
  ThingSpeak.begin(client);

  // Alimentando el sensor
  pinMode(DHT_5V_PIN, OUTPUT);
  pinMode(DHT_GROUND_PIN, OUTPUT);
  digitalWrite(DHT_5V_PIN, HIGH);
  digitalWrite(DHT_GROUND_PIN, LOW);

  // retraso para estabilizar el sensor
  delay(1000);

  // iniciamos el sensor
  dht.begin();

  // lectura del Sensor
  double Temperature = dht.getTempCelcius();
  double humidity = dht.getHumidity();

  // Actualizamos los valores en ThingSpeak con la nueva información
  //ThingSpeak.setField(1, (float)temperature);
  //ThingSpeak.setField(2, (float)humidity);

  // escribimos los valores correspondientes en el canal creado.

```

Continua...

```

ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

//Damos un tiempo para que el mensaje llegue satisfactoriamente a ThingSpeak
delay(5000);

// Colocamos el dispositivo en modo Sleep para el ahorro de energia y datos
//System.sleep(SLEEP_MODE_DEEP, 15 * 60);
}

void loop() {

  double temp = dht.getTempCelcius();
  double hum = dht.getHumidity();
  //double lastTemperature;

  //if (digitalRead(D2) == HIGH) {
  //digitalWrite(D7,HIGH); //mostramos la actividad del sensor
  //if(abs(Temperature-lastTemperature)>2)
  {
    Particle.publish("temp", "{ \"1\": \"" + String(temp) + "\", \"k\":
    \"NRQBNVB3JUUEWU92\" }", 60, PRIVATE); //si deseamos tambien podemos
    publicar en particle, es opcional
    Particle.publish("Hum", String (hum) , 600 ,PRIVATE);
    delay(30000);

    // while (digitalRead(D0) == HIGH); // esperamos que el sensor regrese a su estado
    normal
    //System.sleep(SLEEP_MODE_DEEP, 15 * 60);
    }
    // digitalWrite(D7,LOW);
  }
  // la aplicación retornara porque está en modo sleep

```

Elaboración: propia

**Anexo B6: Programación de detección de movimiento (Electron)**

```

//definimos los pines y sus estados

int inputPin = D5;          // pin de entrada del sensor PIR
int ledPin = D8;           // led de estado
int pirState = LOW;        // se inicia asumiendo que no hay movimiento
int val = 0;               // variable para la lectura del estado del pin del sensor

int calibrateTime = 5000;  // tiempo para calibrar el sensor

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT); // declaracion del sensor como entrada
  Particle.subscribe("hook-response/pushbulletHOME", myHandler, MY_DEVICES);
}

void myHandler(const char *event, const char *data) {

  // integracion con particle.subscribe

}

void loop() {

  // si el sensor esta calibrado
  if (calibrated()) {
    // recogemos informacion del sensor
    readTheSensor();

    // se informa, si el estado ha cambiado
    reportTheData();
  }
}

void readTheSensor() {
  val = digitalRead(inputPin);
}

bool calibrated() {
  return millis() - calibrateTime > 0;
}

void setLED(int state) {
  digitalWrite(ledPin, state);
}

void reportTheData() {

```

Continua....



```

if (val == HIGH) {
  // el estado actual es sin movimiento
  // si el estado cambia, se publicará
  if (pirState == LOW) {
    // el estado cambio a positivo
    Particle.publish("movimiento", "detectado", PRIVATE);
    // actualizamos el estadoE
    pirState = HIGH;
    setLED(pirState);
  }
}
}

```

Elaboración: propia

### Anexo B7: Programación control por internet

```

int led1 = D6; //podemos usar un valor alto o bajo para controlar los sistemas de riego,
              // pH, etc.

```

```

int led2 = D7;

```

```

void setup()

```

```

{
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);

```

```

  Particle.function("led",ledToggle);

```

```

  digitalWrite(led1, LOW);
  digitalWrite(led2, LOW);

```

```

}

```

```

void loop()

```

```

{
}

```

```

int ledToggle(String command) {

```

```

  if (command=="on") {
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    return 1;
  }

```

```

  else if (command=="off") {
    digitalWrite(led1,LOW);
    digitalWrite(led2,LOW);
    return 0;
  }
}

```

```
}  
else {  
    return -1;  
}  
}
```

//escribir esta programación en un editor de texto y guardarlo como un archivo HTML

<!--Reemplazar nuestro device ID y nuestro acces token en las siguientes lineas-->

<!DOCTYPE>

<html>

<body>

<center>

<br>

<br>

<br>

<form action="https://api.particle.io/v1/devices/aqui-va-el-devide-  
ID/led?access\_token=el-acces-token-va-aqui" method="POST">

Decir a nuestro dispositivo que queremos hacer!<br>

<br>

<input type="radio" name="arg" value="on">Turn the LED on.

<br>

<input type="radio" name="arg" value="off">Turn the LED off.

<br>

<br>

<input type="submit" value="Hecho!">

</form>

</center>

</body>

</html>

Elaboración: propia