



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
MONITOREO UTILIZANDO INTERNET DE LAS COSAS Y
REDES MESH PARA LA MEJORA DE LA CALIDAD EN LA
PRODUCCIÓN DE SEMBRÍOS DE QUINUA EN LA CIUDAD DE
JULIACA.**

TESIS

PRESENTADA POR:

Bach. ERICK OLIVER QUISPE MOYA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PUNO – PERÚ

2020



DEDICATORIA

Dedico esta tesis a mi familia

Para el señor Remigio mi papá por todo el apoyo desmesurado que realizo para que pueda hoy dar este paso en mi vida, mi señora madre Amelia por tener toda la paciencia y esperanza en mí y apoyarme en todo sentido, a mi hermana Dyan por estar siempre a mi lado a cada momento, a mi abuelita por ser el motor y estar siempre al tanto de mis acciones y a toda mi familia, amigos y pareja en conjunto porque cada uno apporto en algo para que pueda cumplir hoy con esta meta.

A todos ellos, de todo corazón muchas gracias.

Erick Oliver Quispe Moya



AGRADECIMIENTO

Primeramente, me gustaría agradecer a la prestigiosa Universidad Nacional de Altiplano por haberme abierto sus puertas y así poder formarme profesionalmente, ser especialista competente en el cuidado humano con compromiso social y valores éticos.

También agradecer a los docentes Ingenieros de la Escuela profesional de Ingeniería Electrónica por todas las enseñanzas y consejos recibidos a lo largo de formación profesional y al mismo tiempo a mis compañeros y amigos por haberme compartido angustias, satisfacciones, logros, derrotas y todo tipo de emociones conmigo.

Erick Oliver Quispe Moya



ÍNDICE GENERAL

DEDICATORIA	
AGRADECIMIENTO	
ÍNDICE GENERAL	
ÍNDICE DE FIGURAS	
ÍNDICE DE TABLAS	
ÍNDICE DE ACRÓNIMOS	
RESUMEN	15
ABSTRACT.....	16

CAPITULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA	17
1.2. FORMULACIÓN DEL PROBLEMA.....	17
1.2.1. PROBLEMA GENERAL.....	17
1.2.2. PROBLEMA ESPECÍFICO.....	17
1.3. HIPOTESIS DE LA INVESTIGACION.....	18
1.3.1. Hipótesis General.....	18
1.3.2. Hipótesis Específicos.....	18
1.4. Justificación del Proyecto	18
1.5. Objetivo General.....	20
1.5.1. Objetivos Específicos	20

CAPITULO II

REVISIÓN DE LITERATURA

2.1. MARCO TEÓRICO	22
2.1.1. Antecedentes.....	22
2.1.2. La quinua su descripción, manejo y calidad de producción.....	25
2.1.2.1. Taxonomía.....	25
2.1.2.2. Origen.....	26



2.1.2.3. Requerimiento del cultivo.....	28
2.1.2.4. Morfología de la Quinoa.....	31
2.1.3. Redes Mesh.....	44
2.1.3.1. Protocolos de enrutamiento	47
2.1.3.2. Ventajas de usar Redes Mesh	47
2.1.3.3. Desventajas que existían en Redes Mesh	48
2.1.4. red de Malla Inalámbrica (WMN)	48
2.1.5. Internet de las Cosas (Idc)	50
2.1.5.1. Cosas.....	50
2.1.5.2. Dispositivos del Internet de las cosas	50
2.1.5.3. Protocolos para IdC	51
2.1.5.4. Arquitecturas IdC.....	53
2.1.5.5. Metodologías en IdC.....	54
2.1.5.6. Seguridad de IdC.	54
2.1.6. Agricultura de Precisión	55
2.1.6.1. IdC en la agricultura de precisión.....	56
2.1.6.2. Arquitectura Lambda.....	58
2.1.7. Hardware y sensores.....	58
2.1.7.1. Argón.....	58
2.1.7.2. Xenón.....	61
2.1.7.3. Sensor de Temperatura y humedad: DHT22	63
2.1.7.4. Sensor de movimiento PIR.....	64
2.1.7.5. Sensor de temperatura – Humedad SHT1X.....	65
2.1.7.6. Power Bank.....	66
2.1.8. Software y Plataformas	66
2.1.8.1. Plataforma Particle.....	66
2.1.8.1.1. IDE Web	68
2.1.8.1.2. Particle Tinker.....	69
2.1.8.1.3. Actualizaciones de OTA con Particle	70
2.1.8.1.4. Webhooks.....	70
2.1.8.1.5. API.....	71
2.1.8.1.6. Protocolo de red Thread.....	72



2.1.8.2. Plataforma ThingSpeak.....	73
2.1.8.2.1. Visualizaciones de Matlap.	74
2.1.8.3. GOOGLE CLOUD	76
2.1.8.4. PUSHBULLET.....	77

CAPITULO III

MATERIALES Y MÉTODOS

3.1. Materiales.....	78
3.1.1. Hardware.....	78
3.1.2. Software	78
3.2. TIPO DE DISEÑO DE LA INVESTIGACION.....	79
3.3. POBLACIÓN Y MUESTRA.....	79
3.4. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS.....	79
3.4.1. CAPTURA DE DATOS	80
3.4.1.1 Creación de cuenta en Particle.....	80
3.4.1.2 Descarga de App de particle	80
3.4.1.4. Modos del dispositivo Argon.....	89
3.4.1.5. Configuración del Dispositivo Particle Xenón	91
3.4.1.6. Particle en Windows	97
3.4.1.6.1. Particle CLI.....	97
3.4.2. Instalación del programa Node	97
3.4.1.6.2. Instalación de Particle CLI.	99
3.4.1.6.3. Actualización de Firmware mediante CLI y OTA.....	100
3.4.2. Almacenamiento de datos	102
3.4.2.1 Google cloud.....	102
3.4.3. Procesamiento de Datos.....	113
3.4.3.1. Thingspeak.....	113
3.4.4. Consulta de Datos	116
3.4.4.1 PushBullet.....	116
3.4.4.2. App PushBullet.....	119
3.4.4.3. Plataforma Blynk.	120



CAPITULO IV RESULTADOS Y DISCUSIÓN

4.1. Resultados	123
4.1.1. Resultados de Captura de datos con la Plataforma Particle	123
4.1.1.1. Tiempo de respuesta, latencia y adquisición de datos	125
4.1.2. Almacenamiento de datos con Google Cloud.....	127
4.1.3 Análisis de datos con ThingsSpeak	129
4.1.4. Consulta de datos con PushBullet y Blynk.....	132
4.2. Discusión.	136
4.2.1. Diseño de Monitoreo Económico y sostenible con redes Mesh.	138
4.2.2. Porque Utilizar una red Mesh inalámbrica para IdC.	138
4.2.3. Inconvenientes con el uso del Argon.	139
4.2.4. Convergencia	140
V. CONCLUSIONES.....	142
VI. RECOMENDACIONES.....	144
VII. REFERENCIAS.....	146
ANEXOS.....	150

Área : Telecomunicaciones y redes de datos

Tema : Internet de las cosas y Mesh Networks

FECHA DE SUSTENTACIÓN: 15 de enero de 2020.



ÍNDICE DE FIGURAS

Figura 1: Campos de Quinua en Puno.	31
Figura 2: Fotografía de planta de quinua atacada por Mildiu.....	37
Figura 3: Fotografía de defoliación en el cultivar.....	37
Figura 4: Fotografía en el haz y envés de la hoja (Peronospora farinosa).....	39
Figura 5: Escala de evaluación para mildiu	40
Figura 6: Ciclo de vida de Peronospora farinosa	41
Figura 7: Modelo OSI - 7 capas	45
Figura 8: Topología de Estrella y Mesh.....	46
Figura 9: Hardware de Particle.	49
Figura 10: Dispositivos IdC	51
Figura 11: Siete(7) capas de Internet de las Cosas	53
Figura 12: Diagrama de una Arquitectura Lambda	58
Figura 13: Dispositivo Argón	59
Figura 14: Dispositivo Xenon.....	61
Figura 15: Sensor DHT22.....	63
Figura 16: Conexión usando protocolo Thread	73
Figura 17: Procesamiento de imágenes con Matlap	75
Figura 18: Creación de cuenta Particle	80
Figura 19: Descarga de App Particle	81
Figura 20: Login con a App de particle	81
Figura 21: Agregar el dispositivo Aron a nuestra cuenta Particle	82
Figura 22: Selección de dispositivo a agregar	83
Figura 23: Activación e Bluetooth en smartphone	83
Figura 24: Activando Bluetooth.	84



Figura 25: Escaneo de Código QR del Argon.	84
Figura 26: Activando permisos de Ubicación.....	85
Figura 27 Escaneo de QR del argón.	85
Figura 28: Emparejando Argón al Smartphone mediante Particle	86
Figura 29: Condiciones para crear la Red Mesh.....	86
Figura 30: Redes disponibles para conectar en dispositivo Argon.....	87
Figura 31 Colocación de Nombre ala red Mesh	87
Figura 32: Nombre de la red Mesh	88
Figura 33: Ingreso de clave de red Mesh.....	88
Figura 34: Modo Conectado a red wifi.....	89
Figura 35: Actualización de firmware de OTA:	89
Figura 36: Buscando red Mesh creada.....	90
Figura 37: Incorporación de dispositivo Xenón a la red Mesh.....	91
Figura 38: Emparejamiento de dispositivo Xenon.	92
Figura 39: Conexión exitosa del Xenon.	92
Figura 40: Actualizaciónde dispositivo de Xenon.....	93
Figura 41: Actualizando Updating.....	93
Figura 42: Visualización de la red Mesh	94
Figura 43: Incorporando dispositivo Xenon a la red Mesh	94
Figura 44: Conexión exitosa de Red Mesh.....	95
Figura 45: Ingreso de clave.....	95
Figura 46: Cargando Requisitos para ingresar a la red Mesh.....	96
Figura 47: Mensaje de ingreso satisfactorio a la red Mesh.....	96
Figura 48: Creación de xenon de acuerdo a la cantidad de sensores a utilizar	97
Figura 49: Descarga de Node.js, paso 1.....	98



Figura 50: Permisos de Instalación	98
Figura 51: Descarga de Particle CLI, paso 1	99
Figura 52: Ejecución del setup de Particle CLI, paso 2.....	99
Figura 53: Ingreso a Particle del CLI de Windows	100
Figura 54: Ingreso a la cuenta de Particle a partir del CLI de Windows.....	100
Figura N° 55: Ingreso a cuenta de particle mediante línea de comandos del CLI exitosa	101
Figura 56: Actualización de comandos CLI.	101
Figura 57: Actualización de Firmware mediante OTA	102
Figura 58: Ingreso a la página de GoogleCloud	103
Figura 59: Creación de cuenta en Google Cloud.....	103
Figura 60: Ingreso de datos de tarjeta.....	104
Figura 61: Creación de cuenta satisfactoria.....	104
Figura 62: Creación de nuevo proyecto o Topic.....	105
Figura 63: Creación de tema en Pub/Sub.....	105
Figura 64: Agregar miembro y función al Tema	106
Figura 65: Insertamos una nueva Integración.....	106
Figura 66: Creación de Integración	107
Figura 67: Código para suscribirme a mi propio tema	107
Figura 68: Verificación de llegada de datos	108
Figura 69: Llegada de datos de Google Cloud	108
Figura 70: Proceso de almacenamiento en el Datastore	109
Figura 71: Creación de credenciales.....	109
Figura 72: Archivo Json de Seguridad.....	110
Figura 73: Ingreso Pub/Sub Suscripciones	110



Figura 74: Creación de Suscripción.....	111
Figura 75: Creación de suscripción de almacenamiento	111
Figura 76: Error de Clonación en Windows	112
Figura 77: Descarga de Git para Windows	112
Figura 78: Clonación de Repositorio con GIT.....	113
Figura 79: Creación de cuenta Thingspeak.....	113
Figura 80: Creación de Canal y campos	114
Figura 81: Visualización de los canales creados	114
Figura 82: Visualización de ID del canal.....	115
Figura 83: Código API.....	115
Figura 84: Cuenta creada de PushBullet.....	116
Figura 85: Creación de nuevo canal	116
Figura 86: Creación de Acces token	117
Figura 87: Creación de Webhook	117
Figura 88: Creación de Webhook	119
Figura 89: Descarga de PushBullet.....	119
Figura 90: Permiso de Acceso a SD	119
Figura 91: Permiso de Acceso a Contactos	120
Figura 92: Herramientas de Blynk.....	121
Figura 93: Creación de un botón Virtual.	121
Figura 94: Activación de sistema de riego.....	122
Figura 95: Notificación de riego activado	122
Figura 96: Creación de la red Mesh.....	123
Figura 97: Dispositivos conectados a la Red Mesh	124
Figura 98: Captura de datos del sensor DTH22.....	124



Figura 99: detección de Movimiento con el sensor PIR.....	125
Figura 100: Opciones de "dispositivos" en la consola de Particle.....	125
Figura 101 latencia del servidor,.....	127
Figura 102 Latencia del algoritmo y la programación del Gateway.....	127
Figura 103 Verificación de llegada de datos	128
Figura 104 Llegada de datos de Google Cloud.....	128
Figura 105 Medición de la Temperatura del suelo	129
Figura 106 Histograma de la Temperatura	130
Figura 107 Medición de la Humedad del Suelo	130
Figura 108 Histograma de la Humedad	131
Figura 109 Correlación entre la Humedad y la temperatura ambiente del suelo.....	131
Figura 110 Detección de Movimiento	132
Figura 111 Recepción de temperatura y humedad.....	132
Figura 112 Pin de encendido	133
Figura 113 Mensaje de Activación	133
Figura 114 Led encendido	134
Figura 115 Ubicación de aplicación de Monitoreo.....	140



ÍNDICE DE TABLAS

Tabla 1	Taxonomía de la Quinua	26
Tabla 2:	Características de antena del Argón	90
Tabla 3.	Velocidad de transferencia entre dispositivos Mesh,	126
Tabla 4:	En la siguiente tabla podemos observar los resultados obtenidos en base a los objetivos planteados donde en algunos casos fueron óptimos y el otros no tan favorables.	134
Tabla 5:	Presupuesto Mínimo para la implementación del sistema de Monitoreo.....	139
Tabla 6	Enfoque para selección de rol de dispositivos	144



ÍNDICE DE ACRÓNIMOS

Idc	: Internet de las Cosas
FAO	: Organización de las Naciones Unidas para la Agricultura y la Alimentación
OSI	: Open Systems Interconnection
WMN	: Wireless Mesh Network/ Red de Malla Inalámbrica
MQTT	: MQ Telemetry Transport
AMQP	: Advanced Message Queuing Protocol
WAMP	: Web Application Messaging Protocol/
CoAP	: Constrained Application Protocol/Protocolo de aplicación restringido
STOMP	: Streaming Text Oriented Messaging Protocol
XMPP	: Extensible Messaging and Presence Protocol
WMQ	: WebSphere MQ
BLE	: Bluetooth Low Energy/ Bluetooth de baja energía
IDE	: Integrated Development Environment/ Entorno de desarrollo integrado
API	: Application Programming Interfaces / Interfaces de programación de aplicaciones
CLI	: Command line interface / Interfaz de línea de comandos



RESUMEN

En esta tesis presentamos un diseño de una red Mesh con internet de las cosas para monitorizar los sembríos y así tener una mejor calidad de producción de Quinua, en la ciudad de Juliaca, Región Puno. Con la finalidad de obtener calidad de producción, identificaremos con conceptos básicos las causas principales que lleva a la pérdida de cultivo de Quinua; se utilizara la tecnología para evitar y reducir dichas perdidas, teniendo así la unión de estos dos aspectos: Tecnología y Agricultura daremos resultado a la “Agricultura de Precisión”. Para lograr dicha Agricultura, desarrollaremos también conceptos generales de internet de las cosas y redes Mesh, que nos permitirá comunicar y monitorizar los sembríos de Quinua con el agricultor sin ser necesaria su presencia, también el monitoreo a través de los sensores y actuadores nos permitirá que el sembrío tome sus propias decisiones, y así asegurar una calidad de sembríos de Quinua. En el proceso se hará uso de la plataforma Particle con el uso de sus prototipos que son el argón y xenón; el Particle argón se usara como puerta de enlace o punto final para la comunicación de la red Mesh, dicha red estará formada por distintos sensores que se comunicaran entre sí con la ayuda del Particle Xenón, para así recolectar datos, procesarlos y visualizarlos en cualquier dispositivo que tenga acceso a internet. Con breves análisis y ciertos criterios, la investigación será aplicada con diseño de investigación Mixta y método Experimental.

Palabras Clave: Internet de las cosas, Redes Mesh, Agricultura de Precisión, Particle argón y Xenón.



ABSTRACT

In this thesis we present a design of an Mesh network with internet of things to monitor crops and thus have a better production quality of Quinoa, in the city of Juliaca, Puno Region. In order to obtain production quality, we will identify with basic concepts the main causes that lead to the loss of Quinoa cultivation; the technology will be used to avoid and reduce these losses, thus having the union of these two aspects: Technology and Agriculture we will give result to the “Precision Agriculture”. To achieve this Agriculture, we will also develop general concepts of internet of things and Mesh networks, which will allow us to communicate and monitor the Quinoa crops with the farmer without their presence being necessary, also monitoring through sensors and actuators will allow us to the planting makes its own decisions, and thus ensure a quality of Quinoa crops. In the process, the Particle platform will be used with the use of its prototypes that are argon and xenon; the argon particle will be used as a gateway or end point for the communication of the Mesh network, said network will be formed by different sensors that will communicate with each other with the help of the Xenon Particle, in order to collect data, process and visualize it on any device that has access to the Internet. With brief analysis and certain criteria, the research will be applied with mixed research design and Experimental method.

Keywords: Internet of things, Mesh Networks, Precision Agriculture, Argon Particle and Xenon.



CAPITULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

Asumiendo que la gran demanda de la Quinua es para la exportación y en su gran mayoría es por Quinua orgánica, se realiza este proyecto para alcanzar una buena calidad de producción en los sembríos de Quinua erradicando los problemas que afectan a los cultivos de los mismos, y así alinear las prácticas agrícolas de su producción a las exigencias del mercado nacional e internacional, utilizando la tecnología como solución.

Cabe señalar que los lineamientos y pautas contenidas en este proyecto, están orientadas a los productores de la sierra, exactamente en la ciudad de Juliaca.

1.2. FORMULACIÓN DEL PROBLEMA

1.2.1. PROBLEMA GENERAL.

¿Es posible diseñar e implementar un sistema de monitoreo utilizando internet de las cosas y redes Mesh para la mejora de la calidad en la producción de sembríos de quinua en la ciudad de Juliaca?

1.2.2. PROBLEMA ESPECÍFICO.

¿Es posible Diseñar un sistema de monitoreo utilizando Internet de las Cosas y redes Mesh, para mejorar la calidad de producción en los sembríos de Quinua en la ciudad de Juliaca, en el año 2020?

¿Es posible Implementar un sistema de monitoreo utilizando Internet de las cosas y redes Mesh, para mejorar la calidad de producción en los sembríos de Quinua en la ciudad de Juliaca?



¿Es posible Monitorizar, utilizando integraciones de la plataforma Particle para mejorar la calidad de producción en los sembríos de Quinoa en la ciudad de Juliaca?

¿Es posible Utilizar Internet de las cosas para la comunicación entre la agricultura y la tecnología?

1.3. HIPOTESIS DE LA INVESTIGACION

1.3.1. HIPÓTESIS GENERAL

Es posible diseñar e implementar un sistema de monitoreo utilizando Internet de las Cosas y redes Mesh y así mejorar la calidad de producción en los sembríos de Quinoa e la ciudad de Juliaca

1.3.2. HIPÓTESIS ESPECÍFICOS

Es conveniente el uso de un sistema basado en una red Mesh e internet de las cosas, para mejorar la calidad de los sembríos de quinoa

Es posible Implementar un sistema de monitoreo utilizando IoC y redes Mesh, para mejorar la calidad de producción en los sembríos de Quinoa.

Es posible Monitorizar los cultivos para mejorar la calidad de producción de la Quinoa.

Es posible lograr la comunicación ente la agricultura y la Tecnología

1.4. JUSTIFICACIÓN DEL PROYECTO

Actualmente nuestro país es uno de los más conocidos por su alto potencial de producción y exportación de Quinoa, siendo la zona sur del Perú la que más producción de quinoa genera anualmente, pero los mismos no están libres de perdidas, si no se les hace un monitoreo personalizado durante todo el proceso de la cosecha (Palacios, 2017). Por lo que uno de los propósitos de este proyecto es monitorizar los cultivos de Quinoa utilizando internet de las cosas y redes Mesh para así obtener gran cantidad de datos sobre



el suelo y propiedades de cultivo, por medio de los sensores, y teniendo esos datos, procesarlos para evitar pérdidas y adecuar parámetros requeridos, para la producción de calidad de la Quinoa.

Para tener una arquitectura de Agricultura de Precisión soportada en Idc se plantean cuatro (4) capas las cuales son: La capa de captura de datos está formada por un conjunto de sensores cuyo objetivo es la obtención de variables climáticas asociadas al cultivo. La capa de almacenamiento se encarga de recopilar información en tiempo real a partir de las diferentes fuentes (sensores) distribuida en el cultivo. La capa de procesamiento tiene como función la predicción de información relevante a partir de los datos recopilados, Finalmente, la capa de consulta permite la visualización y seguimiento en tiempo real de las variables del cultivo y de las predicciones obtenidas en la capa de procesamiento. (Quiroga, Jaramillo, Campo, y Chanchí, 2017). En este proyecto se utilizarán las mismas 4 capas: Con la ayuda de Particle Xenón, se capturarán datos de 4 sensores, luego dicha información, con la ayuda de Particle Argón se almacenará en la plataforma Googlecloud. para luego ser procesada y graficada por la plataforma Thingspeak y luego visualizar los datos generados por los sensores en cualquier dispositivo conectado a internet con la ayuda de App PushBullet otros servicios de tercer, para poder finalmente tener el control de nuestro sistema a través de Blynk.

Así mismo innovar el uso de la tecnología en el sector agrícola de la región, con soluciones nuevas y amigables que hagan posible una gran mejora en la agricultura; consecuentemente también unir dos conceptos muy antagónicos entre sí, la Tecnología y la Agricultura.



Teniendo por un lado la Quinoa en su forma natural y por otro la tecnología con este gran concepto Idc: Internet de las cosas, que permitirá la interacción entre la agricultura (naturaleza) y la tecnología, nace un nuevo concepto en el sector agrícola de la región.

Formando así un nuevo concepto en la región, “la agricultura de precisión”, en la que se monitorearan los cultivos de Quinoa y la presencia del agricultor no intervendrá directamente

Muy distinto a como antes se practicaba la agricultura, ahora se tendrá datos más exactos que permitirán dar una estimación a futuro de los factores principales que conllevan a la pérdida de cultivos de Quinoa y así tomar mejores decisiones y prever posibles pérdidas a futuro, todo esto para tener por consecuencia una mejora de calidad de producción en los sembríos de Quinoa.

1.5. OBJETIVO GENERAL

Diseñar e implementar un sistema de monitoreo utilizando Internet de las Cosas y redes Mesh para mejorar la calidad de producción en los sembríos de Quinoa en la ciudad de Juliaca.

1.5.1. OBJETIVOS ESPECÍFICOS

Diseñar un sistema de monitoreo utilizando Internet de las Cosas y redes Mesh, para mejorar la calidad de producción en los sembríos de Quinoa en la ciudad de Juliaca, en el año 2020

Implementar un sistema de monitoreo utilizando Internet de las cosas y redes Mesh, para mejorar la calidad de producción en los sembríos de Quinoa en la ciudad de Juliaca,

Monitorizar, utilizando integraciones de la plataforma Particle para mejorar la calidad de producción en los sembríos de Quinoa en la ciudad de Juliaca,



Utilizar Internet de las cosas para la comunicación entre la agricultura y el usuario final a través del uso de la tecnología de bajo costo.



CAPITULO II

REVISIÓN DE LITERATURA

2.1. MARCO TEÓRICO

2.1.1. ANTECEDENTES

En la Investigación “GUIA DE CULTIVO DE QUINUA” Indica que en que la práctica del cultivo de quinua va desde Sud América en zonas geográficas que van desde el nivel del mar hasta los 4000 m.s.n.m., en zonas con precipitaciones de 0 a 1000 mm, en suelos de diferentes texturas y con un rango de pH que fluctúa entre 4 a 9. En un rango de temperaturas debajo de cero a más de 30°C.(Gomez & Aguilar, 2016)

Así mismo afirma que de estas condiciones variables de clima los estreses más frecuentes son la sequía, las heladas, la salinidad, las plagas y otros factores.

Y finalmente, concluye que la tecnología usada en su cultivo es bastante variable, desde aquella tradicional hasta aquella moderna altamente tecnificada. Dependiendo de la interacción de estos factores de clima, suelo y tecnología los rendimientos varían de 1 a 7 t/ha.(Gómez y Aguilar, 2016)

En la investigación “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL MONITOREO DE CULTIVOS NATIVOS UTILIZANDO INTERNET DEL TODO Y REDES FOG” en esta tesis indica que un modelo de sistema de monitoreo utilizando internet del todo y redes Fog es posible de implementar sin necesidad de recurrir a dispositivos extremadamente costosos y tener comunicación inalámbrica sin presencia de una red LAN, también afirma que si es posible disminuir pérdidas de cultivo mediante el control de parámetros estables de cultivos nativos, haciendo uso de la agricultura de precisión, dichas reducción asegura el autor se logran gracias a que se tienen dos parámetros monitorizados y controlados(Palacios, 2017).



Así mismo en esta tesis el autor llegó a a conclusión que es posible realizar la comunicación remota entre el agricultor y el campo sin necesidad de que el mismo esté presente, ya que los datos son llevados en primera instancia a la nube y así el agricultor puede aprovechar y distribuir mejor su tiempo para encontrar más oportunidades para su producto e invertirlo en lo que considere necesario (Palacios, 2017); con esta investigación se corrobora que se puede lograr la monitorización de los sembríos, utilizando la tecnología.

En la investigación “UBICACIÓN DE RECURSOS DE LA PLATAFORMA DE ANALISIS DE DATOS EN FOG COMPUTING ” hace referencia a las ventajas que tiene una red FOG, la minimización de latencia de servicio y el tiempo medio de respuesta de las aplicaciones, para una mejor experiencia del usuario, esta investigación busca desarrollar una colocación consiente de los recursos de la plataforma de análisis de datos en la arquitectura de computación niebla, para reducir ampliamente los costes de red y el tiempo de respuesta para el usuario. (Taneja & Davy, 2016)

En la investigación “Sistema de Riego basado en internet de las cosas (Idc)” en este artículo menciona que con el pasar de los años, la situación respecto al agua irá empeorando creando problemáticas en todos los campos, especialmente en el sector agrícola, es por esto que urge una solución eficiente para el uso racional del agua en la agricultura, dando como solución el uso de internet de las cosas para poder monitorizar los campos de agricultura.(Yachay, Castro, Espinosa, y Anaya, 2017); teniendo en cuenta que una de las finalidades de la presente tesis es activar un sistema de riego, para disminuir gastos de agua; con esta investigación se puede reafirmar que se puede lograr una comunicación utilizando Idc para así poder activar o desactivar sistemas de riego solo cuando el sembrío lo necesite.



En la Investigación “SgreemH-Idc: Plataforma Idc para la agricultura de precisión” en esta investigación se presenta SgreemH-Idc una plataforma Idc a bajo costo y consumo energético para la monitorización de campos de cultivos e invernaderos. En La plataforma la conforman una arquitectura de 4 capas un protocolo de comunicación, el diseño de un nodo a bajo costo consumo de energía y aplicación web para visualización de dato. Los resultados y conclusiones obtenidos mediante la experimentación muestran el funcionamiento efectivo de la plataforma para la recolección de datos y la interacción con la plataforma de decisiones con los porcentajes de perdida de datos del cero por ciento.(Guerrero et al., 2017); Asumiendo que para la recolección de datos se necesitaran plataformas gratuitas, en este proyecto nos da una alternativa de la plataforma SgreemH-Idc, que es una plataforma gratuita para evitar pérdidas de datos generados por los sensores que se instalaran en los sembríos de Quinoa.

En la investigación Nuevo Protocolo Thread 1.2 de IdC y Hogares Inteligentes Habla de la versión de protocolo de comunicación inalámbrica “Thread 1.2” que trae como una de las nuevas ventajas la escalabilidad y la adición simple y segura de dispositivos de anuncios de nodos.(Secureweek, 2020), para este proyecto se utiliza esta versión de comunicación inalámbrica, logrando efecto dramático en cuanto a la duración de la batería y teniendo una escalabilidad amplia de dispositivos como end points.

Quiroga et al. (2017) en esta articulo demuestra la gran flexibilidad y funcionalidad que ofrece la combinación de herramientas hardware de bajo costo y software libre, se ha conseguido el diseño, implementación y despliegue de una arquitectura soportada en IoT, la cual se aborda desde tres vistas arquitectónicas de modelo de negocio, funcionalidad e implementación en un escenario de agricultura de precisión. Además, se despliegan cuatro capas de operación comprendiendo la captura de las variables climáticas, el almacenamiento de esta información, su procesamiento y su consulta; sirviendo como



referencia para los agricultores y la comunidad interesada en utilizar nuevas tecnológicas, técnicas y servicios basados en IoT que buscar mejorar la productividad agrícola.

2.1.2. LA QUINUA SU DESCRIPCIÓN, MANEJO Y CALIDAD DE PRODUCCIÓN.

Haciendo una investigación se encontró las siguientes peculiaridades de la quinua en cuanto a su descripción, manejo y producción.

2.1.2.1. TAXONOMÍA.

La quínoa es una hierba anual, Dicotiledónea, que alcanza una altura de 0,2 a 2,0 m. Las plantas, así como sus frutos y semillas, presentan una alta diversidad de colores que van desde verde, morado a rojo y colores intermedios entre éstos.(León-lobos, Morales, Ruf, Zurita, y Alfaro, 2018).

El tallo principal puede ser ramificado o no, dependiendo del genotipo, densidad de siembra y de las condiciones del medio en que se cultive.

Sus hojas son dentadas, polimórficas; las basales son grandes y pueden ser romboidales o triangulares, mientras que las hojas superiores generalmente alrededor de la panoja son lanceoladas.

La inflorescencia es una panoja con un eje principal más desarrollado, del cual se originan los ejes secundarios y en algunos casos terciarios. Las flores son muy pequeñas y densas, se agrupan en glomérulos, son sésiles y, pueden ser hermafroditas, pistiladas o androestériles (León-lobos et al., 2018).

El fruto es un aquenio indehisciente que contiene un grano que puede medir hasta 2,66 mm de diámetro. El pericarpio es la capa externa del fruto que presenta alveolos. Contiene saponina, en distinta concentración dependiendo de la variedad o eco tipo. Sus colores

son variados y pueden ser translúcidos, blancos, cremas, amarillos, anaranjados, rosados, rojos, púrpuras, marrones, grises y negros, entre otros.

La episperma, que envuelve a la semilla, está compuesta por cuatro capas. La externa, determina el color de la semilla, es de superficie rugosa, quebradiza y se desprende fácilmente con agua. Es de color similar al pericarpio (León-lobos et al., 2018).

Tabla 1 Taxonomía de la Quinua

Reino: vegetal	Familia: Chenopodiceas
División: fanerógamas	Genero: Chenopodium
Clase: Dicotiledoneas	Seccion: Chenopodia
Sub-Clase: Angiospermales	Subseccion: Celluta
Origen: Centrospermales	Especie: Chenopodium quinua

Fuente: León-lobos et al. (2018)

2.1.2.2. ORIGEN.

Es un cultivo muy antiguo de los andes, en 1970 el historiador Núñez indica que al norte de Perú en un complejo Arqueológico, encontró granos de quinua que datan de 3000 A.C. Max Hule en 1919, historiador peruano indica que la quinua tiene una antigüedad de 5000 años A.C. en forma general, se puede indicar que en diferentes lugares donde se han encontrado estos granos de quinua al ser analizados mediante el C.14 ratifican esta antigüedad. (León-lobos et al., 2018),.

La singularidad encontrada es que mientras más antigua sea la semilla, se encontraran un mayor porcentaje de semillas de quinuas silvestres o ayaras (grano negro), lo que indica que el proceso de selección ha tenido varios siglos para poder lograrse una variedad.



Se atribuye su origen a la zona del Altiplano Peruano-boliviano, por estar caracterizada por la gran cantidad de especies silvestres y la gran variabilidad genética, principalmente en eco tipos, reconociéndose cinco categorías básicas.

Las quinuas de los valles crecen en los valles interandinos de 2000 a 3600 m.s.n.m, se caracterizan porque tienen gran desarrollo, pueden llegar a 2 a 2.5 m de altura, son ramificadas, su periodo vegetativo es largo con panojas laxas, con inflorescencia amarantiforme, son tolerantes al mildiu, en este grupo tenemos a la blanca de Junín, amarilla de Marangani y rosada de Junín.

Las quinuas altiplánicas, crecen en lugares aledaños al lago Titicaca a una altura de 3 800 m.s.n.m. estos cultivos se caracterizan por tener buena resistencia a las heladas, son bajos de tamaño, no ramificados (tienen un solo tallo y panoja terminal que es glomerulada densa), llegan a tener una altura de 1.00 a 2.00 m., con periodo vegetativo corto, se tiene quinuas precoces como: Illpa-INIA y Salcedo-INIA, semitardías: blanca de Juli, tardías: como la Kancolla, chewecca, tahuaco, Amarilla de Marangani.

Las quinuas de los salares de Bolivia son salares, como su nombre lo indica son resistentes y se adaptan a suelos salinos y alcalinos, los granos son amargos y tienen alto porcentaje de proteínas miden de 1 a 1.5m. de altura, presentan un solo tallo desarrollado; tenemos: la real boliviana, ratuqui, rabura, sayaña (variedades del altiplano boliviano).

Las quinuas al nivel del mar, crecen en el Sur de Perú, son en su generalidad no ramificadas y los granos son de color amarillo a rosados y a su vez amargas, como en el sur de Perú en Concepción, las quinuas se caracterizan por tener una foto de periodo largo y la coloración de los granos de color verde intenso y al madurar toman una coloración anaranjada y los granos son de tamaño pequeño y de color blanco y anaranjado.



Las quinuas sub-tropicales, crecen en los valles interandinos de Bolivia, se caracterizan por ser plantas de color intenso y al madurar toman una coloración anaranjada y los granos son de tamaño pequeño y de color blanco o anaranjado.(León-lobos et al., 2018)

2.1.2.3. REQUERIMIENTO DEL CULTIVO

Factores ambientales: la región, las condiciones climáticas y el suelo tienen influencias muy marcadas en la producción y productividad de la quinua. El clima está determinado por una serie de factores tales como altitud, precipitación, temperatura, latitud, vientos, iluminación, etc. Dado a su cultivo en zonas marginales de los andes, la quinua se enfrenta con altos riesgos ambientales como la helada, sequías prolongadas, granizo, vientos fuertes, suelos pobres de ácido. En lo siguiente se detallan las tolerancias u necesidades de la quinua frente a estos factores ambientales.

Región: La quinua fue domesticada y sembrada durante miles de años en zonas que van desde el nivel de mar o costa (0 a 500 m.s.n.m.), la yunga (500 hasta 2500 m.s.n.m.); sierra media – zona quechua o valles interandinos (2500 – 3500 m.s.n.m.) y hasta la sierra alta, Suni o Altiplano (3500 a 4000 m.s.n.m.); dando lugar al surgimiento de diversos tipos de quinuas llamados ecotipos y de los cuales deben ser elegidas las variedades a sembrar; para lograr una buena productividad y calidad de granos. (Gomez y Aguilar, 2016)

Suelo: En lo referente al suelo la quinua prefiere de un suelo franco arenoso a franco arcilloso, con buen drenaje, con pendientes moderadas, con profundidad promedia y un contenido medio de nutrientes, puesto que la planta depende de los nutrientes aplicados al cultivo anterior que es generalmente papa. La quinua se adapta a diferentes tipos de suelos.



Agua: En cuanto a la precipitación se considera óptimo de 300 a 500 mm, y como máximo de 600 a 800 mm. En cuanto al agua, la quinua es un organismo eficiente en el uso, a pesar de ser una planta C3, puesto que posee mecanismos morfológicos, anatómicos, fenológicos y bioquímicos que le permiten no solo escapar al déficit de la humedad, sino tolerar y resistir la falta de humedad del suelo en los años más o menos seco de 300 a 500 mm de agua, pero sin heladas se obtiene buena producción.

Temperatura: La temperatura óptima para la quinua este alrededor de 8-15°C, puede soportar hasta -4°C, en diferentes etapas fenológicas siendo más tolerantes en la ramificación y las más susceptibles la floración y llenado de grano. La temperatura está determinada por la altura, la inclinación y exposición del campo y por la densidad del cultivo. La única posibilidad del productor de influir sobre la temperatura es mediante la selección del campo bien ubicado y de la densidad de la siembra. Para una germinación aceptable la temperatura mínima para la quinua es de 5°C. Temperaturas mayores a 15°C, causan pérdidas por respiración, traen el riesgo de ataques de insectos (si las condiciones son secas) u hongos (si las condiciones son húmedas). La presencia de veranillos de la Pantoja y su maduración, lo que repercute en bajos rendimientos.(Gomez y Aguilar, 2016)

Humedad-precipitación: Si las siembras se realizan, en presencia de precipitaciones pluviales y/o temporadas de lluvias, la mayor parte de la semilla debe ir a los laterales de los surcos y no al fondo del surco vana a tener problemas de la pudrición por la excesiva humedad.

En el caso específico de la quinua, los sistemas de preparación de terreno y siembras son diferentes al resto de las siembras y preparaciones de terreno de otras regiones de la Sierra del país.



Sequia: La quinua soporta épocas de sequía prolongada hasta 60 días excepto en los estados fenológicos de germinación hasta las 4 hojas verdaderas, floración, madurez de estados lechosos. Durante estas fases necesita casi 5mm/día para un abastecimiento óptimo.

Fotoperiodo: La quinua ha sido domesticada y cultivada desde tiempos ancestrales en una región comprendida entre 5°N (Colombia) a 40°S (Perú y Argentina), y desde el nivel del mar hasta los 4000 m.s.n.m. aproximadamente. La respuesta al fotoperiodo y a la temperatura está relacionada con el lugar de origen, es muy compleja y puede afectar el rendimiento. Variedades que se originan en el trópico se caracterizan por una mayor sensibilidad al fotoperiodo y por una larga fase hasta antesis. Las variedades del Altiplano de Perú y Bolivia y las quinuas del nivel del mar son las de menor sensibilidad al fotoperiodo y son las que tienen menor longitud del ciclo a antesis. La duración del ciclo a antesis también está influenciada por la altitud sobre el nivel del mar de la zona de origen de la quinua.

En el proceso de la introducción de las variedades de quinua a nuevas áreas, es importante considerar la zona de origen de las variedades, es decir la latitud y la altitud de donde proceden. Por ejemplo, las variedades ecuatorianas necesitan por lo menos 15 días con 10 horas de luz cada día para llegar a antesis. Se recomienda, en general, evitar épocas con alta temperatura y días largos por que afectan el proceso de formación de los granos y por consiguiente el rendimiento.(Gómez y Aguilar, 2016)



Figura 1: Campos de Quinoa en Puno.
Fuente: (Gomez y Aguilar, 2016)

2.1.2.4. MORFOLOGÍA DE LA QUINUA.

Raíz

La raíz de quinoa es del tipo pivotante, consta de una raíz principal de la cual salen un gran número de raíces laterales muy ramificadas. La longitud de las raíces es variable, de 0.8 a 1.5 m. Su desarrollo y crecimiento está determinado por el genotipo, tipo de suelos, nutrición y humedad entre otros factores.(FAO & CIRAD, 2013)

Tallo

El tallo en la unión con el cuello de raíz es cilíndrico y a medida que se aleja del suelo se vuelve anguloso en las zonas de nacimiento de hojas y ramas. La corteza es firme y compacta formada por tejidos fuertes y lignificados. Cuando los tallos son jóvenes la médula es suave, cuando los tallos maduran la médula es esponjosa y seca y en la cosecha se cae y el tallo queda hueco o vacío; el color básico del tallo en la época de floración, puede ser verde, verde-amarillo, naranja, rosado, rojo y púrpura. En algunas variedades se puede apreciar la presencia de estrías con colores variables como el verde, amarillo, rosado y púrpura y en otras la presencia de axilas de color rosado, rojo y púrpura. Las combinaciones resultantes del color básico del tallo, el color de las estrías y el color de



axilas puede ser empleado para identificar variedades. A la madurez el color del tallo, en general, se torna de un color crema o rosado con diferentes intensidades.(FAO y CIRAD, 2013)

De acuerdo al hábito de ramificación el tallo puede ser de: Hábito sencillo, con un solo tallo y una inflorescencia terminal definida. Este tipo de tallo; se encuentra predominantemente en variedades de los ecotipos del altiplano y los de los salares; el hábito ramificado con las variantes: a) las ramas laterales tienen casi la misma longitud que el tallo principal y todas terminan en panojas, y b) el tallo principal tiene mayor longitud que los tallos secundarios dando a la planta una forma cónica con la base bastante amplia. Este tipo de ramificación es más frecuente en los ecotipos de Valle; el hábito de ramificación depende del genotipo y está altamente influenciado por la densidad de siembra, nutrientes y otros factores. La cosecha mecánica puede dificultarse en las variedades ramificadas, debido a la longitud de las diversas ramificaciones con inflorescencia y al excesivo follaje; la altura de planta, desde la base del tallo al ápice de la inflorescencia, varía de 0.5 m a más de 3 m; depende de la variedad, de la densidad de siembra, de la nutrición y del medio ambiente. Generalmente las variedades de los ecotipos de valle son más altas que las del Altiplano. (FAO & CIRAD, 2013).

Hojas

Las hojas tienen dos partes diferenciadas: el peciolo y la lámina. El peciolo de las hojas es largo y acanalado, su longitud depende de su origen; son más largos los peciolos que se originan directamente del tallo y más cortos los que se originan en las ramas. El color del peciolo puede ser verde, rosado, rojo y púrpura; y así mismo la lámina de la hoja tiene tres venas principales que se originan del peciolo. Las láminas son más grandes en el follaje y más pequeñas en la inflorescencia. Las láminas son polimórficas en la misma planta. Las láminas de la planta o el follaje pueden ser triangulares o romboidales y las



de la inflorescencia pueden ser triangulares o lanceoladas. Las hojas pueden tener márgenes enteros, dentados o aserrados. El número de dientes es variable, puede variar de 3 a 20. Las hojas y las partes tiernas de la planta están generalmente cubiertas con una pubescencia vesicular–granular blanca, rosada o púrpura. Esta pubescencia granular contiene oxalato de calcio capaz de absorber agua del medio ambiente e incrementar la humedad relativa de la atmósfera que rodea las hojas, influenciando el comportamiento de las células guarda de las estomas; por lo tanto, en la transpiración. El color de la lámina predominantemente es verde; en algunas variedades puede observarse hojas de color verde-púrpura. A la madurez las láminas se tornan amarillas, naranjas, rosadas, rojas o púrpuras.(FAO & CIRAD, 2013)

Inflorescencia

Es una panoja con una longitud variable de 15 – 70 cm. Generalmente se encuentra en el ápice de la planta y en el ápice de las ramas. Tiene un eje principal, ejes secundarios y eje terciarios. Considerando la forma y posición de los glomérulos (grupos de flores) se clasifican en amarantiformes, glomerulatas e intermedias. En el grupo amarantiforme los glomérulos están directamente insertados en el eje secundario y los glomérulos tienen una forma casi rectangular, muy semejantes a dedos. En el tipo glomerulata los glomérulos están ubicados en el eje terciario que se origina del eje secundario y toman la apariencia redondeada como las cuentas de un rosario. En el tipo intermedio los glomérulos tienen una forma no definida (entre rectangulares o redondeados). La longitud de los ejes secundarios y terciarios determina si la inflorescencia puede ser laxa, intermedia o compacta; esta última característica está asociada al tamaño de los granos, siendo los más pequeños, los formados en panojas compactas.(Alandia et al., 1979)



Flores

Las flores son sésiles o pediceladas y están agrupadas en glomérulos. La posición del glomérulo en la inflorescencia y la posición de las flores dentro del glomérulo, determinan el tamaño y el número de los granos o frutos. Es una planta ginomonoica porque presenta dos tipos de flores en la misma planta; hermafroditas y pistiladas. Las flores hermafroditas se encuentran en el ápice del glomérulo y son más grandes que las pistiladas, con un diámetro de 3 a 5 mm; tienen cinco tépalos, cinco anteras y un ovario súpero con dos o tres ramificaciones estigmáticas. Las flores pistiladas se encuentran alrededor y debajo de las flores hermafroditas, están formadas de cinco tépalos, un ovario súpero y dos o tres ramificaciones estigmáticas y tienen un diámetro de 2 a 3 mm. La proporción de flores hermafroditas y pistiladas es variable; el rango encontrado varía de 2 a 98%; esta proporción es importante si el cultivo se siembra en forma aislada, ya que influye en la cantidad de frutos formados. Además de ello, algunas variedades de quinua tienen esterilidad masculina. La quinua se considera autógena con un porcentaje de cruzamiento de 17%, aproximadamente.(FAO & CIRAD, 2013).

Fruto

Es un aquenio de forma lenticular, elipsoidal, cónico o esferoidal, cubierto por el perigonio sepaloide o las envolturas florales que rodean el fruto y se desprenden con facilidad a la madurez; sin embargo, en algunos casos puede permanecer adherido al grano incluso después de la trilla dificultando la cosecha y el procesamiento industrial de los granos. El fruto está constituido del pericarpio (capa del fruto) y la semilla. El pericarpio está adherido a la capa de las semillas y el nivel de adherencia es variable, tienen alveolos en su superficie y la saponina que le da el sabor amargo al grano. El fruto puede alcanzar un diámetro de 1.5 a 3mm. (FAO & CIRAD, 2013)

Semilla



Presenta tres partes bien definidas que son: epispermo, embrión y perisperma. El epispermo, es la capa que cubre la semilla y está adherida al pericarpio. El embrión, está formado por dos cotiledones y la radícula y constituye, aproximadamente, el 30% del volumen total de la semilla y envuelve al perispermo como un anillo, con una curvatura de 320 grados. La radícula, muestra una pigmentación de color castaño oscuro. El perispermo es el principal tejido de almacenamiento; reemplaza al endospermo y está constituido mayormente por granos de almidón, es de color blanquecino y representa prácticamente el 60% de la semilla. El color de los granos depende de la capa en observación. Si las variedades mantienen el perigonio sepaloide (tépalos de las flores) los colores son verdes, rojos y púrpura. Si se observa el pericarpio los colores pueden ser blanco, crema, amarillo, naranja, rojo, rosado, púrpura, marrón, gris y negro. Por otro lado, si el pericarpio se desprende durante el proceso de eliminación de la saponina, la capa observada es la envoltura de la semilla o epispermo y puede ser blanca, crema, roja, marrón, gris o negra. La intensidad del color puede disminuir o desaparecer en el proceso de secado de los granos en maduración en campo y la luminosidad del ambiente de almacenamiento del grano o puede ser eliminada en el agua durante el lavado de la quinua y el color del pericarpio o capa del fruto y el color del epispermo o capa de las semillas puede ser diferente en la misma semilla.(FAO & CIRAD, 2013)

2.1.3. Enfermedades de la Quinua.

En los últimos años, se ha incrementado considerablemente el área cultivada con quinua en Sudamérica, Norteamérica y Europa. Simultáneamente, las enfermedades que atacan a este cultivo van cobrando mayor importancia; sin embargo, son escasos los estudios integrales sobre identificación, distribución y caracterización de las enfermedades, plantas hospedantes, etiología, ciclo de vida y epidemiología de los



patógenos, mecanismos de resistencia y estrategias de prevención o de control hasta el momento, se han identificado tres tipos de enfermedades:(Gomez & Aguilar, 2016)

ENFERMEDADES DE LA QUINUA:

Principales o claves

- Mildiu (*Perenospora variabilis*)
- Podredumbre marrón del tallo (*Phoma exigua* var. *foveata*)

Secundarias

- Podredumbre radicular (*Rhizoctonia solani*)
- Manchas foliares (*Ascochyta hyalospora*)
- Ojo de gallo (*Cercospora* sp)
- Mancha ojival del tallo (*Phoma* spp)
- Moho verde (*Cladosporium* sp)
- Mancha bacteriana (*Pseudomonas* spp) (Gomez & Aguilar, 2016)

2.1.3.1. Mildiu o quemado

El mildiu es el patógeno más severo en la quinua y la afecta tanto en costa, en el altiplano como en valles interandinos, también se informa sus daños en otros lugares fuera de la región andina. Los mayores daños de la enfermedad se presentan en las hojas, provocando la reducción del área fotosintética de la planta, y consecuentemente afecta negativamente en el desarrollo de la planta y en el rendimiento. La enfermedad provoca el enanismo (infección sistémica) y la defoliación prematura, los cuales se traducen en la reducción del rendimiento entre el 10 y el 30%. En ataques severos y en las fases fenológicas más críticas de la planta, la enfermedad puede provocar la pérdida total en caso de variedades susceptibles. Gomez y Aguilar, (2016).

Se presenta como manchas pequeñas de forma irregular y que van creciendo a medida que la enfermedad desarrolla, su coloración puede ser clorótica o amarilla, rosada, rojiza u otro dependiendo del color de la planta y se observa un micelio de color gris en el envés de las hojas, siendo muy abundante en las variedades susceptibles. Si bien, se la encuentra más en las hojas, se pueden observar síntomas en tallos, ramas, inflorescencia y granos. Existen evidencias de la presencia de oósporos prendidos exteriormente en el epispermo de las semillas cosechadas en plantas enfermas; siendo una forma importante de dispersión de la enfermedad. Gomez y Aguilar, (2016)



Figura 2: Fotografía de planta de quinua atacada por Mildiu
Fuente: Cultivos Andinos FAO



Figura 3: Fotografía de defoliación en el cultivar
Fuente: Cultivos Andinos FAO



Generalmente, las condiciones ambientales con alta humedad favorecen el desarrollo del mildiu. Esta enfermedad se presenta en la mayoría de los lugares donde se cultiva la quinua debido a la gran diversidad genética del patógeno (Danielsen & Ames, 2000) y su amplio rango de adaptabilidad la bacteria *Bacillus subtilis* y el hongo *Trichoderma spp* junto con el abono orgánico. *Bacillus subtilis* se desarrolla eficazmente en un rango de pH 5-8, una temperatura de 15 a 50 °C con un óptimo de 28 a 35 °C y en ambientes húmedos su crecimiento es progresivo; la antibiosis es el mecanismo que emplea y por el cual ejerce su antagonismo y tiene alto espectro como biocontrolador. Dicha enfermedad se halla distribuido en todos los países donde se cultiva quinua: Sudamérica, Norteamérica y Europa.(Danielsen & Ames, 2000)

Plantas hospedantes. *P. farinosa* es un patógeno altamente especializado. Bajo condiciones naturales o pruebas de inoculación artificial de *P. farinosa* aislada de quinua, se ha encontrado infección sobre kañiwa (*Chenopodium pallidicaule*), espinaca (*Spinaca oleracea*), remolacha (*Beta vulgaris*) u otros chenopodiaceas Síntomas. La sintomatología varía en las diferentes variedades, fases fenológicas de desarrollo y órgano infectado de la planta. Generalmente, la enfermedad se inicia en las hojas inferiores, propagándose hacia las hojas superiores. En la cara superior, se observan manchas amarillas pálidas (cloróticas) o rojizas de tamaño y forma variable. En la cara inferior, se ve una pelusilla de color plomo o gris violáceo (esporángio y esporangióforos). Los síntomas van aumentando sucesivamente en tamaño y número.(Danielsen & Ames, 2000)



Figura 4: Fotografía en el haz y envés de la hoja (*Peronospora farinosa*)
Fuente: (S. Danielsen).

En algunos casos, las lesiones están bien localizadas y definidas; sin embargo, en otros, las lesiones son muy tenues y amplias. En ambos casos, pueden cubrir la totalidad del área foliar. Ocasionalmente ocasionan alteraciones fisiológicas disminuyendo severamente la fotosíntesis. En infecciones graves, llega a necrosar toda la hoja o área afectada de la planta y produce defoliación generalizada. (Danielsen & Ames, 2000)

Cuando las variedades son susceptibles y el ataque es severo, se observa una distorsión de los tejidos afectados y las hojas muestran depresiones pronunciadas, semejándose a ampollas pálidas o coloreadas. En otros casos, las infecciones del patógeno adoptan una característica de tipo sistémico, capaz de llevar a una confusión por ataque de virus, las plantas se quedan amarillentas y con enanismo pronunciado. Esta sintomatología se expresa cuando la infección del patógeno se inicia por medio de oosporas en el momento de la germinación de la semilla. Alandia, Otazu, y Salas, (1979)

Los ataques intensos al follaje se reflejan en las panojas, las que se caracterizan por una coloración oscura. Esto ocasiona que los granos de quinua se conviertan en portadores de esporas, que producirán plantas enfermas en la siguiente campaña agrícola. En los tallos y ramas secundarias, las manchas son menos pronunciadas que en las hojas. Evaluación de la enfermedad. Evaluar una enfermedad en campo es complicado y depende de la

persona, metodología, época e instrumentos (escalas) de evaluación.(Danielsen & Ames, 2000), aconsejan determinar el porcentaje del área foliar afectado en hojas individuales y no en plantas enteras.

Se mide la severidad en 3 hojas por planta, una de cada tercio escogidas al azar, según una escala de 0% hasta 100% (Fig. 4); luego, se calcula el promedio de las 3 hojas para obtener el valor de la planta. El valor mínimo que indica presencia de enfermedad es 1%. Con un mínimo de 3 evaluaciones, los valores de severidad se pueden usar para calcular el AUDPC (área bajo la curva de progreso de la enfermedad).(Danielsen & Ames, 2000)

Parámetro usado para comparar resistencia/susceptibilidad y comportamiento de diferentes cultivares, bajo diferentes ambientes climáticos.

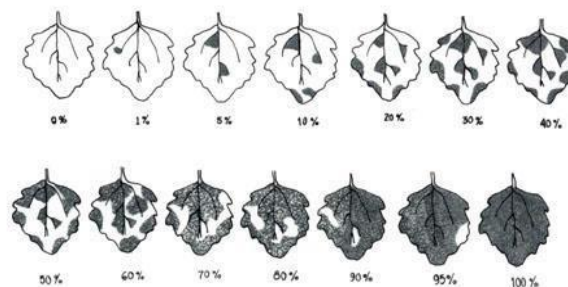


Figura 5: Escala de evaluación para mildiu

Fuente: Danielsen y Ames, 2000

Ciclo de la enfermedad. El ciclo de vida de *P. farinosa* y su relación con el desarrollo del mildiu en la quinua es registrada para condiciones del altiplano peruano-boliviano (Alandia et al., 1979)

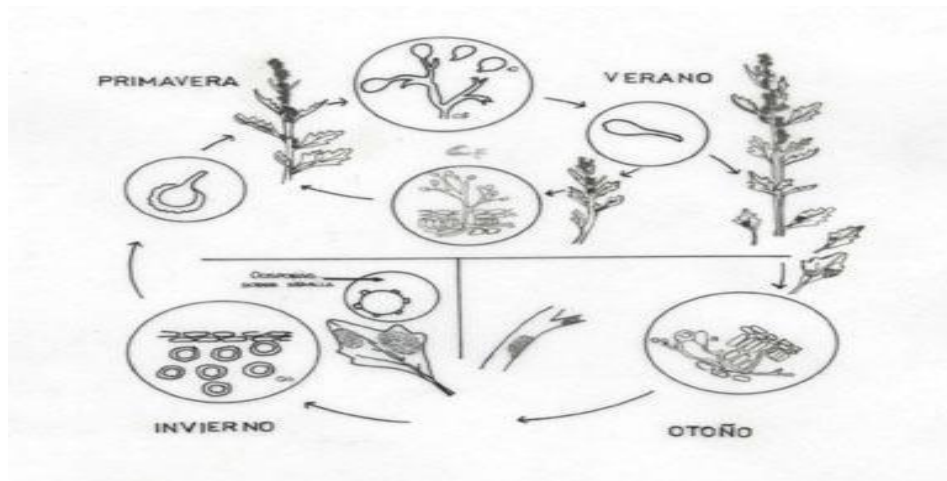


Figura 6: Ciclo de vida de *Peronospora farinosa*

Fuente: Alandia et al. (1979)

Generalmente, los signos iniciales del mildiu se evidencian a fines de la primavera, en plantas que crecen en ambientes húmedos. Las hojas basales presentan manchas cloróticas o amarilla-pálida, las cuales aumentan en tamaño y número conforme se incrementa la humedad del medio ambiente; por cuya razón, al aumentar la humedad atmosférica, la enfermedad se desarrolla rápidamente en el haz y en vez de las hojas. (Alandia et al., 1979)

En el envés, el síntoma inicial se convierte en manchas cloróticas sobre las cuales se ubican las estructuras vegetativas y de fructificación del patógeno (esporangios y esporangioforos); a partir de éstas, la enfermedad es diseminada en todo el campo por el viento y la lluvia. El micelio del hongo se desarrolla en espacios intercelulares del hospedero (hojas, tallos y otros) y emite haustorios que penetran en las células. Los esporangioforos emergen por las estomas de las hojas y llevan apicalmente los esporangios que constituyen los órganos de diseminación del patógeno. Los oogonios son de forma esférica y los anteridios claviformes aparecen en las hojas al finalizar el



verano. Las oosporas se producen durante el otoño y son abundantes en tejidos foliares. (Alandia et al., 1979)

Se han encontrado oosporas adheridas en la parte externa del episperma del grano. Se desconoce el momento en que tiene lugar la germinación de las oosporas y la penetración en los tejidos de plántulas (Alandia et al., 1979). Así, también, son estructuras de sobrevivencia y sirven como fuente primaria de inóculo en la siguiente campaña agrícola. Epidemiología. El inóculo del mildiu se disemina a través del viento, lluvias (esporangios), semilla y suelo (oosporas). La infección es estimulada por alta humedad relativa (>80%) y las temperaturas moderadas (13– 18°C). La germinación de esporangios será abundante cuando las condiciones ambientales son favorables, sea cual fuera la fuente de inóculo o diseminación. En la época del cultivo, se pueden producir varias generaciones de patógenos que se reproducen asexualmente (esporangios), ocasionando infecciones sucesivas (policíclicos). Identificaron 3 factores de resistencia y 4 grupos de virulencia. La interacción específica entre aislamiento y hospedante indica la presencia de genes mayores. Las accesiones ECU-291, ECU-470, ECU-379 y ECU-288 fueron propuestas como un set preliminar de diferenciales para identificar grupos de virulencia (razas) de mildiu en quinua. Sin embargo, pruebas de virulencia en las que se empleó un set de diferenciales que representa la mayoría de la variación dentro de *C. quinua* mostraron que la variabilidad dentro de *P. farinosa* es mucho más complejo que lo que se pudo revelar con los diferenciales ecuatorianos (Danielsen et al., 2000b).

El cultivo de quinua presenta un amplio rango de susceptibilidad al Mildiu (Bonifacio & Saravia, 1999); Hay variedades que sufren infecciones escalonadas o sistémicas, algunas sufren lesiones que abarcan todo el limbo folia y otras evidencian un alto grado de tolerancia y/o resistencia.



Danielsen et al. (2000a) reportan que Utusaya, cultivar precoz, es altamente susceptible al ataque de Mildiu, que bajo condiciones de valle provocó una defoliación completa con una pérdida de 99% en rendimiento. Los cultivares más susceptibles son restringidos para áreas secas donde, debido a la baja humedad, el Mildiu no se presenta. En cambio, los cultivares tardíos, La Molina 89, Amarilla de Maranganí e Ingapirca, son resistentes.

Mancha foliar

(Ascochyta Hyalospora). Los síntomas de esta enfermedad son manchas necróticas en las hojas, más o menos circulares, con el centro de color crema y los bordes algo marrones. Éstas, son de un tamaño aproximado de 5 a 10 mm de diámetro. Sus fuertes ataques pueden provocar defolaciones. Podredumbre marrón del tallo (Phoma Exigua Var. Foveata). Esta enfermedad causa daños considerables en la quinua cultivada e incluso la muerte de ésta. El hongo que transmite la enfermedad penetra en las plantas a través de sus heridas mecánicas; su propagación es favorecida por las granizadas. Los síntomas de su ataque se observan en los tallos y en las panojas, con lesiones marrones. Al interior de estas lesiones, se pueden observar puntitos negros; éstos son los hongos, que causan la enfermedad. El tamaño de las lesiones es de unos 5 a 15 cm. El tallo presenta un aspecto chupado, con la parte superior amarillenta. Las partes cercanas al ápice no cuentan con hojas, pudiendo doblarse y quebrarse en los puntos afectados.(Gomez & Aguilar, 2016)

Ojo de Gallo

Se presenta en las hojas como pequeñas lesiones de color marrón claro, aumentando de tamaño a medida que crece la infección. La enfermedad se hace mucho más evidente en periodos secos o de prolongada sequía. Sin embargo, su aparición en forma severa es después del ataque del mildiu o cuando la planta está próxima a la



madurez. Su control puede realizarse con rotaciones adecuadas durante dos a tres años con cualquier otro cultivo que no pertenezca a la sub familia de las Chenopodioideae. Se recomienda la aplicación de Benomyl en rotación con Clorotalonil, Difenoconazol o Tiabendazol. En plantas jóvenes o cuando el tercio inferior está afectado se recomienda aplicar Metalaxil.(Gomez & Aguilar, 2016)

2.1.3. REDES MESH.

Una red Mesh es una red de enrutadores entrelazados llamados nodos o puntos. Estos nodos trabajan entre sí para proporcionar cobertura de Internet en un área amplia, algo que una red doméstica tradicional no puede proporcionar. En una configuración regular de la red doméstica, hay un único router que proporciona cobertura de la Internet por lo que se puede alcanzar, a través de paredes, por la puerta, subir las escaleras, etc. Una Mesh de red consiste en múltiples routers que cada uno proporcionan cobertura o red wifi, pero es sigue siendo una sola red, ya que cada una también se está comunicando con los otros nodos cercanos. El resultado es una red única que tiene una cobertura mucho mejor que la que puede proporcionar un solo enrutador. Una red Mesh puede proporcionar acceso a Internet en cualquier lugar al que puedan llegar los nodos, como a través de una casa de tres pisos o incluso a través de varias cuadras de la ciudad.(Fisher, 2020)

En una red Mesh inalámbrica, la red se extiende entre docenas de cosas que se comunican entre sí. Cuantas más cosas haya en la red Mesh, más fuerte será la red. Solo una cosa, llamada controlador, debe conectarse a Internet a través de un módem /enrutador o Gateway. Las redes Mesh están creciendo en popularidad en los hogares inteligentes, especialmente Zigbee y Z-Wave.(SAS THE Power to Know, 2017)

Para empezar, estamos hablando de internet, las redes y las conexiones entre dispositivos. Por lo tanto, es importante presentar el modelo OSI, que es un modelo de referencia de cómo es el sistema se comunican a través de una red. Este proceso de comunicación se puede dividir en siete grupos distintos de funciones relacionadas. En una red, todos los dispositivos, también llamados nodos, usan esas siete capas para comunicarse entre sí. Cada capa en el modelo OSI sirve a la capa superior, que a su vez sirve a la capa superior. Entonces, por ejemplo, cuando se intercambia información entre nodos, el proceso funcionará así: un flujo de datos circula a través de las capas de la máquina que envió el mensaje, luego viaja a través de la red y finalmente fluye a través de las capas del destinatario dispositivo. ¿Por qué es esto relevante para este artículo? En el contexto de las redes Mesh, Este modelo está en uso dentro de los sistemas activos en la red. En consecuencia, las capas más importantes que se discutirán son las capas de medios, capa 3 y siguientes, que se describen más adelante.(Schutter, 2018)

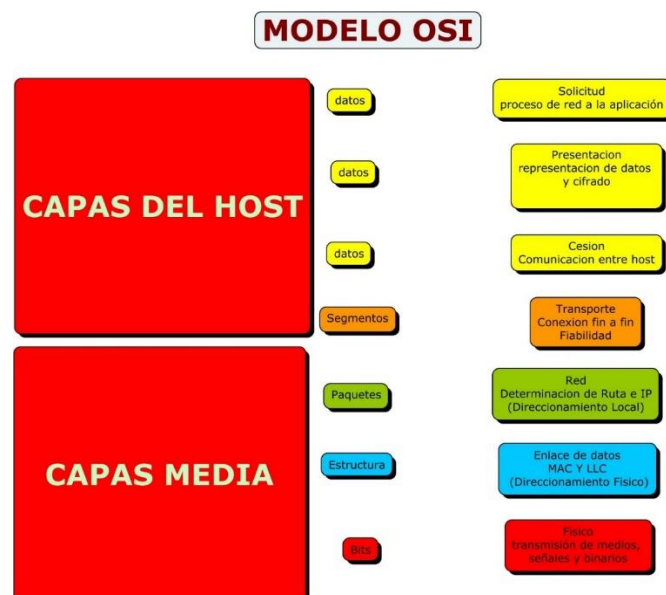


Figura 7: Modelo OSI - 7 capas

Fuente: Hackernoon (2018)

Queremos entender qué es una topología Mesh y explicar, procederemos a compararla con la topología más utilizada en la actualidad. Esa topología es la estrella, en la que los dispositivos están conectados a un punto de acceso central (red centralizada). Por el contrario, en una red Mesh, los nodos se conectan directamente entre sí (red descentralizada). Los dos diagramas a continuación ilustran estos dos ejemplos.

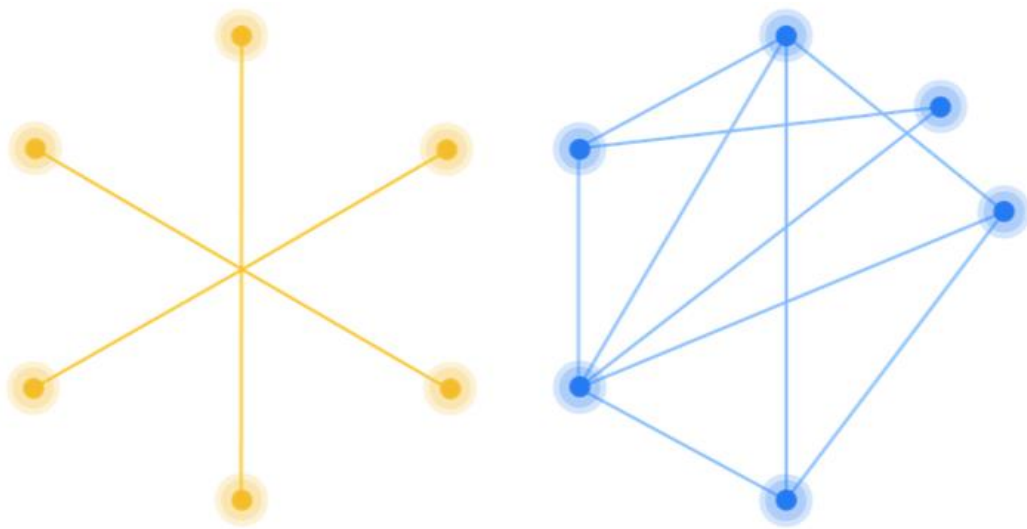


Figura 8: Topología de Estrella y Mesh

Fuente: Hackernoon (2018)

Entonces, ¿qué implica esto? En la topología en estrella, el punto central tiene que manejar todo el tráfico en la red. También tiene que reenviar la información a los destinos en nombre de las fuentes. Por el contrario, en la topología de la red Mesh, los nodos permiten la comunicación punto a punto o igual a igual (P2P). Esto elimina la necesidad de una entidad central. Como puede imaginar, estos diferentes diseños ofrecen ventajas y desventajas en el uso de la red. Estos se discutirán en detalle más adelante.(Schutter, 2018).

Las redes Mesh se pueden implementar en varios tipos de radios (Radio Access Technology RAT) simultáneamente (por ejemplo: Bluetooth, WIFI, operador de telefonía



móvil, etc.). En este caso se dice que son multi-transporte. Estos dispositivos van desde tener acceso exclusivo a Bluetooth, Wi-Fi y el teléfono se transportan como, por ejemplo, los teléfonos inteligentes actuales, hasta nodos que admiten todas las plataformas de conectividad posibles. (Schutter, 2018)

2.1.3.1. PROTOCOLOS DE ENRUTAMIENTO

La forma en que fluye el tráfico o cómo se comunican los nodos se denomina protocolo de enrutamiento. Se refiere a la capa 3 del modelo OSI. La taxonomía del protocolo de enrutamiento de red Mesh se presenta como tres tipos principales de protocolos: proactivo, reactivo o híbrido. Estos tipos de protocolos difieren en su mayor parte en el paradigma que utilizan para realizar el descubrimiento de la red. Esto finalmente tiene consecuencias en el rendimiento y la escala.

2.1.3.2. VENTAJAS DE USAR REDES MESH

- No más problemas de punto único de falla, que es el problema en las topologías en estrella (y aún peor en las topologías de bus). Si un nodo ya no puede funcionar, la red tiene la capacidad de redirigir, lo que le permite comunicarse entre los nodos restantes.
- La desconexión de la red es imposible a menos que haya algún tipo de catástrofe mundial que anule todos los dispositivos electrónicos del mundo.
- La red funciona con una infraestructura mínima y, por lo tanto, se puede implementar más rápido a un costo menor que la infraestructura tradicional.
- Dado que los dispositivos en una red de malla pueden retransmitir aún más las señales, tienen la capacidad de conectar miles de sensores en un área amplia (por ejemplo, ciudades). Otras instancias incluyen operar en áreas con grandes multitudes (ej .:



conciertos, festivales, etc.) o conectar dispositivos en áreas remotas (ej .: en montañas o en transporte público bajo tierra), y muchos, muchos más.

- No hay autoridad centralizada en una red de malla. Por esa razón, algunas personas lo comparan con lo que era Internet en el pasado: comunicación localizada, anónima, basada en los ciudadanos y segura.

2.1.3.3. DESVENTAJAS QUE EXISTÍAN EN REDES MESH

- Las fuerzas reguladoras y del mercado dificultan la implementación de redes en malla.
- Las redes de malla pueden reemplazar a los proveedores de Wi-Fi, operadores telefónicos y otros intermediarios que brindan conectividad a las personas. Como consecuencia, el intermediario pierde, por lo que para ellos no existe un incentivo financiero para desarrollar esta tecnología.
- Las redes de malla son difíciles de administrar y solucionar problemas. Para redes grandes, se necesita una tecnología de malla fuerte para que valga la pena. Esto puede ser difícil de encontrar.
- La duración de la batería afecta la disponibilidad de los nodos. Por ejemplo, un teléfono sin batería podría interrumpir la red, causando más sobrecarga de enrutamiento y menos confiabilidad.
- El costo de implementación a veces puede ser problemático en ciertos escenarios. (Schutter, 2018)

2.1.4. RED DE MALLA INALÁMBRICA (WMN)

Una red de malla inalámbrica es una infraestructura de nodos (una topología de malla) que están conectados entre sí de forma inalámbrica. Estos nodos se complementan entre sí para extender una señal de radio (como una conexión Wi-Fi o celular) para

enrutar, retransmitir y proxy de tráfico hacia / desde los clientes. Cada nodo difunde la señal de radio un poco más allá que el anterior, minimizando la posibilidad de zonas muertas.(Lee, 2018).

Componentes:

- **Puerta de enlace:** los enrutadores de borde son los dispositivos que tienen conectividades adicionales más allá de la malla que les permiten pasar mensajes entre redes. Puede pensar en estos dispositivos como una "red de retorno" a Internet para la red de malla local.
- **Repetidor:** los enrutadores son dispositivos que reenvían mensajes entre dispositivos finales (puntos finales) en una red de malla. Por lo general, no están diseñados para dormir porque son parte de la infraestructura de las redes de malla.
- **Punto final:** los dispositivos finales son dispositivos solo de malla que no enrutan mensajes para otros dispositivos en la red de malla. Debido a que no tienen responsabilidades de red, pueden ingresar al modo de suspensión y son buenos candidatos para nodos y sensores alimentados por batería.(Lee, 2018)

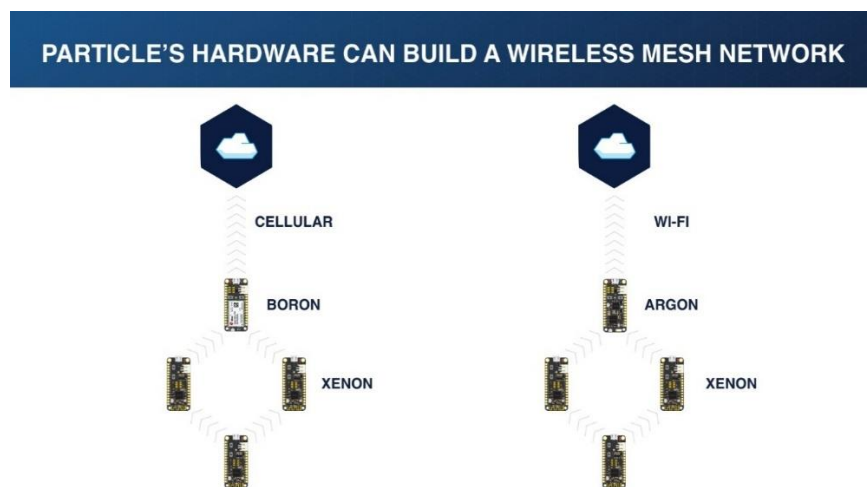


Figura 9: Hardware de Particle.

Fuente: Particle (2019)



2.1.5. INTERNET DE LAS COSAS (IDC)

Internet de las Cosas es “la interconexión a través de Internet de dispositivos informáticos integrados en objetos cotidianos, lo que les permite enviar y recibir datos”. En otras palabras, IdC conecta tus dispositivos a Internet o a otros aparatos, para que puedan realizar nuevas funciones, como por ejemplo controlar elementos inteligentes de forma remota y recibir alertas y actualizaciones de estado. Se refiere a los miles de millones de dispositivos físicos en todo el mundo que ahora están conectados, recolectando y compartiendo datos. Gracias a las redes inalámbricas y el bajo costo de los nuevos procesadores, es posible que casi cualquier cosa, desde una aspiradora inteligente hasta un vehículo autónomo, forme parte de la IdC. Esto agrega un nivel de inteligencia digital a los dispositivos que les permite comunicar datos en tiempo real sin la participación de un ser humano, fusionando de alguna manera el mundo digital con el físico. (Peña, 2019).

2.1.5.1. COSAS.

Cuando hablamos de Internet de las cosas, ¿qué es exactamente un “¿Cosa?” La respuesta depende de a quién le preguntes. No es una simple respuesta correcta porque depende del contexto, y el mismo depende del que lo va usar. En términos generales, una cosa podría ser un sensor, un dispositivo (móvil o no) y / o cualquier cosa con una IP Address. Lo cierto es que la población de las cosas está explotando Pronto habrá decenas de miles de millones de cosas en el planeta. Las cosas dependen de la conectividad para ser "inteligentes".(SAS THE Power to Know, 2017)

2.1.5.2. DISPOSITIVOS DEL INTERNET DE LAS COSAS

Casi cualquier objeto físico puede transformarse en un dispositivo IdC si puede conectarse a Internet y controlarse de esa manera. Una bombilla que se puede encender usando una aplicación de teléfono inteligente es un dispositivo IdC. Al igual que un sensor

de movimiento o un termostato inteligente en tu oficina o una farola conectada. (RS Components, 2016)



Figura 10: Dispositivos IdC
Fuente: Digital Trends

Un dispositivo IdC podría ser tan sencillo como el juguete de un niño o tan serio como un camión sin conductor. O tan complicado como un motor a reacción que ahora está lleno de miles de sensores que recopilan y transmiten datos para asegurarse de que esté funcionando de manera eficiente. A una escala aún mayor, los proyectos de ciudades inteligentes están llenando regiones enteras con sensores. Todo con el fin último de ayudar a comprender y controlar el medio ambiente. El término IdC se usa principalmente para dispositivos que generalmente no se espera que tengan una conexión a Internet y que pueden comunicarse con la red independientemente de la acción humana.(Antevenio, 2020).

2.1.5.3. PROTOCOLOS PARA IDC.

Ahora que hemos visto la necesidad y planteamiento de los protocolos destinados a aplicaciones de IdC, vamos a ver algunos de los muchos protocolos M2M disponibles.



- **MQTT (MQ Telemetry Transport)** es un protocolo PubSub de Message Service que actúa sobre TCP. Destaca por ser ligero, sencillo de implementar. Resulta apropiado para dispositivos de baja potencia como los que frecuentemente tenemos en IdC. Está optimizado para el routing activo de un gran número de clientes conectados de forma simultánea.
- **AMQP (Advanced Message Queuing Protocol)** es un protocolo PubSub de Message Queue. AMQP está diseñado para asegurar la confiabilidad e interoperabilidad. Está pensado para aplicaciones corporativas, con mayor rendimiento y redes de baja latencia. No resulta tan adecuado para aplicaciones de IdC con dispositivos de bajos recursos.
- **WAMP (Web Application Messaging Protocol)** es un protocolo abierto que se ejecuta sobre WebSockets, y provee tanto aplicaciones de PubSub como rRPC.
- **CoAP (Constrained Application Protocol)** es un protocolo pensado para emplearse en dispositivos de IdC de baja capacidad. Emplea el modelo REST de HTTP con cabeceras reducidas, añadiendo soporte UDP, multicast, y mecanismos de seguridad adicionales.
- **STOMP (Streaming Text Oriented Messaging Protocol)**, es un protocolo sencillo que emplea HTTP y mensajes de texto para buscar el máximo de interoperabilidad.
- **XMPP (Extensible Messaging and Presence Protocol)** es un protocolo abierto basado en XML diseñado para aplicaciones de mensajería instantánea.
- **WMQ (WebSphere MQ)** es un protocolo de Message Queue desarrollado por IBM.

Hasta aquí esta entrada sobre protocolos de comunicación M2M en IdC. Hemos visto las necesidades especiales de estos protocolos y cómo resolverlo con una externalización en un servicio de mensajería.(Llamas, 2019)

2.1.5.4. ARQUITECTURAS IDC

La arquitectura describe la estructura de su solución de IdC, lo que incluye los aspectos físicos (esto es, las cosas) y los aspectos virtuales (como los servicios y los protocolos de comunicación). Adoptar una arquitectura con múltiples niveles le permite concentrarse en mejorar su comprensión acerca de cómo todos los aspectos más importantes de la arquitectura funcionan antes de que los integre dentro de su aplicación de IdC. Este enfoque modular le ayuda a gestionar la complejidad de las soluciones IdC. (Guerrero et al., 2017)

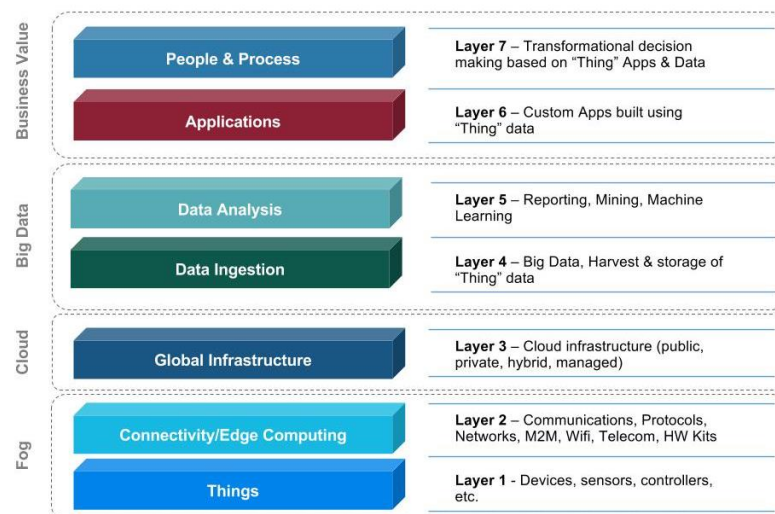


Figura 11: Siete(7) capas de Internet de las Cosas

Fuente: (Guerrero et al., 2017)



2.1.5.5. METODOLOGÍAS EN IDC.

Vamos a repasar un par de términos en cuanto a metodologías que podemos encontrar en IdC. En realidad, los conceptos son parecidos, pero conviene entender las diferencias (aunque sólo sea para hablar con propiedad).(Llamas, 2019)

PUBLISH / SUSCRIBE (PUBSUB)

La metodología Pub/Sub es un patrón de mensajería donde un agente, el 'Subscriber', informa al Router que quiere recibir un tipo de mensajes. Otro agente, el 'Publisher' puede publicar mensajes. El Router distribuye los mensajes a los Subscribers.(Llamas, 2019)

ROUTER REMODER PROCEDURE CALLS (RRPC)

El rRPC es un patrón de ejecución remota de procedimientos donde un agente, llamado 'Callee', comunica al Router que proporciona un cierto procedimiento. Otro agente, llamado 'Caller', puede llamar a este procedimiento. El Router invoca el procedimiento en el Callee, recoge el resultado del proceso, y lo comunica al Caller que lo ha invocado.(Llamas, 2019)

2.1.5.6. SEGURIDAD DE IDC.

La seguridad de IdC es una cuestión de proteger los dispositivos conectados y sus redes adyacentes de posibles amenazas de seguridad (ya sea de competidores, estados nacionales o personas maliciosas). Las prácticas y las tecnologías utilizadas para proteger los dispositivos conectados evolucionan constantemente porque se descubren nuevos hacks y vulnerabilidades de seguridad todo el tiempo. Eso significa que una sólida



solución de seguridad de IdC debe incluir características y prácticas que mantengan seguros los sistemas conectados hoy y mañana. (The Particle Team, 2019)

Para proteger los dispositivos, los clientes y las empresas, los responsables de la toma de decisiones deben estar atentos a los riesgos únicos de un sistema IdC. Estos riesgos incluyen:

- Exposición de datos del cliente
- Exposición de datos del cliente
- Daño Físico
- Tiempo de inactividad de alto riesgo
- Responsabilidad más amplia
- Reputación y daño a la Marca

2.1.6. AGRICULTURA DE PRECISIÓN

La Agricultura de Precisión (aquella basada en tecnología), consiste en gestionar las variaciones en el campo de manera precisa para aumentar la producción, utilizando menos recursos y reduciendo los costos de producción. Uno de los aspectos claves de este tipo de agricultura es que procura que los cultivos tengan en cuenta la variabilidad especial de los suelos y el estado de las plantas. Asimismo, permite la recolección, almacenamiento, análisis y procesamiento de información georreferenciada de un campo o cultivo definido.(El Peruano, 2017).

Tradicionalmente, la agricultura se practica realizando una tarea particular, tal como siembra o cosecha, en función de un horario predeterminado. Sin embargo, al recolectar datos en tiempo real sobre el clima, la calidad del suelo y del aire, la madurez de los cultivos e incluso los costos, la disponibilidad de los equipos y la mano de obra,



puede hacerse un análisis predictivo para tomar decisiones más inteligentes, lo cual es conocido como agricultura de precisión (Ibañez, 2009). La implementación de técnicas de agricultura de precisión se realiza con el objetivo de poder determinar con exactitud las necesidades de riego y de fertilizantes, las fases de desarrollo y de maduración de los productos, los puntos óptimos de siembra y de recolección, etc. Es decir, predecir adecuadamente las distintas etapas de producción de los cultivos. Para ello, es importante obtener toda la información posible del agua, del suelo, de las plantas y del ambiente. (Quiroga et al., 2017)

2.1.6.1. IDC EN LA AGRICULTURA DE PRECISIÓN.

El Internet de las Cosas (IoC) es una red de objetos tales como sensores y actuadores conectados a Internet a través de redes fijas e inalámbricas donde pueden capturar datos de forma autónoma y autoconfigurarse de forma inteligente basada en hechos del mundo físico; permitiendo que estos sistemas se conviertan en participantes activos en diversos procesos públicos, comerciales, científicos y personales (Fundación de la Innovación Bankinter., 2011)

El IoC es impulsado por una combinación de métodos y procesos que comprenden la utilización de sensores, una red y un dispositivo final, los cuales tienen como objetivo principal proporcionar al usuario final una representación de los datos de una manera visual, legible y práctica. Cuando se habla de IoC enfocado en la agricultura no sólo se hace referencia a dispositivos móviles como tabletas o drones, pues este concepto implica una interconexión de sistemas agrícolas de comunicaciones compatibles y colaborativos que comparten información (sensores, cámaras, lámparas, estaciones meteorológicas, etc.) (Fundación de la Innovación Bankinter, 2011).



Beneficios

Los beneficios que los agricultores obtienen cuando usan la IdC son de dos tipos. En primer lugar, estos sistemas ayudan a los agricultores a disminuir los costos de producción y los desechos al optimizar el uso de insumos. Además, los datos más precisos que se pueden recopilar ayudan a tomar mejores decisiones y, por ende, se pueden obtener mayores rendimientos, por lo que es uno de los objetivos del proyecto.

Desafíos

Sin embargo, persisten los desafíos para la IdC en la agricultura en las regiones menos desarrolladas. Primero, en las zonas alejadas suele no haber redes de comunicación. Asimismo, se deben ofrecer incentivos apropiados a los agricultores para que adopten los sistemas de IdC, cuya instalación sigue siendo bastante cara en un principio. La buena noticia es que hay organizaciones e iniciativas que ya han comenzado a abordar estos desafíos. Por ejemplo, MimosasTEK ayuda a los pequeños agricultores de Viet Nam a adoptar la agricultura de precisión que es posible gracias a la IdC. Las cooperativas de agricultores pueden arrendar dispositivos de hardware, lo que reduce los costos para los pequeños agricultores. Otro ejemplo es Eruvaka, (i) una nueva empresa india que ofrece soluciones de gestión de estanques acuícolas basadas en la IdC para ayudar a los agricultores como Nguyen a disminuir los riesgos y aumentar la productividad.(Hyea-Won & Vikas-Choudhary, 2017), para este tipo de desafíos es que se implementa este proyecto, y con la ayuda de la Plataforma particle y los productos que ofrece, la implementación tendrá sostenibilidad.

2.1.6.2. ARQUITECTURA LAMBDA.

La arquitectura Lambda intenta equilibrar en tiempo de respuesta, el rendimiento y la tolerancia a fallos mediante el procesamiento por lotes para proporcionar vistas completas y precisas de los datos por lotes, mientras que simultáneamente se utiliza el procesamiento de flujo en tiempo real para proporcionar vistas de datos en línea. (Quiroga et al., 2017)

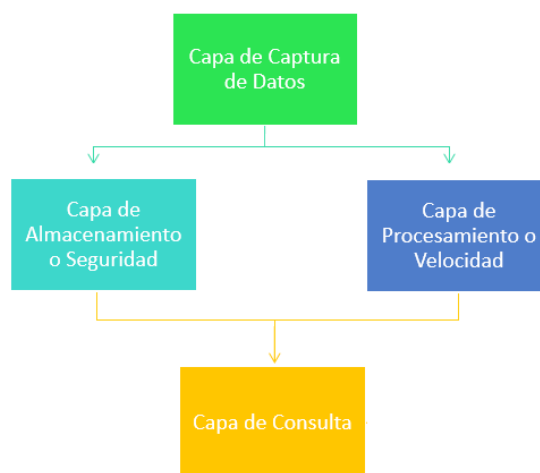


Figura 12: Diagrama de una Arquitectura Lambda

Fuente: RISTI (2017)

2.1.7. HARDWARE Y SENSORES.

En este capítulo se presenta el hardware y sensores utilizados para llegar al objetivo de la investigación la cual es diseñar e implementar un sistema de monitoreo utilizando internet de las cosas y redes Mesh, para la mejora de la calidad en la producción de sembríos de quinua en la ciudad de Juliaca.

2.1.7.1. ARGÓN.

El Argón es un potente kit de desarrollo habilitado para Wi-Fi que puede actuar como un punto final de Wi-Fi independiente o como una puerta de enlace habilitada para

Wi-Fi para redes Particle Mesh. Se basa en el Nordic nRF52840 y tiene un circuito de carga de batería incorporado, por lo que es fácil conectar un Li-Po e implementar su red local en minutos.

El Argon es ideal para conectar proyectos existentes a Particle Device Cloud o como una puerta de enlace para conectar un grupo completo de puntos finales locales. (Particle, 2019).

En el proyecto, el dispositivo Argon se comprotara como dispositivo líder encargado de recopilar los datos que los sensores detectaran en los senbrios de quinua.



Figura 13: Dispositivo Argón
Fuente: Particle, (2019)

Características del dispositivo Argón

Junta de desarrollo de argón

Mini placa de pruebas

Antena flexible de 2.4GHz

cable USB



El proyecto inicial incluye dos resistencias (220 ohmios), un LED rojo y un fotodiodo.

Especificaciones:

- Espressif ESP32-D0WD Coprocesador Wi-Fi 2.4G -Flash de
 - 4MB en placa para ESP32
 - -802.11 b / g / n soporte
 - -802.11 n (2.4 GHz), hasta 150 Mbps
- Nordic Semiconductor nRF52840 SoC
 - -ARM Cortex-M4F Procesador de 32 bits @ 64MHz
 - -1MB flash, 256KB RAM
 - -IEEE 802.15.4-2006: 250 Kbps
 - -Bluetooth 5: 2 Mbps, 1 Mbps, 500 Kbps, 125 Kbps
 - -Soporta instrucciones DSP, Cálculos de unidad de punto flotante (FPU)
 - acelerado HW -ARM TrustZone CryptoCell-310 Módulo criptográfico y de seguridad
 - -Hasta +8 dBm de potencia TX (hasta -20 dBm en pasos de 4 dB)
 - -NFC-A etiqueta
- Flash SPI adicional de 2 MB a bordo
- 20 señales mixtas GPIO (6 x analógico, 8 x PWM), UART, I2C, SPI
- Micro USB 2.0 a toda velocidad (12 Mbps)
- Conector integrado de carga y batería Li-Po
- Conector JTAG (SWD)
- LED de estado RGB
- Botones de reinicio y modo
- Antena PCB de 2.4GHz integrada para Thread / BLE (no es compatible con Wi-Fi)

- Dos conectores UFL para antenas externas (uno para Thread / BLE, otro para Wi-Fi)
- Cumple con la especificación de pluma en dimensiones y pinout
- Certificación FCC, CE e IC
- Cumple con RoHS (sin plomo). (Particle., 2019)

2.1.7.2. XENÓN.

El Xenon es un kit de desarrollo de malla de bajo costo que puede actuar como punto final o repetidor dentro de una red Particle Mesh.

Equipado con el Nordic nRF52840, el Xenón tiene un circuito de carga de batería incorporado que facilita la conexión de una batería Li-Po y 20 GPIO de señal mixta para interactuar con sensores, actuadores y otros dispositivos electrónicos.

En el proyecto el dispositivo Xenón es el mejor para conectar sensores, motores, bombas, válvulas que necesitamos para lograr el objetivo de Monitorizar los cultivos de quinua. Y se combinará con una puerta de enlace de argón, para obtener todos esos excelentes datos en la nube de dispositivos, .(Particle., 2019).



Figura 14: Dispositivo Xenon

Fuente: Particle. (2019)



Especificaciones:

Nordic Semiconductor nRF52840 SoC

- ARM Cortex-M4F Procesador de 32 bits @ 64MHz
- 1MB flash, 256KB RAM
- IEEE 802.15.4-2006: 250 Kbps
- Bluetooth 5: 2 Mbps, 1 Mbps, 500 Kbps, 125 Kbps
- Soporta instrucciones DSP , Cálculos de unidad de punto flotante (FPU) acelerado HW -ARM TrustZone CryptoCell-310 Módulo criptográfico y de seguridad
- Hasta +8 dBm de potencia TX (hasta -20 dBm en pasos de 4 dB)
- NFC-A etiqueta
- Flash SPI adicional de 2 MB a bordo
- 20 señales mixtas GPIO (6 x analógico, 8 x PWM), UART, I2C, SPI
- Micro USB 2.0 a toda velocidad (12 Mbps)
- Conector integrado de carga y batería Li-Po
- Conector JTAG (SWD)
- LED de estado RGB
- Botones de reinicio y modo
- Antena PCB a bordo
- Conector UFL para antena externa
- Cumple con la especificación de pluma en dimensiones y pinout
- Certificación FCC, CE e IC
- Cumple con RoHS (sin plomo) (Particle., 2019)

2.1.7.3. SENSOR DE TEMPERATURA Y HUMEDAD: DHT22

El DHT22 Sensor digital de temperatura y humedad. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no hay pines de entrada analógica) . Es bastante simple de usar, pero requiere sincronización cuidadosa para tomar datos. El único inconveniente de este sensor es que sólo se puede obtener nuevos datos una vez cada 2 segundos, así que las lecturas que se pueden realizar serán mínimas cada 2 segundos.

Sólo tienes que conectar el primer pin de la izquierda a la fuente de alimentación 3 -5V, el segundo pin a tu pin de entrada de datos y el cuarto (último) pin a tierra. ¡A pesar de que utiliza un solo cable para enviar datos no es compatible con el protocolo de Dallas One-Wire! Si desea conectar varios sensores, cada uno debe tener su propio pin de datos Hay una librería de ejemplo disponible para Arduino.

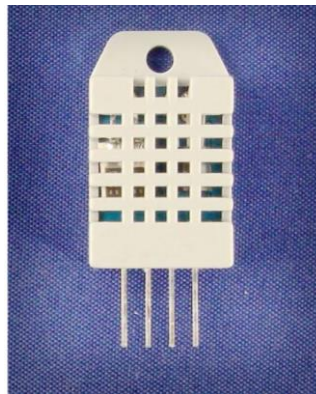


Figura 15: Sensor DHT22

Fuente: Aosong Electronics (2017)

Características

- Alimentación: $3.3Vdc \leq Vcc \leq 6Vdc$
- Rango de medición de temperatura: $-40^{\circ}C$ a $80^{\circ}C$



- Precisión de medición de temperatura: $<\pm 0.5$ °C
- Resolución Temperatura: 0.1°C
- Rango de medición de humedad: De 0 a 100% RH
- Precisión de medición de humedad: 2% RH
- Resolución Humedad: 0.1%RH
- Tiempo de censado: 2s
- Sensor de Temperatura y humedad (ambiente)

2.1.7.4. SENSOR DE MOVIMIENTO PIR.

El sensor PIR detecta el movimiento midiendo los cambios en los niveles de infrarrojos (calor) que emiten los objetos que lo rodean. Cuando se detecta el movimiento del sensor PIR, emite una señal de alto en su pin de salida. Rango de detección de 7 metros y ángulo operativo de 120°. Imprescindible en cualquier alarma y proyecto domótico. (Guerrero et al., 2017)

Características Técnicas

- Rango de detección de 7 metros
- Ángulo operativo 120°
- Sensor infrarrojo con circuito de control
- Sensibilidad y tiempo ajustables de forma sencilla
- Voltaje de funcionamiento: 4.5V- 20V
- Consumo en estado inactivo: $<60\mu\text{A}$
- Voltaje de salida: 3.3V TTL output
- Intervalo de tiempo: 0,5-200s (ajustable, por defecto 5s $\pm 3\%$)



- Tiempo de bloque: 2.5s (por defecto)
- Temperatura de trabajo: -20°C a 70°C
- Dimensiones: 3,2 x 2,4 x 1,8cm

2.1.7.5. SENSOR DE TEMPERATURA – HUMEDAD SHT1X.

Este sensor combina un módulo SHT1X de Sensirion capaz de medir humedad y temperatura, junto con una carcasa para exterior porosa de acero inoxidable para protegerlo del óxido y condiciones adversas. Está especialmente diseñado para exteriores o aplicaciones que necesiten medir humedad y/o temperatura en condiciones adversas, por ejemplo, medir la humedad en la tierra de las plantas. La interfaz de comunicaciones se realiza de manera serial a 2 hilos, de tal forma que es muy sencillo conectarlo y manejarlo con Arduino o cualquier Microcontrolador.

Características:

- Rango de medición de temperatura: -40 ~ + 123.8°C
- Rango de medición de humedad: 0 ~ 100% HR
- Precisión de temperatura: ± 0.4 ° C
- Precisión de humedad: $\pm 3\%$ HR
- Longitud del cable: 1 m
- Cables: Rojo = VCC (3-5VDC), Negro o Verde = Tierra, Amarillo = Reloj, Azul = Datos



2.1.7.6. POWER BANK.

Un Power Bank es un dispositivo que almacena energía en forma electroquímica y es el más ampliamente usado para almacenar energía en una variedad de aplicaciones.

Existen dos tipos básicos de baterías:

- Batería primaria:** su reacción electroquímica es irreversible, es decir, después de que la batería se ha descargado no puede volver a cargarse.

- Batería secundaria:** su reacción electroquímica es reversible, es decir después de que la batería se ha descargado puede ser cargada inyectándole corriente continua desde una fuente externa. Su eficiencia en un ciclo de carga y descarga está entre el 70% y 80%.

Teniendo en cuenta la salida Micro USB 2.0 con la que cuenta los dispositivos de Particle, (Argon y Xenon), esta nos servirá para la conexión de un power bank de 30000 mah.

2.1.8. SOFTWARE Y PLATAFORMAS

2.1.8.1. PLATAFORMA PARTICLE.

Particle es una plataforma para el desarrollo de Idc que comenzó como un proyecto en Kickstarter que ofrecía una manera sencilla de conectar focos convencionales domésticos a internet; esto mediante un socket inteligente, llamado Spark Socket, que se conectaba entre un foco común y el socket habitual. El proyecto falló en Kickstarter ya que no pudo llegar a su meta, sin embargo, el equipo notó que el valor estaba en la plataforma que habían desarrollado para IdC; por lo que meses más tarde, lanzaron en Kickstarter un microcontrolador con un módulo WiFi integrado que pudiera Particle, una plataforma para el desarrollo de IdC conectarse a la red para realizar lecturas de sensores



o controlar dispositivos. A la par del hardware que desarrollaron crearon maneras sencillas de controlar los dispositivos mediante una API REST, reduciendo en gran manera la complejidad para el desarrollo de dispositivos conectados. Agregaron un IDE web (Integrated Development Environment) y uno nativo para la programación avanzada de estos dispositivos y finalmente agregaron la posibilidad de gestionar flotillas de dispositivos conectados.(IdC Republic, 2018).

Para lograr la comunicación entre la agricultura y la tecnología, tenemos que utilizar IdC, y gracias a la plataforma Particle, y su desarrollo con IdC, y la captura de datos de sensores, se lograra este objetivo.

Enfoque de Particle para la seguridad de IDC

En Particle, la seguridad de IdC es una consideración constante en cada decisión que tomamos. Tampoco tiene que ser un ingeniero de seguridad de red certificado para usar la plataforma Particle. Nuestra plataforma proporciona una infraestructura segura y escalable para productos IdC, así como herramientas fáciles de usar para administrar sus dispositivos y el software que ejecutan. La plataforma incluye prácticas y características importantes como:

- Seguro por defecto: cada mensaje enviado a través de Particle está encriptado y es seguro. No se permite texto sin formato.
- Llaves de hardware: cada dispositivo tiene su propia clave privada, por lo que el hardware no autorizado no puede colarse en su flota.
- Controles de acceso de equipo: una característica completamente nueva para ayudar a las empresas a administrar los niveles adecuados de acceso a las flotas de dispositivos. Con los controles de acceso del equipo, puede asignar roles (un conjunto de permisos) a cada miembro de su equipo.



- Monitoreo continuo: monitoreamos nuestros servidores y el panorama de seguridad para garantizar que sus dispositivos permanezcan bloqueados.
- No hay puertos abiertos: los dispositivos de partículas no dejan ningún puerto entrante abierto para escáneres de puertos o ataques secundarios activos.
- Conexiones de radio encriptadas: las conexiones de radio están encriptadas por el estándar industrial WPA2.
- Actualizaciones de firmware de OTA: Particle proporciona la mejor experiencia de actualización de firmware de OTA . Puede enviar sin problemas nuevas actualizaciones de firmware para cambiar los comportamientos del producto o abordar las vulnerabilidades de seguridad.
- Autenticación en dos pasos: Particle admite la autenticación en dos pasos , una característica de seguridad estándar de la industria para ayudar a mantener protegida su cuenta y, posteriormente, su flota de dispositivos.
- Política de datos: como cuestión de política, Particle limita intencionalmente el alcance de los datos de usuario almacenados en la Nube de dispositivos. A medida que pasa información confidencial, Device Cloud protegerá los datos, pero no los almacenará. No almacenamos ninguna información o datos de identificación personal que puedan utilizarse para comprometer productos o clientes en la Nube de dispositivos.(Iot Republic, 2018)

2.1.8.1.1. IDE WEB

Particle Build es un entorno de desarrollo integrado, o IDE; eso significa que puede realizar el desarrollo de software en una aplicación fácil de usar, que simplemente se ejecuta en su navegador web.



Particle Build comienza con la barra de navegación a la izquierda. En la parte superior, hay tres botones que cumplen funciones importantes:

Flash: muestra el código actual en el dispositivo. Esto inicia una actualización de firmware por aire y carga el nuevo software en su dispositivo.

Verificar: esto compila su código sin actualizarlo al dispositivo; Si hay algún error en su código, se mostrará en la consola de depuración en la parte inferior de la pantalla.

Guardar: guarda los cambios que haya realizado en su código.

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

Código: muestra una lista de sus aplicaciones de firmware y le permite seleccionar cuál editar / flashear.

BibliIdCeca: explore las bibliIdCecas enviadas por otros usuarios y desarrolle la suya.

Documentos: lo lleva a la documentación de Partícula.

Dispositivos: muestra una lista de sus dispositivos, para que pueda elegir cuál flashear y obtener más información sobre cada dispositivo.

Configuración: cambie su contraseña, cierre sesión u obtenga su token de acceso para llamadas API.

2.1.8.1.2. PARTICLE 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.

Con la nueva actualización Tinker de la aplicación móvil Particle hace que sea muy fácil comenzar a jugar con su dispositivo Particle sin escribir ningún código. Es ideal para el desarrollo temprano, el aprendizaje y la creación de prototipos

2.1.8.1.3. ACTUALIZACIONES DE OTA CON PARTICLE

La plataforma de Particle tiene una posición única para proporcionar capacidades de actualización OTA líderes en la industria para dispositivos integrados. El uso de Particle para las actualizaciones de OTA le brinda los siguientes beneficios:

Una solución completa

Una actualización exitosa de OTA requiere una coordinación compleja entre el hardware de IdC, el firmware del dispositivo, la conectividad de red y una nube de dispositivos de IdC. Confía en nosotros, este es un problema muy difícil de resolver correctamente.

Por suerte para usted y su equipo, aquí es donde brilla la plataforma IdC totalmente integrada de Particle. Estas cuatro partes de la "pila de IdC" son los pilares centrales de la plataforma de Particle y le permiten entregar actualizaciones OTA a dispositivos a cualquier escala.

2.1.8.1.4. WEBHOOKS.

Un Webhooks es una manera de ser notificado cuando un evento ha ocurrido en tu aplicación la de un tercero. Es básicamente una solicitud POST que se envía a una URL



específica. Esa URL está configuración para recibir el cuerpo de la solicitud POST y procesarla de alguna manera. Por lo tanto, si alguna vez has creado un formulario y lo has enviado en código, sabes cómo crear una solicitud POST.

2.1.8.1.5. API

La abreviación de Application Programming Interface (Interfaz de Programación de Aplicaciones). Es una interfaz de servicio realtime que permite a los desarrolladores y aplicaciones de terceros tomar data de un determinado sitio en internet.

Una API es básicamente una forma de hacer una solicitud a otra aplicación. Por ejemplo, si estamos en este proyecto desarrollando un sistema de monitoreo, en la que nos estamos guiando de 4 pasos y uno de ellos es el Almacenamiento de datos, por lo que hacemos uso de una plataforma externa denominada GoogleCloud, a la cual le diremos que almacene datos que la plataforma Particle esta recibiendo, mediante la creación de Webhooks. No hay diferencias si nuestra aplicación está en Ruby, Php, Javascript, etc., y Google es otro lenguaje porque la API transmite y recibe datos en un formato común (normalmente JSON o XML). Esto permite que dos aplicaciones totalmente separadas envíen y reciban datos.

Así que una API se utiliza para hacer preguntas directas y un Webhook se utiliza para notificar cuando se producen ciertos eventos. En lugar de preguntar constantemente si algo ha cambiado, un Webhooks puede activarse y notificarnos automáticamente apenas se produzca el evento.(Manzo, 2017).



2.1.8.1.6. PROTOCOLO DE RED THREAD.

En la actualidad, el protocolo de red más innovador basado en IPv6 es Thread. Diseñado para domótica, está basado en 6LowPAN, y del mismo modo que aquel, no es un protocolo de aplicaciones IdC como Bluetooth o ZigBee. Se diseñó como un complemento WiFi, puesto que, aunque la tecnología Wi-Fi funciona muy bien en dispositivos de consumo, tiene limitaciones al utilizar en configuraciones de domótica. Lanzado a mediados del 2014 por Thread Group, este protocolo sin canon de uso se basa en varios protocolos como IEEE 802.15.4, IPv6 y 6LoWPAN. Es una solución resistente basada en IP para aplicaciones IdC. Diseñado para trabajar sobre chips IEEE 802.15.4 ya existentes de fabricantes como Freescale y Silicon Labs, Thread es compatible con redes de topología de malla al utilizar radio transceptores IEEE802.15.4, siendo capaz de manejar hasta 250 nodos con altos niveles de autenticación y cifrado. Una actualización de software relativamente sencilla permite a los usuarios utilizar thread en dispositivos ya compatibles con IEEE 802.15.4. (RS Components, 2016).

- Estándar: Thread, basado en IEEE802.15.4 y 6LowPAN
- Frecuencia: 2,4GHz (ISM)
- Alcance: N/A
- Velocidad de transferencia: N/A

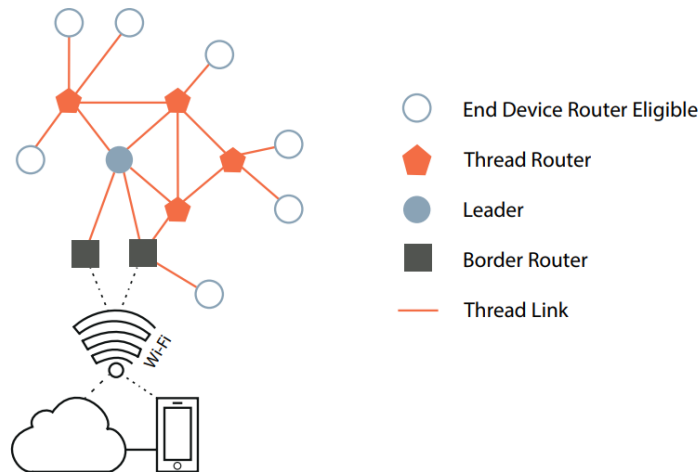


Figura 16: Conexión usando protocolo Thread

Fuente: RS Componets

2.1.8.2. PLATAFORMA THINGSPEAK.

ThingSpeak es una API y aplicación de código abierto para el Internet de las Cosas que permite almacenar y recopilar datos de objetos conectados a través del protocolo HTTP a través de Internet o de una red local.

Con ThingSpeak, el usuario puede crear aplicaciones de registro de datos de sensores, aplicaciones de seguimiento de ubicación y una red social para los objetos conectados, con actualizaciones de estado.

Funciones ThingSpeak:

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



- Plugins

En el proyecto nos servirá para la Tercera etapa de nuestro proyecto, en cual es graficar los datos del monitoreo de los sensores que se van obtener a través de la plataforma Particle.

2.1.8.2.1. VISUALIZACIONES DE MATLAB.

Plantillas:

Utilice las plantillas de visualización de MATLAB para comenzar a trazar líneas interactivas. Puede desplazar y cambiar el tamaño de las parcelas ThingSpeak MATLAB. También puede desplazarse sobre los puntos de datos para obtener más información.

Con ala Visualización de MATLAB se puede:

- Hacer un histograma para comprender la variación en los datos para los valores de temperatura leídos desde un canal público. Este ejemplo lee 10 horas de datos de temperatura y muestra la variación en un histograma.
- Visualizar datos direccionales con Compass Plot y grafique datos basados en el resultado de un cálculo que involucra dos o más campos. Este ejemplo lee la velocidad del viento y la dirección del viento desde un canal público y calcula y muestra la velocidad del viento en una gráfica de brújula.
- Usar el diagrama de área para comparar conjuntos de datos de tráfico para el tráfico en dirección este y oeste. Este ejemplo lee 33 horas de datos de tráfico de un canal público y muestra datos promediados para cada hora.

- Comparar los datos de temperatura de tres días diferentes en un diagrama lineal. Este ejemplo lee tres segmentos de 24 horas de datos de temperatura de un canal público.
- Trazar la temperatura y la velocidad del viento en dos ejes diferentes para ver simultáneamente dos variables diferentes. Este ejemplo lee los datos de temperatura y velocidad del viento de un canal público y crea un diagrama con múltiples ejes y.
- Visualizar la correlación entre temperatura y humedad en un diagrama de dispersión utilizando mediciones de un canal público.

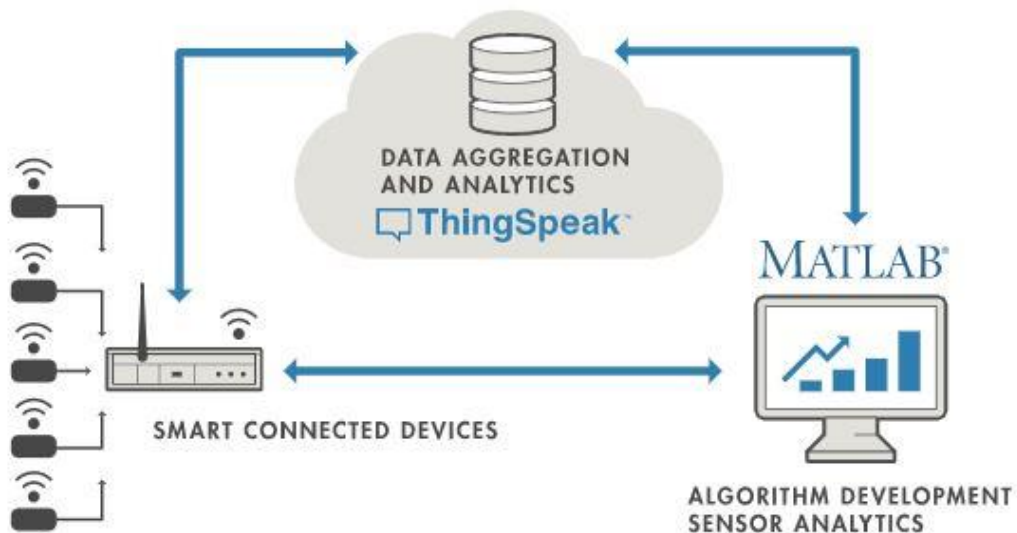


Figura 17: Procesamiento de imágenes con Matlap

Fuente: Thinspeak



2.1.8.3. GOOGLE CLOUD.

Google Cloud Platform, es una suite que contiene todos los servicios que funcionan en la misma infraestructura que utiliza Google de manera interna, por ejemplo, con servicios como Youtube o Google Search. Es utilizada para crear ciertos tipos de soluciones a través de tecnología almacenada en la Nube y permite destacar la rapidez y escalabilidad de la infraestructura en las aplicaciones de buscador.

Por lo tanto, Google cloud se refiere al espacio virtual en el cual podremos realizar una serie de tareas, en el presente proyecto nos servirá de alojamiento de datos para poder

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 (Palacios, 2017)

2.1.8.4. PUSHBULLET.

Pushbullet es una aplicación que permitía que varios dispositivos se intercomunicaran entre ellos. Disponible para Android, iOS, Windows y otros sistemas mediante creaciones de la comunidad (incluyendo Mac y Linux) la idea principal es ver lo que pasa en el móvil en el PC. También en sentido inverso: poder enviar cosas y responder mensajes en el móvil, desde el PC.(Ramírez, 2020).



CAPITULO III

MATERIALES Y MÉTODOS

3.1. MATERIALES.

3.1.1. HARDWARE.

- Dispositivo particle argon
- Dispositivo Particle Xenon
- Sensor de temperatura – humedad DTH22 (Ambiente)
- Sensor de temperatura – humedad SHT1X (Suelo)
- Señor de movimiento PIR
- Relay
- Bomba de agua
- Power Back
- Smartphone
- Laptop HP G240

3.1.2. SOFTWARE

- Sistema operativo de 32-64 bits
- GIT Bash
- Node.js
- Particle
- App Particle
- App PushBullet
- App Blynk

3.2. TIPO DE DISEÑO DE LA INVESTIGACION

Para poder definir el tipo de investigación, se basara en la clasificación descrita en trabajo realizado por (Tam, 2008) titulado “Tipos, métodos y estrategias de investigación científica” ”

Para llegar a nuestro objetivo se estudió o prueba de hipótesis, la investigación se desarrollará siguiendo el diseño o enfoque de investigación mixto

Para contrastar la hipótesis previamente formulada, se intenta abarcar todos los niveles de investigación involucrados en el diseño de investigación mixto como son: exploratorio, explicativo y experimental. Pero el proceso de metodología de la investigación que más se ha adecuado en el proceso es el método experimental.

3.3. POBLACIÓN Y MUESTRA

La población de para la presente investigación fue el área de cultivo de quinua situado en la ciudad de Juliaca, región de Puno por la carretera Juliaca-Lampa continuidad rancho Victoria a 13 Km de la ciudad de Juliaca

3.4 . TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Como se dijo a un inicio de esta investigación, nos basaremos en cuatro (4) Pasos para lograr en objetivo del mismo, la primera capa de captura de datos está formada por un conjunto de sensores cuyo objetivo es la obtención de variables climáticas asociadas al cultivo. La capa de almacenamiento se encarga de recopilar información en tiempo real a partir de las diferentes fuentes (sensores) distribuida en el cultivo. La capa de procesamiento tiene como función la predicción de información relevante a partir de los datos recopilados, Finalmente, la capa de consulta permite la visualización y seguimiento en tiempo real de las variables del cultivo y de las predicciones obtenidas en la capa de procesamiento. Como se muestra en la Figura 115

3.4.1. CAPTURA DE DATOS

Para la captura de datos se utilizó la plataforma particle por lo que los pasos a seguir son:

3.4.1.1 CREACIÓN DE CUENTA EN PARTICLE

Para la inicialización de este proyecto, se tiene que crear una cuenta en Particle a partir de su página principal la cual es <https://login.particle.io/setup>.

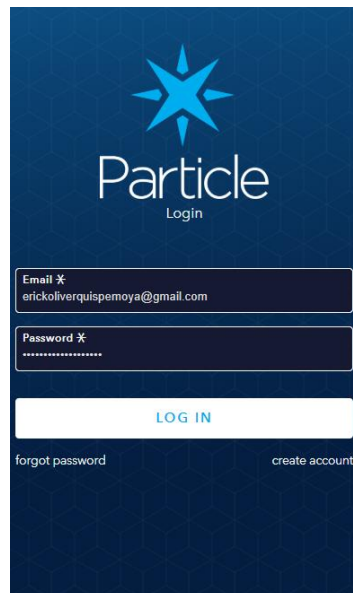


Figura 18: Creación de cuenta Particle
Elaboración propia

3.4.1.2 DESCARGA DE APP DE PARTICLE

Para poder añadir los kits de desarrollo de Particle se tiene que descargar el App DE Particle a partir de un Smartphone. (requiere Android 4.0.3 o superior).

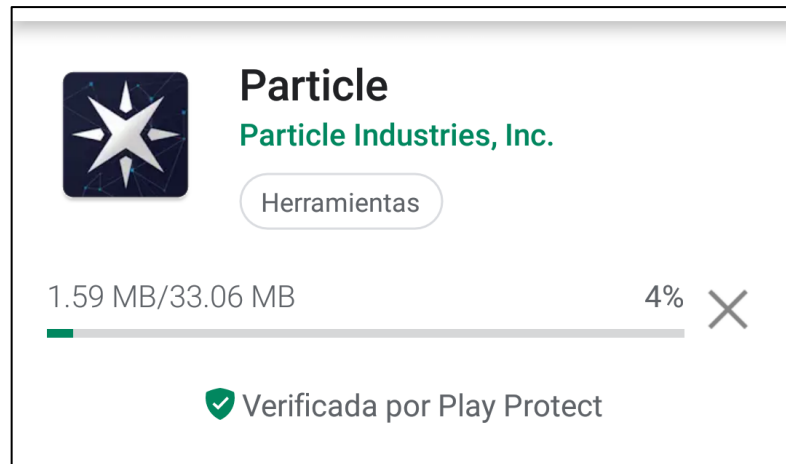


Figura 19: Descarga de App Particle
Elaboración propia

Luego de instalar la aplicación particle ingresar con la cuenta antes creada en la página principal de la plataforma particle

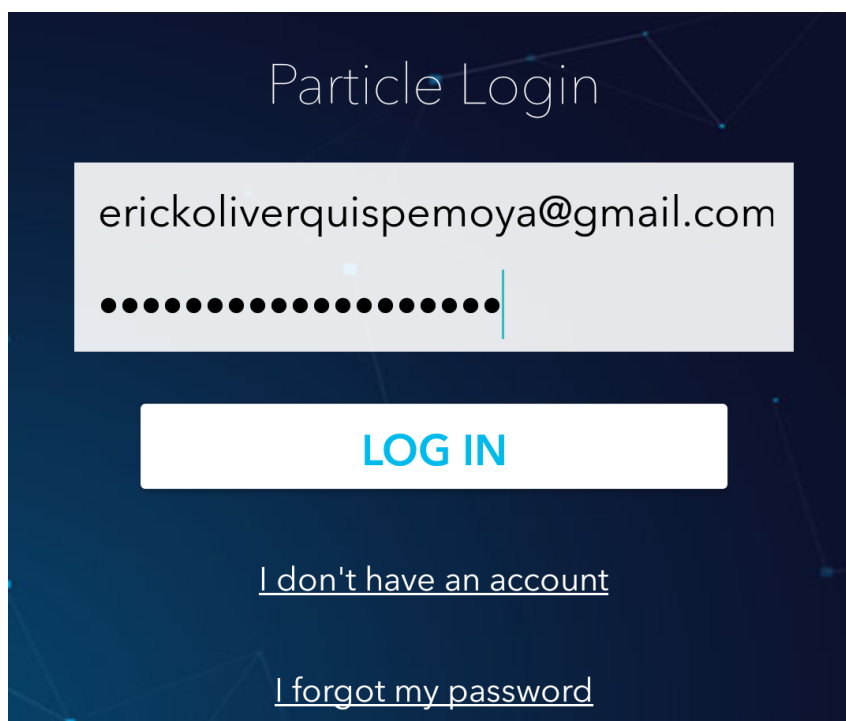


Figura 20: Login con a App de particle
Elaboración propia

3.4.1.3. Configuración del dispositivo Particle Argón

Una vez dentro del App de particle, tenemos que añadir el dispositivo particle argón, por lo que encenderemos el dispositivo argón y lo pondremos en modo escucha, luego nos trasladaremos a la Aplicación de Particle en el Smartphone, dándole clic en el símbolo “+”

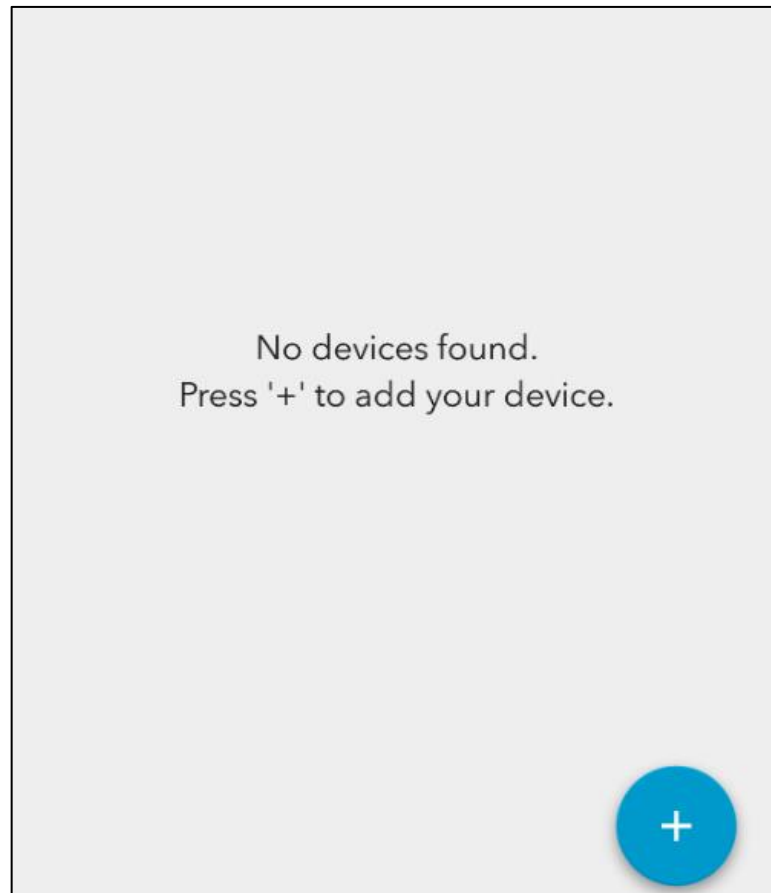


Figura 21: Agregar el dispositivo Aron a nuestra cuenta Particle
Elaboración propia

El primer dispositivo a añadir será nuestro dispositivo Aron, por lo que, una vez hecho el anterior paso, tenemos que darle clic en la opción “ set up a Aron, Boron or Xenón +”

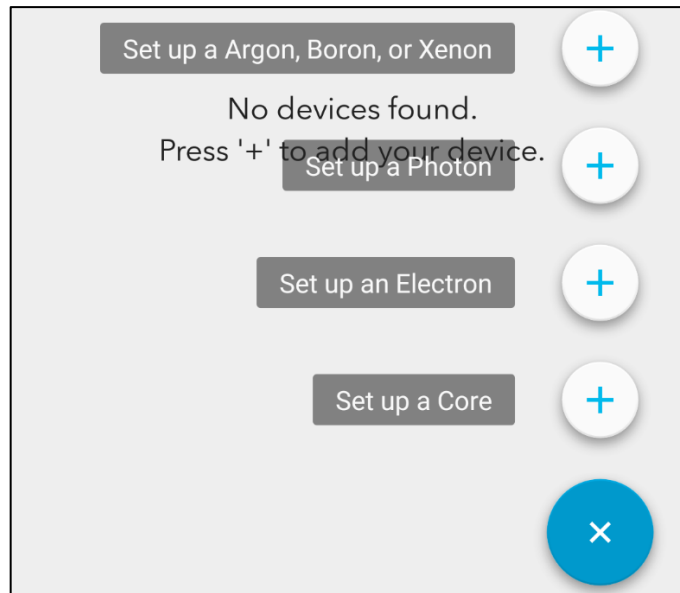


Figura 22: Selección de dispositivo a agregar
Elaboración propia

Luego tendremos que activar nuestro Bluetooth en nuestro Smartphone para poder crear nuestra red Mesh, por lo que le damos clic en la opción “Permitir”.

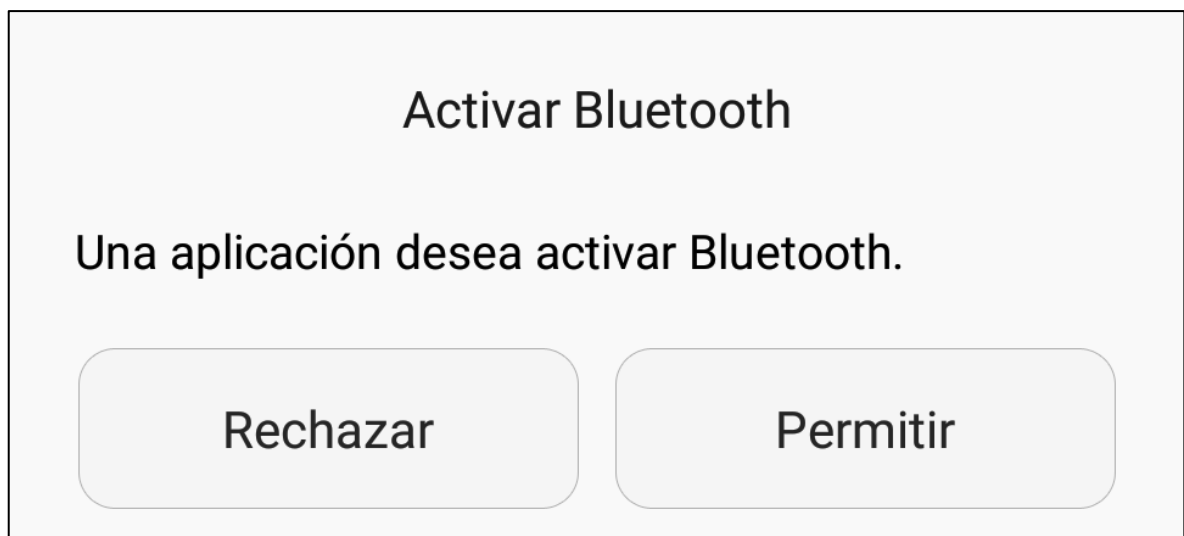


Figura 23: Activación e Bluetooth en smartphone
Elaboración propia

Se activará el bluetooth de tu celular para tener acceso.

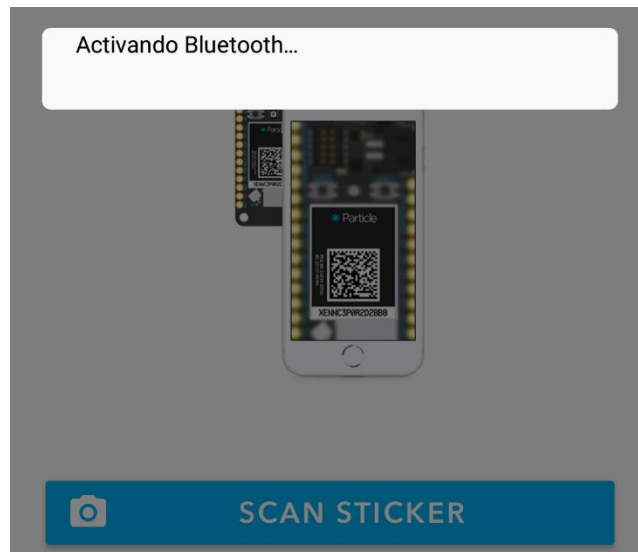


Figura 24: Activando Bluetooth.
Elaboración propia

Ya activado tu bluetooth, tenemos que agregar el Aron, por lo que debemos de escanear la etiqueta que el dispositivo lleva en la parte frontal dándole clic en “scan sticker”

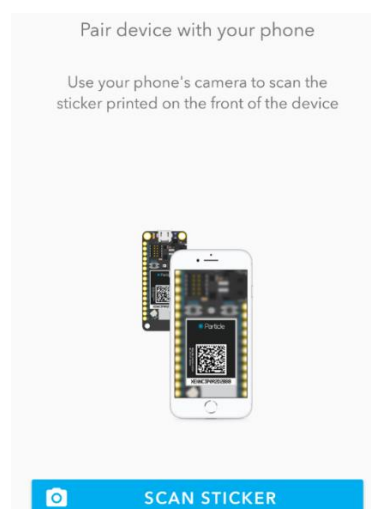


Figura 25: Escaneo de Código QR del Argón.
Elaboración propia

Para poder escanear el App necesitara acceder a la ubicación de tu dispositivo por lo tanto de damos clic en “permitir”.

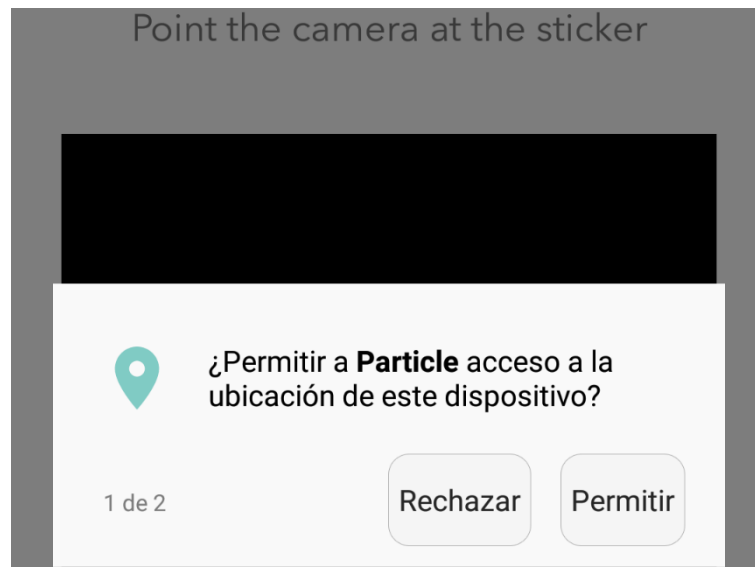


Figura 26: Activando permisos de Ubicación
Elaboración propia

Ingresaremos a la cámara del dispositivo para escanear el código QR, para eso tenemos que direccionar la cámara al código QR del dispositivo Aron.



Figura 27 Escaneo de QR del argón.
Elaboración propia

El dispositivo Smartphone empezará el emparejamiento con el dispositivo Argón como se muestra en la imagen:

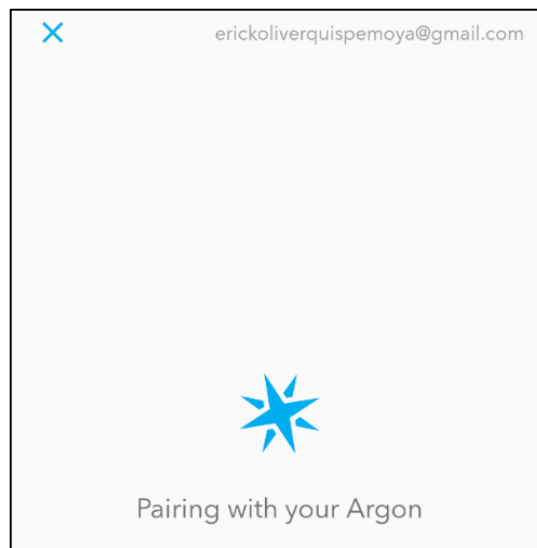


Figura 28: Emparejando Argón al Smartphone mediante Particle
Elaboración propia

Ya emparejado el Smartphone con el dispositivo Argón, no pedirá si deseamos usar el Dispositivo Argón como una red Mesh, por lo que le damos clic en “YES, USE IN MESH NETWORK”.

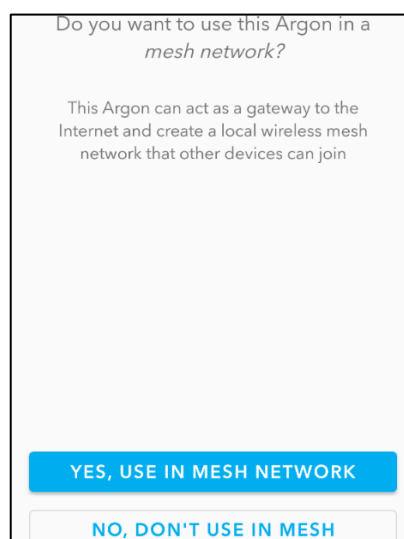


Figura 29: Condiciones para crear la Red Mesh.
Elaboración propia

Conectamos el dispositivo Argon a la red local Wi-fi. Local,

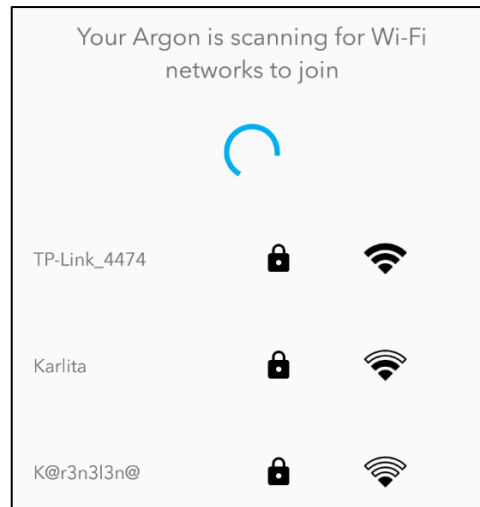


Figura 30: Redes disponibles para conectar en dispositivo Argon
Elaboración propia

Si las credenciales están correctas el led RGB del Argón cambiará a parpadeo color verde por un instante, luego a color azul claro en parpadeo rápido, y cuando ya este conectado a internet, el parpadeo será en azul claro

Una vez conectado a la red Wi-fi , se deberá nombrar nuestro nuevo dispositivo en este caso se nombrara como “tesis_mesh_argon_1” y luego le damos en “Next”

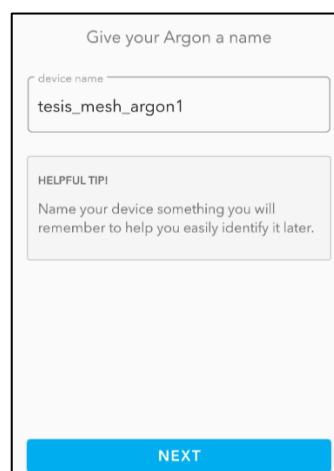


Figura 31 Colocación de Nombre ala red Mesh
Elaboración propia

Luego nombramos nuestra nueva red Mesh, en este caso se nombrará la red con el nombre de “monitoreoquinua”

Give your network a name

Network name

HELPFUL TIP!
Name your network, as this name will

NEXT

Figura 32: Nombre de la red Mesh
Elaboración propia

Tenemos que ingresar una clave para la red Mesh y luego le damos clic en crear red como se muestra la imagen

Set network admin password

Network password

Confirm network password

Figura 33: Ingreso de clave de red Mesh
Elaboración propia

3.4.1.4. MODOS DEL DISPOSITIVO ARGON

Para poder trabajar con fluidez con el dispositivo Argon es necesario conocer los diversos modos de trabajo que tiene de acuerdo al color que emite el dispositivo.

Modo conectado esta cuando el dispositivo ARGON esa conectado a una red Wifi Existente

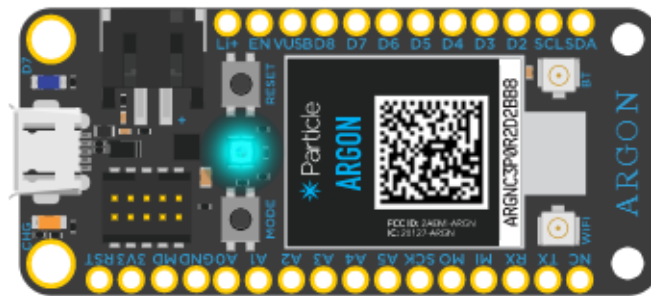


Figura 34: Modo Conectado a red wifi
Fuente: Particle (2019)

Actualización de firmware de OTA: Si su Argon parpadea en color magenta (rojo y azul al mismo tiempo), actualmente está cargando una aplicación o actualizando su firmware. Este estado se desencadena por una actualización de firmware o por un código intermitente del IDE web o IDE de escritorio. Es posible que vea este modo cuando conecte su Argon a la nube por primera vez.

Se tiene que tener mucho cuidado cuando se actualiza el firmware no pase ningún problema con la desconexión del dispositivo o la red

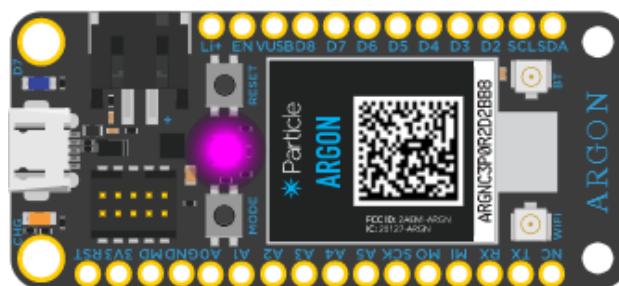


Figura 35: Actualización de firmware de OTA:
Fuente: Particle (2019)

Si su Argon parpadea en verde, está intentando conectarse a Wi-Fi

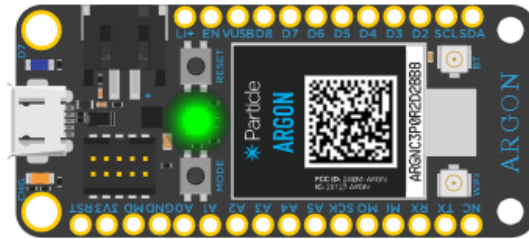


Figura 36: Buscando red Mesh creada

Fuente: Particle (2019)

Malla y WiFi

Características de la antena del dispositivo Argón:

Tabla 2: Características de antena del Argón

Dispositivo de partículas	Frecuencia	Tipo de antena	Fabricante	MFG Parte #	Ganancia
Argón	2400-2500 MHz	Antena PCB	Particle	ANT-FLXV2	Pico 2.0dBi

Fuente: Particle (2019)

También es posible usar la mayoría de las antenas diseñadas para Wi-Fi (2.4 GHz) como antena de malla. Por ejemplo, puede usar antenas de pato o incluso Yagi para un mayor alcance. En algunos casos, se requerirá un adaptador u.FL a RP-SMA. Si está construyendo un producto con antenas alternativas, es posible que se requiera una certificación adicional.

3.4.1.5. CONFIGURACIÓN DEL DISPOSITIVO PARTICLE XENÓN.

Una vez culminada la configuración del dispositivo particle Argon, pasaremos a añadir los dispositivos Xenon para formar la red mesh, por lo que daremos en “next” para añadir nuevo xenón.

Para lo cual deberemos encender el dispositivo xenón, ponerlo en modo escucha, y así escanear el código QR que tienen el dispositivo Xenon en la parte frontal

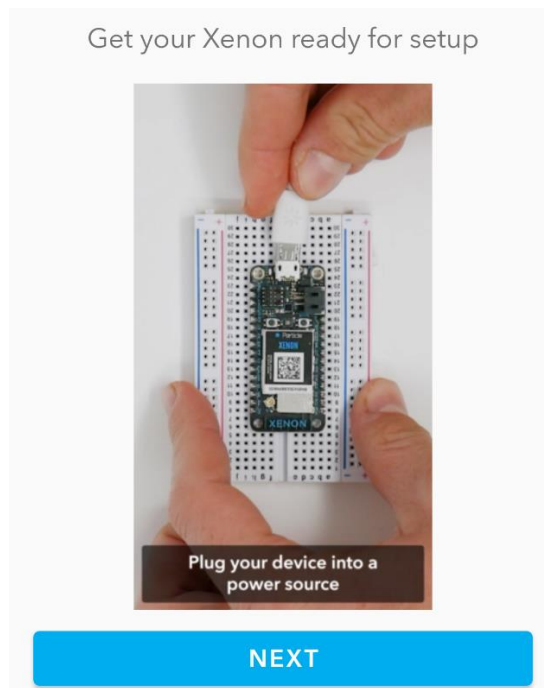


Figura 37: Incorporación de dispositivo Xenón a la red Mesh
Elaboración propia

Ya escaneado en código QR, nos aparecerá la pantalla de carga de emparejamiento (Particle Xenón - Smartphone).

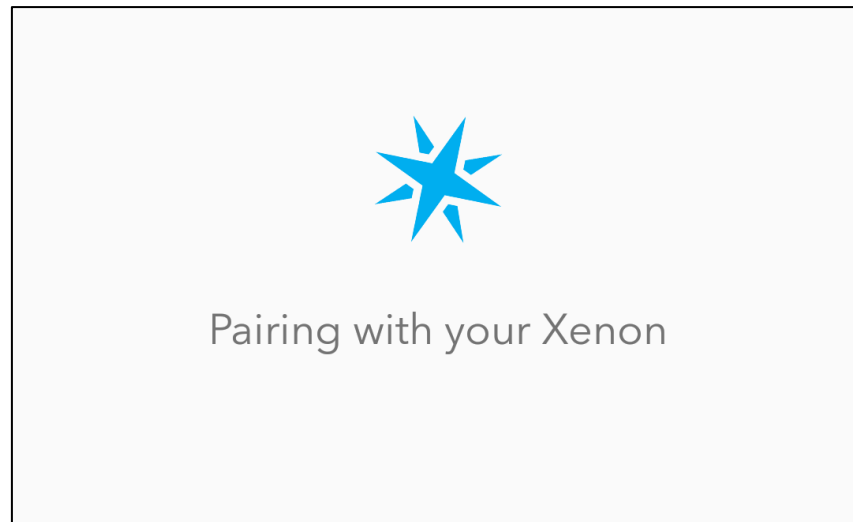


Figura 38: Emparejamiento de dispositivo Xenon.
Elaboración propia

Para asegurar el emparejamiento se visualizará el mensaje de confirmación satisfactoria ente el dispositivo particle argón y el smartphone

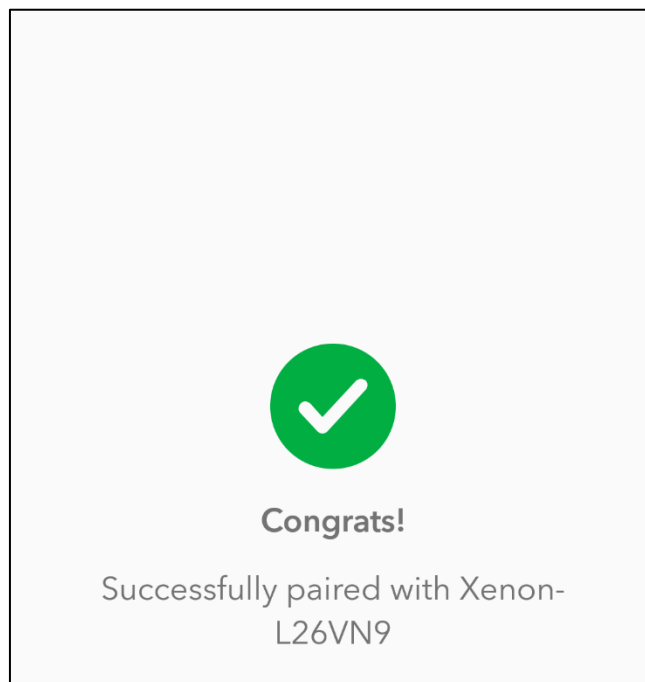


Figura 39: Conexión exitosa del Xenon.
Elaboración propia

Luego del emparejamiento nos pedirá la actualización del OS de dispositivo Xenón por que le daremos clic en “Begin update”.

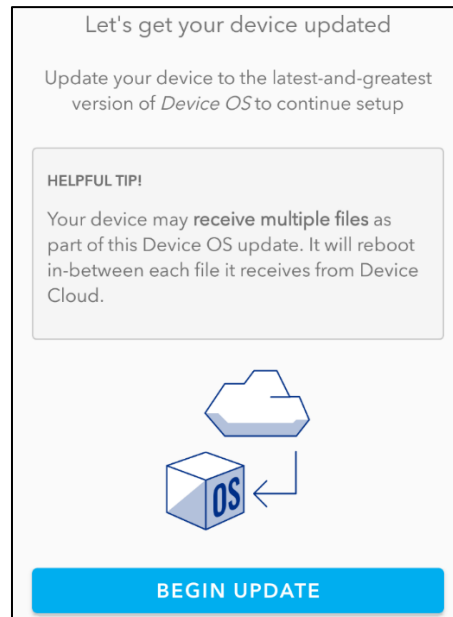


Figura 40: Actualización de dispositivo de Xenon
Elaboración propia

Se visualizará la carga de actualización de OS del dispositivo



Figura 41: Actualizando Updating.
Elaboración propia

Luego se tiene que incluir el dispositivo Particle Xenon a la red Mesh creada anteriormente, por lo que empezará a buscar redes Mesh creadas, seleccionaremos la red Mesh Monitoreo quinua creada anteriormente en la Figura N° “”

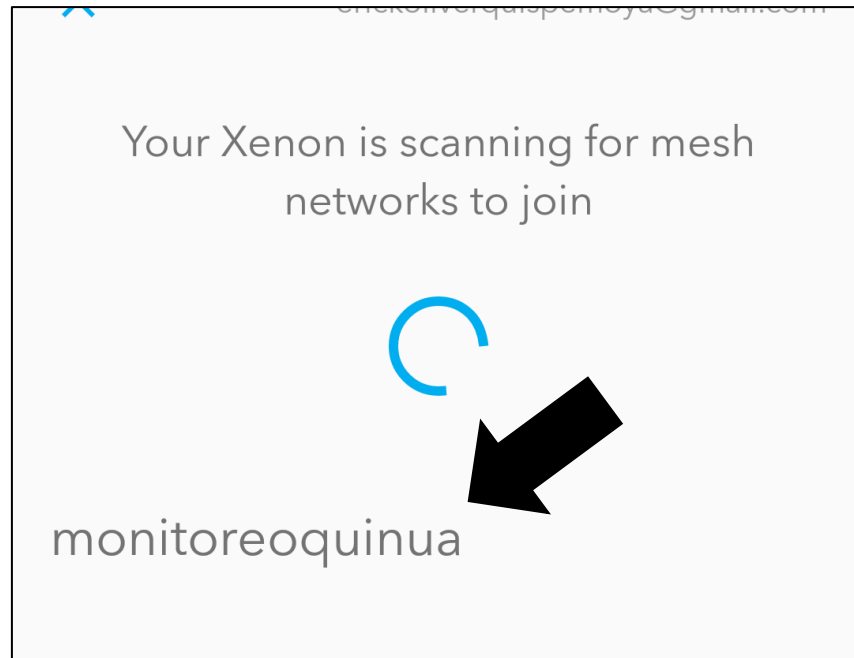


Figura 42: Visualización de la red Mesh
Elaboración propia

Para emparejar el xenón con el argón en la red Mesh se tiene que poner a ambos dispositivos en modo escucha, Escaneamos en QR del dispositivo Particle Argón, para que el Xenón pueda ingresar a la red Mes

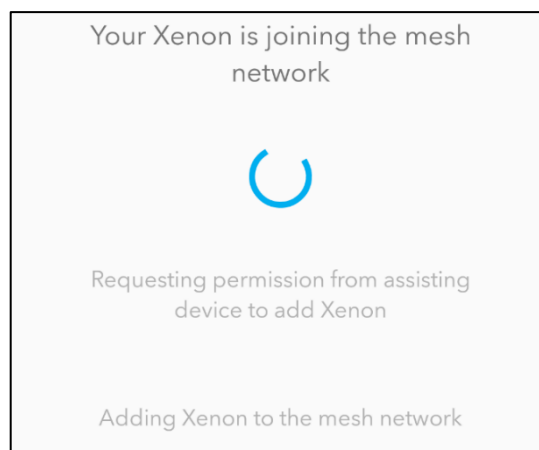


Figura 43: Incorporando dispositivo Xenon a la red Mesh
Elaboración propia

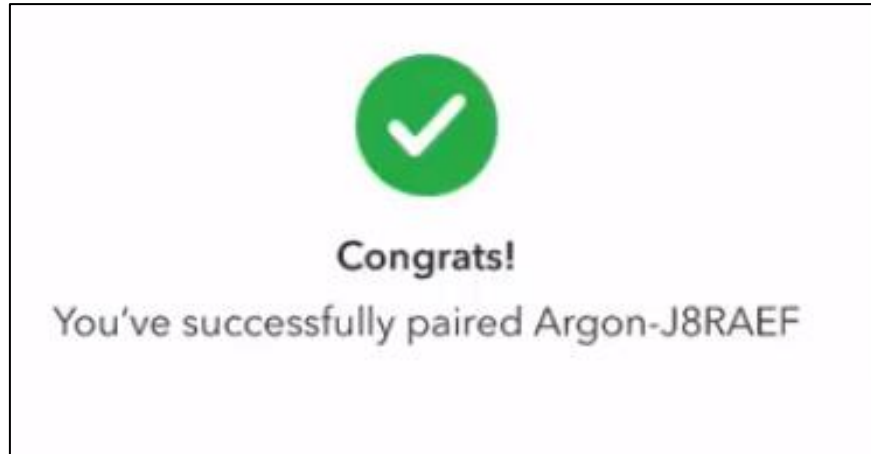


Figura 44: Conexión exitosa de Red Mesh

Elaboración propia

Nos pedira la clave para ingreso a la red mesh, creada a un inicio en la imagen “” por el dispositivo Argon.

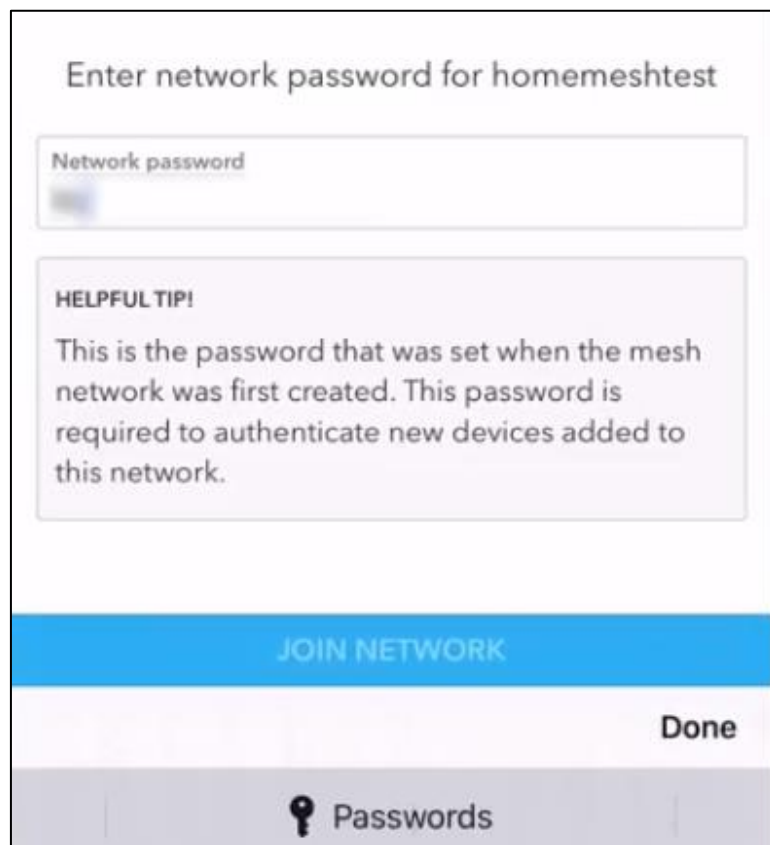


Figura 45: Ingreso de clave

Elaboración propia

Se empezará a cargar los permisos para ingresar el dispositivo Xenón a la red Mesh creada en un principio por el dispositivo Particle Argón.

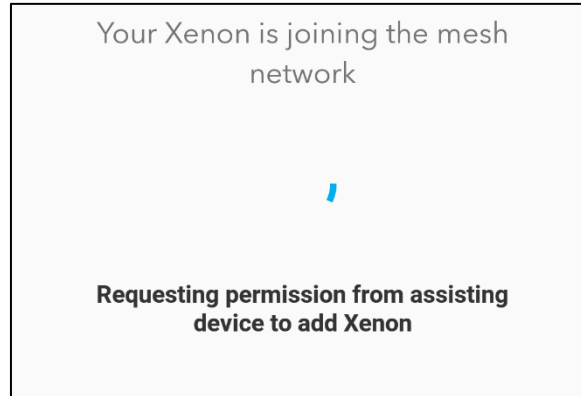


Figura 46: Cargando Requisitos para ingresar a la red Mesh
Elaboración propia

Finalmente nos aparecera la el mensaje de que el dispositivo Xenon a sido añadido satisfactoriamente al ared Mesh

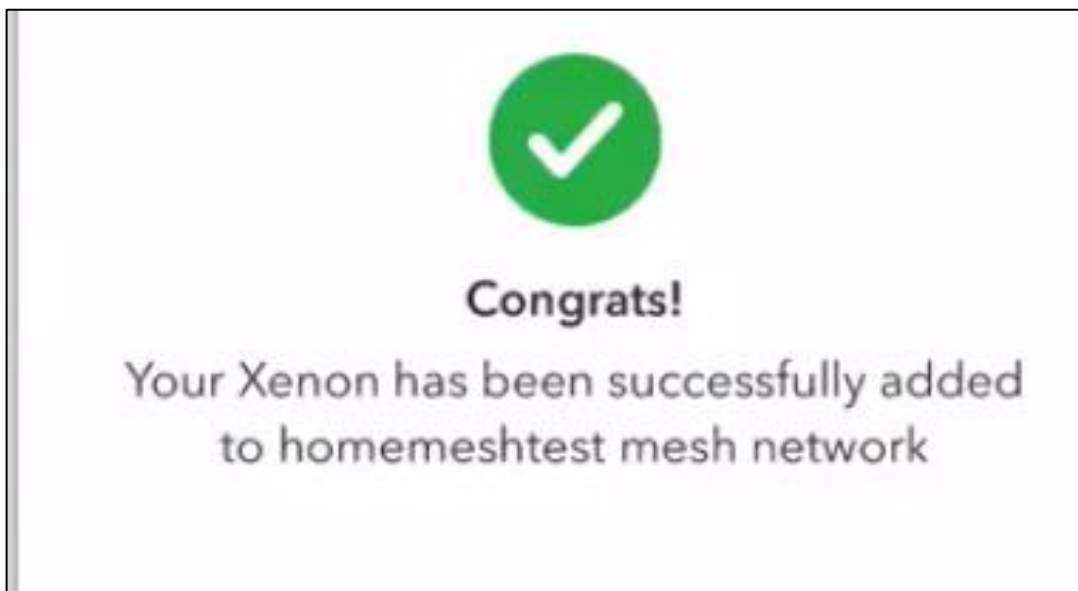


Figura 47: Mensaje de ingreso satisfactorio a la red Mesh
Elaboración propia

El mismo procedimiento se hace para todos los dispositivos Particle Xenón que se usaran en el proyecto, dicha cantidad dependerá de los distintos sensores utilizados, una vez añadidos todos los dispositivos Particle, nuestra App nos mostrara en su pantalla principal los dispositivos vinculados.

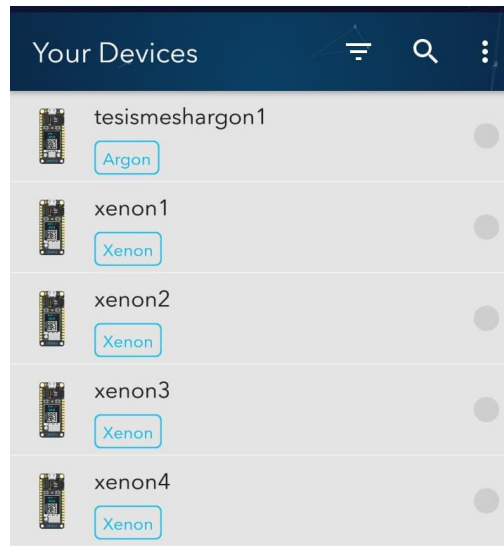


Figura 48: Creación de xenon de acuerdo a la cantidad de sensores a utilizar

Elaboración propia

3.4.1.6. PARTICLE EN WINDOWS

3.4.1.6.1. PARTICLE CLI

Particle CLI es una potente herramienta para interactuar con sus dispositivos y Particle Device Cloud. La CLI usa Node.js y puede ejecutarse fácilmente en Windows, macOS (OS X) y Linux. También es de código abierto para que pueda editarlo y cambiarlo.

3.4.2. INSTALACIÓN DEL PROGRAMA NODE

Particle JS es una biblioteca de JavaScript para Particle Device Cloud API para Node.js y el navegador. Es de código abierto para que pueda editar, cambiar o incluso enviar solicitudes de extracción si desea compartir.

Para la descarga de Node.js nos dirigimos a la página nodejs.org/ja/blog/reléase/v6.14.3/ y descargamos node.js para Windows de 64 bits.

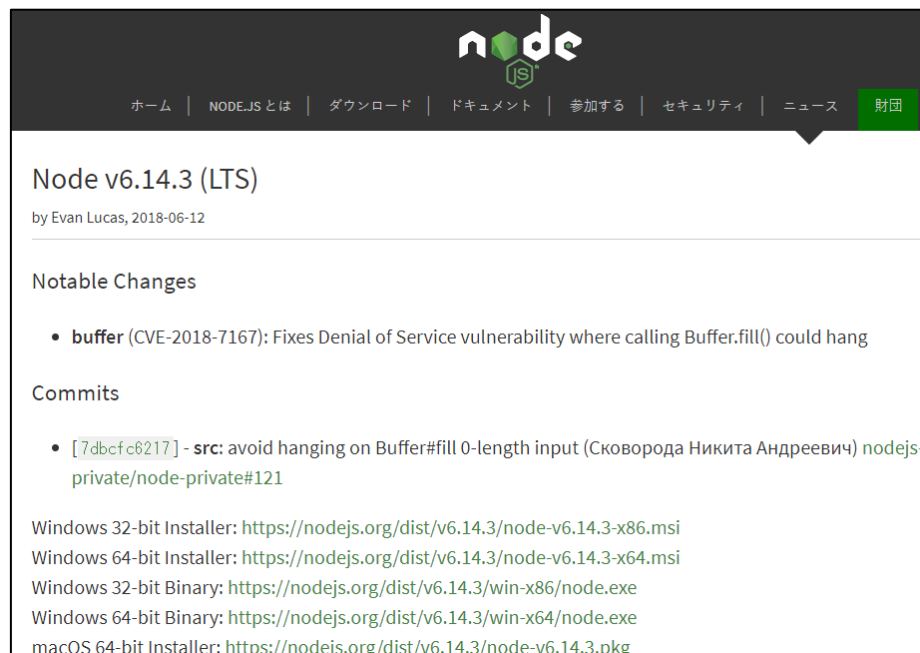


Figura 49: Descarga de Node.js, paso 1.

Elaboración propia

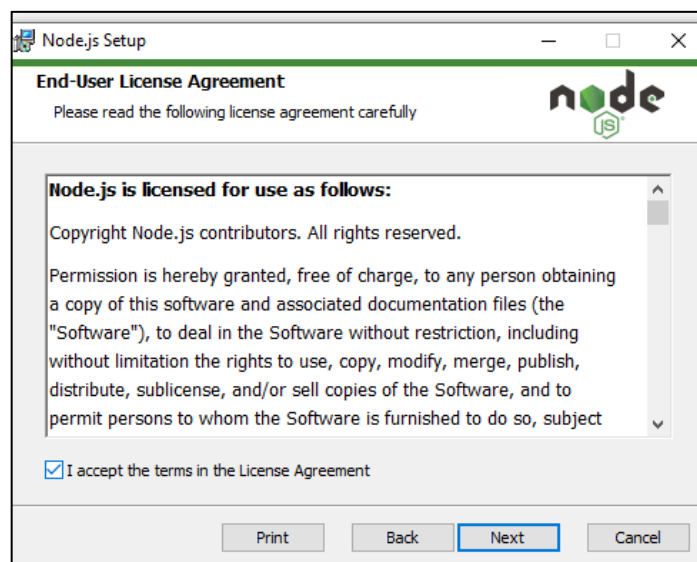


Figura 50: Permisos de Instalación

Elaboración propia

3.4.1.6.2. INSTALACIÓN DE PARTICLE CLI.

Para la descarga de Particle CLI no dirigimos a la plataforma de Particle con el siguiente enlace <https://docs.particle.io/tutorials/developer-tools/cli/>, para fines prácticos lo descargamos para Windows haciendo clic en “Windows CLI Installer”.

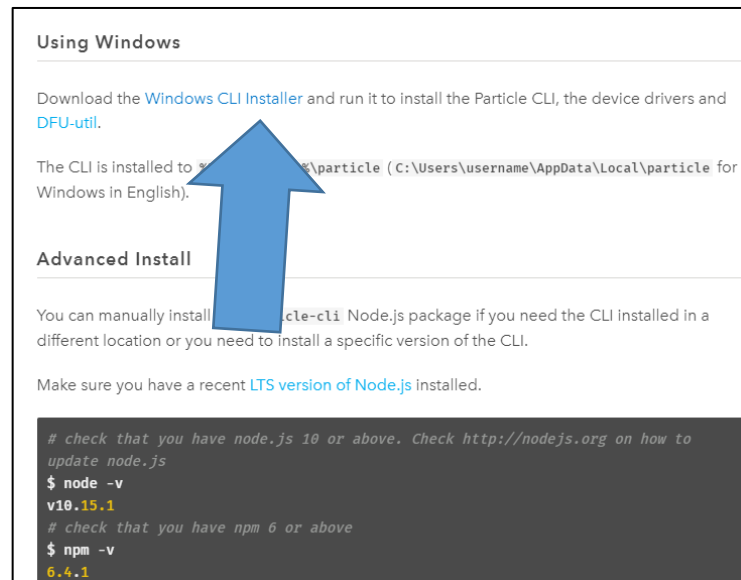


Figura 51: Descarga de Particle CLI, paso 1

Elaboración propia

Iniciamos la instalación de Particle CLI dándole Clic en “next”



Figura 52: Ejecución del setup de Particle CLI, paso 2.

Elaboración propia

3.4.1.6.3. ACTUALIZACIÓN DE FIRMWARE MÉDIATE CLI Y OTA

Una vez instalado Particle CLI, podemos tener acceso a particle y toda la línea de comandos del mismo mediante del CLI, escribimos la palabra “Particle” + enter

```
C:\Windows\system32\CMD.exe
Microsoft Windows [Versión 10.0.17763.678]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIO>
C:\Users\USUARIO>particle ?
Welcome to the Particle Command Line Interface!
Version 1.43.3
https://github.com/particle-iot/particle-cli

Usage: particle <command>
Help: particle help <command>

Commands:
binary      Inspect binaries
call        Call a particular function on a device
cloud       Access Particle cloud functionality
compile     Compile a source file, or directory using the cloud compiler
config      Configure and switch between multiple accounts
device      Manipulate a device
doctor      Put your device back into a healthy state
flash       Send firmware to your device
function    Call functions on your device
get         Retrieve a value from your device
identify    Ask for and display device ID via serial
keys        Manage your device's key pair and server public key
library     Manage firmware libraries
list        Display a list of your devices, as well as their variables and functions
login       Login to the cloud and store an access token locally
logout      Log out of your session and clear your saved access token
mesh        Manage mesh networks
```

Figura 53: Ingreso a Particle del CLI de Windows
Elaboración propia

Sincronizamos la línea de comandos ingresando a nuestra cuenta particle, mediante el CLI, escribimos “particle login” + enter:

```
particle login
update      Update the system firmware of a device via USB
update-cli  Update the Particle CLI to the latest version
usb         Control USB devices
variable    Retrieve and monitor variables on your device
webhook     Manage webhooks that react to device event streams
whoami      prints signed-in username

Global Options:
-v, --verbose  Increases how much logging to display [count]
-q, --quiet    Decreases how much logging to display [count]

Options:
--version     Show the version of particle-cli installed. [boolean]

Examples:
particle setup      Set up your Particle account and your device
particle list       Show all your devices and their functions and variables
particle flash my_device tinker  Remotely update your device to run the default Tinker app
particle call my_device_name digitalwrite D7=HIGH Call a function on your device running Tinker to toggle the onboard LED

For more information, visit our documentation at https://docs.particle.io

No such command '?'.

C:\Users\USUARIO>particle login
Please enter your email address:
```

Figura 54: Ingreso a la cuenta de Particle a partir del CLI de Windows
Elaboración propia

```
C:\Windows\system32\CMD.exe
Options:
--version Show the version of particle-cli installed. [boolean]

Examples:
particle setup          Set up your Particle account and your device
particle list          Show all your devices and their functions and variables
particle flash my_device tinker Remotely update your device to run the default Tinker app
particle call my_device_name digitalwrite D7=HIGH Call a function on your device running Tinker to toggle the onboard LED

For more information, visit our documentation at https://docs.particle.io

No such command '?'

C:\Users\USUARIO>particle login
? Please enter your email address erickoliverquispemoya@gmail.com
? Please enter your password [hidden]
? Successfully completed login!

C:\Users\USUARIO>
```

Figura 55: Ingreso a cuenta de particle mediante línea de comandos del CLI exitosa
Elaboración propia

Una vez dentro de nuestra cuenta Particle a partir de los comandos CLI, iniciaremos con la actualización de comandos Particle escribiendo “install -g particle-cli”.

```
C:\Users\USUARIO>particle login
? Please enter your email address erickoliverquispemoya@gmail.com
? Please enter your password [hidden]
? Successfully completed login!

C:\Users\USUARIO>npm install -g particle-cli
C:\Users\USUARIO>npm install -g particle-cli
[.....] - loadRequestedDeps: sill install loadAllDepsIntoIdealTree
```

Figura 56: Actualización de comandos CLI.

Elaboración propia

Para poder actualizar firmware de los dispositivos, tenemos dos opciones vía OTA haciendo clic en “FLASH” Y Vía CLI

Para actualizar firmware vía CLI se ejecutan con dos comandos:

Con el comando “particle flash –serial (modo listen)” //parpadeo azul

Con el comando “particle flash –USB (modo dfu)” //parpadeo amarillo

Por defecto se flashea con el firmware “Tinker” que es un OS básico para que se pueda comunicar con la nube directamente, y podamos verificar dicha comunicación a través de la comunicación de particule.

```
argon.ino
Flash // This #include statement was automatically added by the Particle IDE.
2 #include <ThingSpeak.h>
3
4 // This #include statement was automatically added by the Particle IDE.
5
6
7
8 TCPClient client;//conectar con thingspeak
9 unsigned int myChannelNumber = 184077; // replace with your ChannelID
10 const char * myWriteAPIKey = "39662E74UQB6RK0D"; // replace with your WriteAPIKey
11
12 int tempambiente;
13
14 void setup()
15 {
16     Serial.begin(9600);
17     //pinMode(PinSensor, INPUT);
18     ThingSpeak.begin(client);//iniciamos thingspeak
19     Mesh.subscribe();
20 }
21
22 void loop ()
23 {
24     //ThingSpeak.setField(1,tempambiente);
25     //ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
26     //delay (3000);
27
28 }
```

Figura 57: Actualización de Firmware mediante OTA
Elaboración propia

3.4.2. ALMACENAMIENTO DE DATOS

3.4.2.1 GOOGLE CLOUD

Para el almacenamiento de datos utilizaremos la plataforma Google Cloud con primer paso la Creación de cuenta.

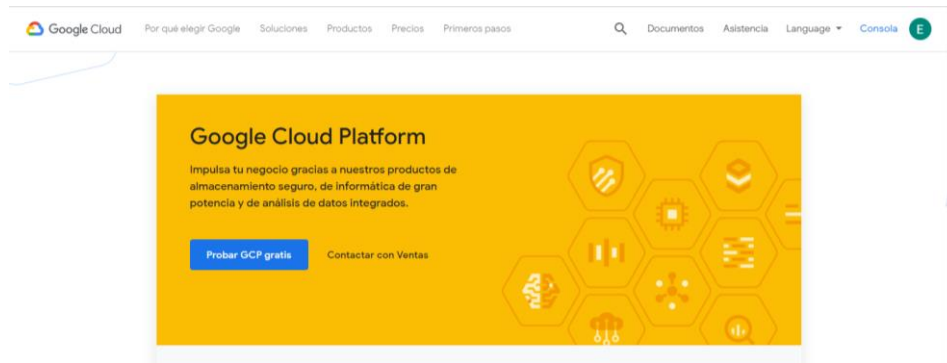


Figura 58: Ingreso a la página de GoogleCloud

Elaboración propia

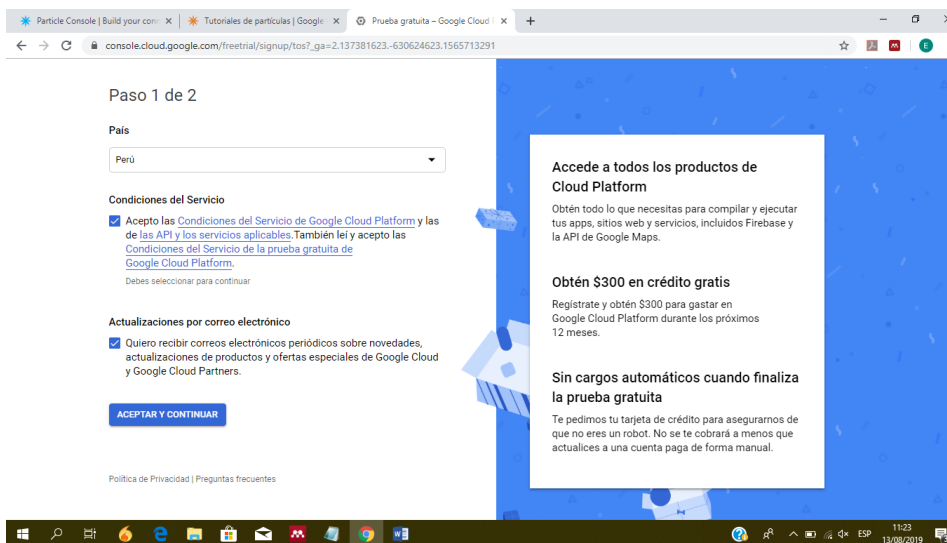


Figura 59: Creación de cuenta en Google Cloud

Elaboración propia

Para la creación de la cuenta gratuita en Google cloud, nos pedirá que insertemos una tarjeta de CREDITO, ya que por temas de seguridad son necesarios los datos de esa tarjeta, teniendo en cuenta que no se hará ningún tipo de cobro, ya que se especifica que se está creando una cuenta gratuita con 300 dólares y con una duración de 12 meses, en mi caso se activó con una tarjeta de crédito de Interbank.

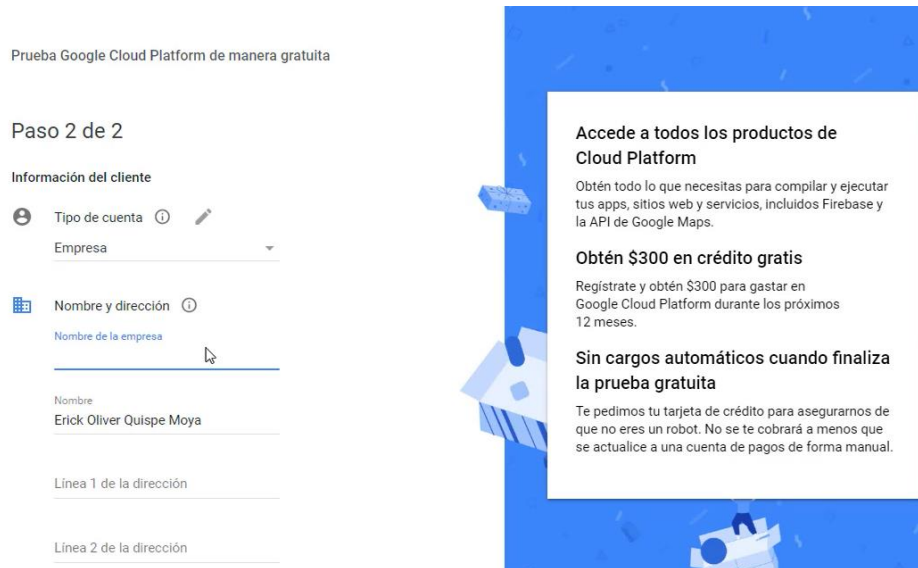


Figura 60: Ingreso de datos de tarjeta
Elaboración propia



Figura 61: Creación de cuenta satisfactoria.
Elaboración propia

Para crear un proyecto hacemos clic en la opción "My First project" no aparecerá un cuadro donde colocamos en nombre de nuestro nuevo proyecto o nuevo TOPIC

Nuevo proyecto

Tienes 13 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. [Más información](#)

[MANAGE QUOTAS](#)

Nombre del proyecto *

ID de proyecto: tesisunap.No se podrá cambiar más tarde. [EDITAR](#)

Ubicación * [EXPLORAR](#)

Organización o carpeta superior

[CREAR](#) [CANCELAR](#)

Figura 62: Creación de nuevo proyecto o Topic
Elaboración propia

Una vez creada el Topic, pasamos un tema en Pub/sub

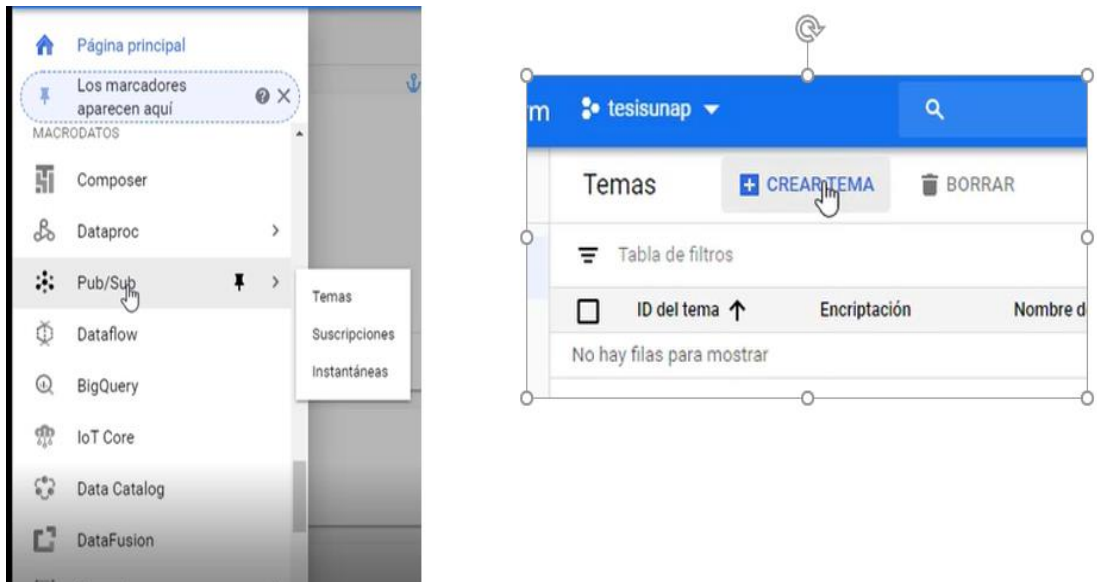


Figura 63: Creación de tema en Pub/Sub
Elaboración propia

Agregamos miembros a tempambiente, pegamos el siguiente link `particle-public@particle-public.iam.gserviceaccount.com` y en función elegimos Publicador de Pub/Sub como lo muestra la Figura N° 64

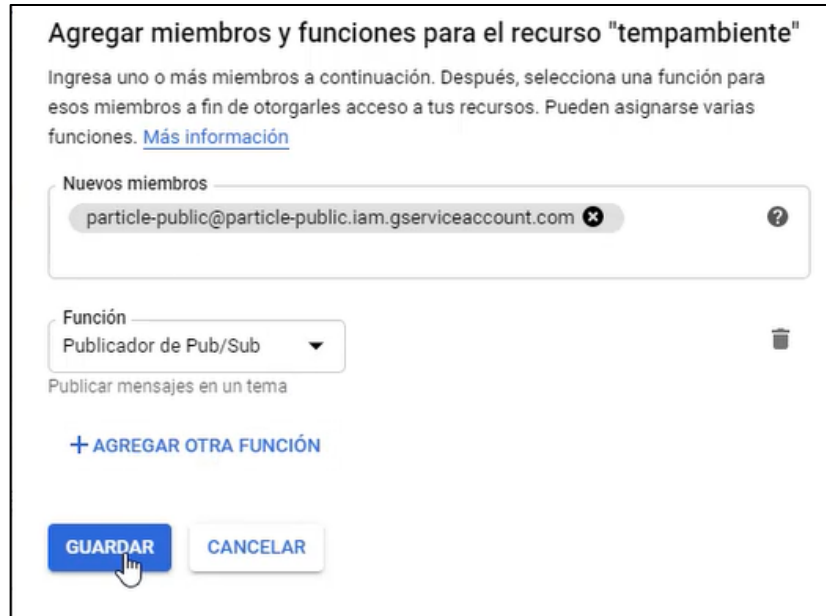


Figura 64: Agregar miembro y función al Tema
Elaboración propia

Luego ingresalos a la consola de Particle para añadir una nueva integración, en este caso se añadira la integración de Google Cloud, danle clic en “+” Como muestra la Figura N° 63.

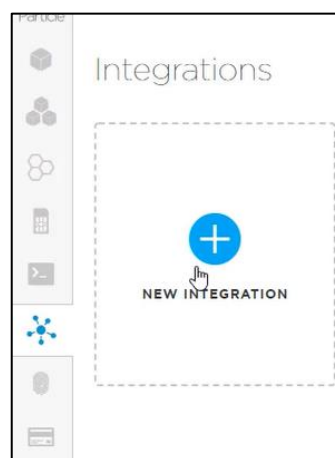


Figura 65: Insertamos una nueva Integración
Elaboración propia

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° 64, recordar que en event name, debe ir el nombre del evento del comando Particle.publish()

Integrations > Edit Integration

Docs Setup

Event Name

Google Cloud Pub/Sub Topic

Device

Figura 66: Creación de Integración
Elaboración propia

En donde crearemos la suscripción, para poder ver nuestros datos con la siguiente línea de comandos (Fig. n° 66):

To create a subscription:

```
~$ gcloud beta pubsub subscriptions create test_sub --topic [topic_name]
```

Where `topic_name` is the name of your Google Cloud Pub/Sub topic (don't include the `projects/{project-id}/topics/` prefix, just the topic name). This subscriber will now listen for all messages published to your topic.

Now, to pull some published messages:

```
~$ gcloud beta pubsub subscriptions pull test_sub --auto-ack --limit 100
```

If things are looking good, you should see a table like this:

Figura 67: Código para suscribirme a mi propio tema
Elaboración propia

Para poder verificar que están llegando los datos, accedemos al CLI de Google Cloud donde tipeamos el código que se muestra en la Figura N° 65

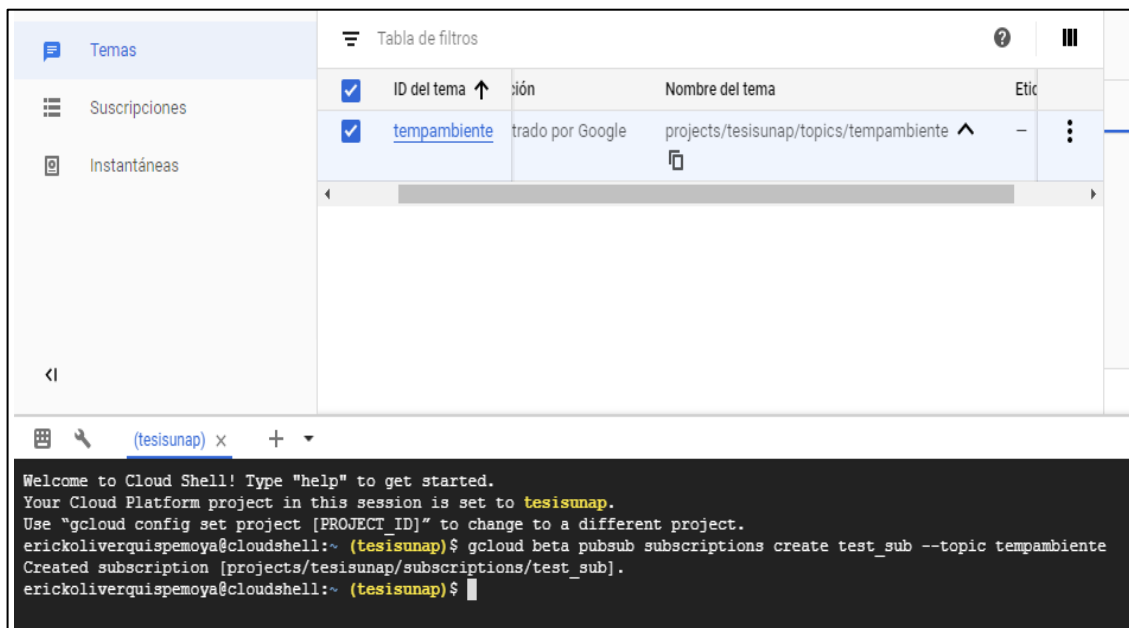


Figura 68: Verificación de llegada de datos
Elaboración propia

Visualización de llegada de datos de Google Cloud mediante CLI en el código de suscripción como se, muestra en la Figura N° 67

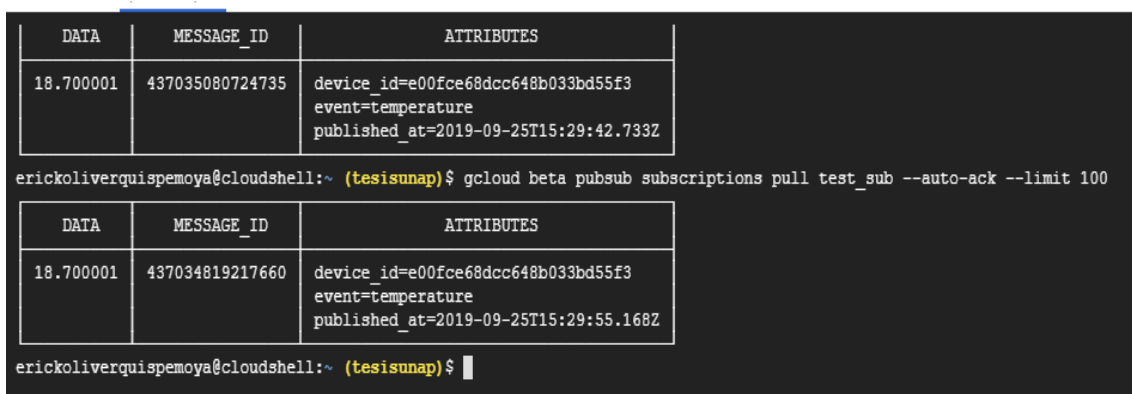


Figura 69: Llegada de datos de Google Cloud
Elaboración propia

Ya con los datos almacenados en Google, le damos respaldo a nuestra información, teniendo en cuenta que tendremos los datos almacenados el tiempo que Google cloud nos brinda con su modo gratuito, 12 meses.

Para pasar los datos a un Datastore ejecutaremos un Script Node.js que actuara como intermediario al suscribirse a un tema de Google Cloud Pub/Sub, con esta ejecución los datos llegaran tanto a Particle Device Cloud como a Google Cloud Pub/Sub, Y lo que script hará es canalizar dichos eventos al Datastore como se muestra en la Figura N° 68.

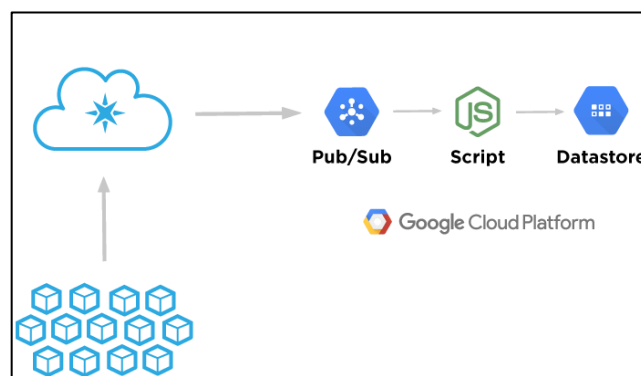


Figura 70: Proceso de almacenamiento en el Datastore

Fuente: Particle (2019)

Por temas de seguridad se creará también una clave de acceso a los temas creados en Google Cloud,

La imagen muestra la interfaz de usuario de Google Cloud Platform en español, específicamente la pantalla 'Crear una clave de cuenta de servicio'. El encabezado muestra 'Google Cloud Platform' y 'My First Project'. El título de la página es 'Crear una clave de cuenta de servicio'. El formulario contiene los siguientes campos:

- Cuenta de servicio:** Un menú desplegable con la opción 'Nueva cuenta de servicio' seleccionada.
- Nombre de la cuenta de servicio:** Un campo de texto con el valor 'nueva cuenta de servicio'.
- Función:** Un menú desplegable con la opción 'Editor' seleccionada.
- ID de cuenta de servicio:** Un campo de texto que muestra 'nueva-cuenta-de-servicio' y '@linear-facet-249716.iam.gserviceacc' con un icono de recarga.
- Tipo de clave:** Sección con el subtítulo 'Descarga un archivo que contiene la clave privada. Almacena el archivo en un lugar seguro, ya que no es posible recuperar la clave si se pierde.'
 - JSON (Recomendada)
 - P12 (Para compatibilidad inversa con código en formato P12)

En la parte inferior del formulario, hay dos botones: 'Crear' (en azul) y 'Cancelar'.

Figura 71: Creación de credenciales
Elaboración propia

El archivo Json permitirá acceder a todos los recursos de la nube, se sugiere guardar en un lugar seguro dicho contenido

```
{
  "type": "service_account",
  "project_id": "linear-facet-249716",
  "private_key_id": "8ef53cc2fbf73c42945d1a25119d91454569cb3a",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvglBADANBgkqhkiG9w0BAQEFAASCBKggSkAgEAAoIBAQCtNvNFYr8ITGXY\nXz4p/5oRTS\nngEwRR2VZr1UoLNNVSAjs2GVkYUWoK3bc6aueXTyAjlInsQCCyP4Kyzbo1o3M2p\n\nwivo1MbbnNyhkUrVnO+hshrYUzygSKXY8/xf2kvFQfxYqebQkTCAS>
  "client_email": "tesis-unap@linear-facet-249716.iam.gserviceaccount.com",
  "client_id": "115949987540986163083",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/tesis-unap%40linear-facet-249716.iam.gserviceaccount.com"
}
```

Figura 72: Archivo Json de Seguridad
Elaboración propia

Lo siguiente que se necesita es una suscripción a Google Cloud Pub / Sub. Recuerde que Particle Device Cloud publica un tema en Google Cloud. Un suscriptor escucha los mensajes publicados en un tema. Necesitaremos un suscriptor para canalizar eventos del tema a la base de datos del almacén de datos.

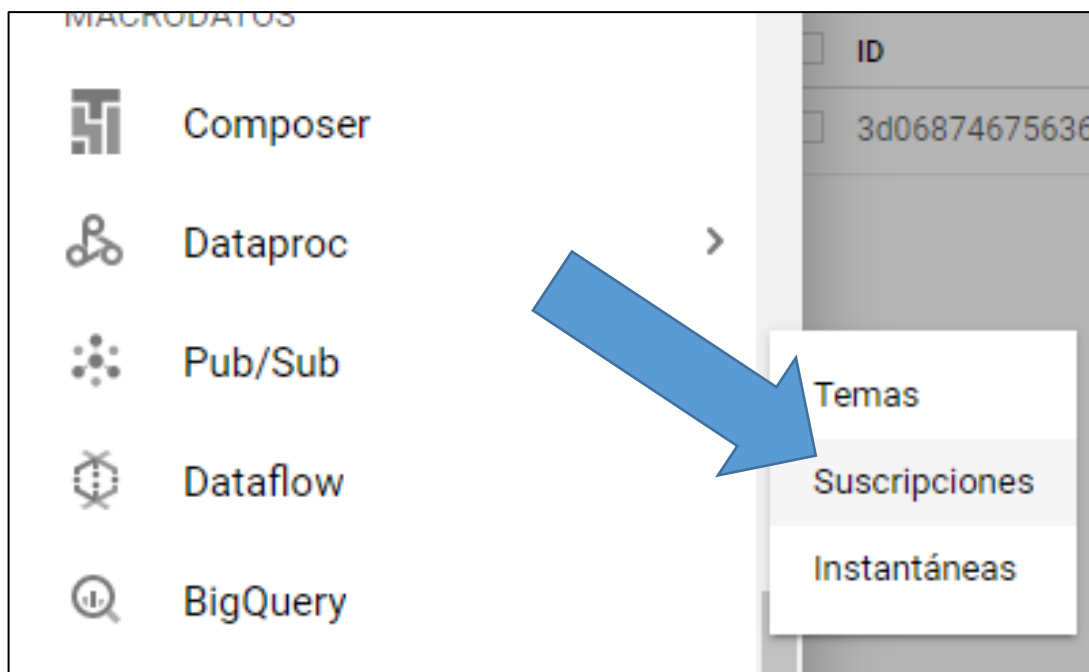


Figura 73: Ingreso Pub/Sub Suscripciones
Elaboración propia

Encuentre su tema en la lista y haga clic en **Nueva suscripción**. Es importante que este nombre de tema coincida con el tema que agregó como parte de habilitar la integración de Google Cloud Plataforma en partículas

← Crear suscripción

Una suscripción permite dirigir mensajes sobre un tema a los suscriptores. Los mensajes pueden enviarse a los suscriptores de forma inmediata, o los suscriptores pueden extraerlos cuando sea necesario.

ID de la suscripción * ?
Nombre de la suscripción: projects/tesisunap/subscriptions/

Nombre del tema *
Tema existente de Cloud Pub/Sub desde el cual suscribirse. El nombre del tema debe tener el formato projects/<project-id>/topics/<topic-id>

Tipo de envío ?
 Extracción
 Envío

Vencimiento de la suscripción ?
 Vence después de muchos días de inactividad (hasta 365)
31 Días
Una suscripción está inactiva si no hay actividad del suscriptor como conexiones abiertas, extracciones activas o inserciones exitosas.
 Nunca se vence

Figura 74: Creación de Suscripción
Elaboración propia

Se colocará de nombre “almacenamientodato” se configura en Tipo de entrega en **Pull**. El nombre de su suscripción se colocará automáticamente como prefijo, como lo hizo su tema.

☰ Tabla de filtros ?

ID de la suscripción ↑	Tipo de envío	Nombre del tema
almacenamientodato	Pull	projects/tesisunap/topics/tempambiente

Figura 75: Creación de suscripción de almacenamiento
Elaboración Propia

Necesitaremos un intermediario para permitir que los datos del dispositivo Particle fluyan desde Google Cloud Pub / Sub al Datastore. Aquí es donde entra en juego el script Node.

Clonación de Repositorio: para la clonación de repositorio se ejecutará el siguiente comando “git clone <https://github.com/particle-IdC/google-cloud-datastore-tutorial.git>” del CMD de Windows

```
Microsoft Windows [Versión 10.0.17763.615]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIO>git
"git" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\USUARIO>
```

Figura 76: Error de Clonación en Windows
Elaboración propia

Para ejecutar el comando de Clonación, se debe descargar el programa Git para Windows de su página principal:

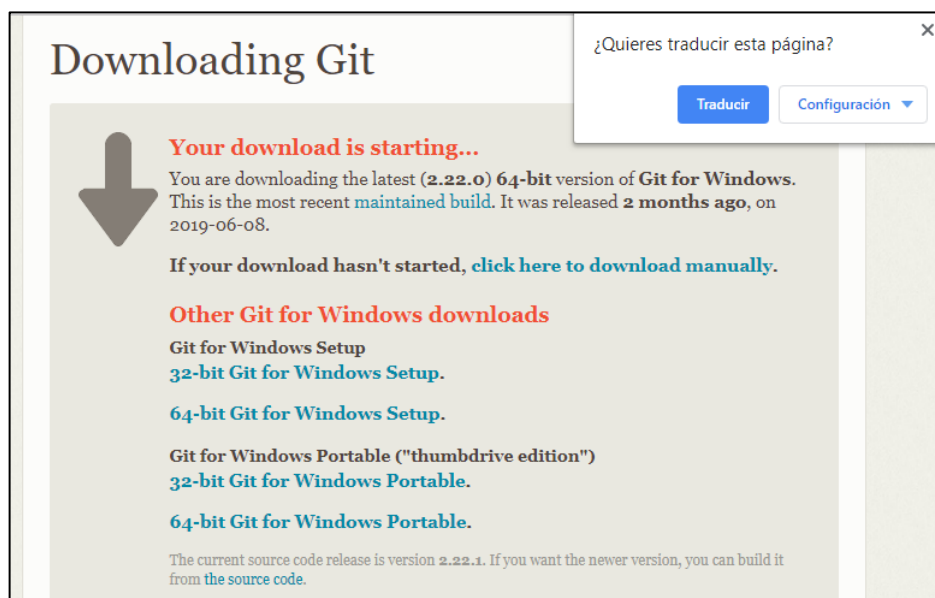


Figura 77: Descarga de Git para Windows
Elaboración propia

Una vez descargada el programa GIT lo ejecutamos y escribimos el código que anteriormente se intentó en con el CMD de Windows


```
MINGW64:/c/Users/USUARIO
rebase      Reapply commits on top of another base tip
tag         Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
fetch      Download objects and refs from another repository
pull       Fetch from and integrate with another repository or a local branch
push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.

USUARIO@DESKTOP-BPJE88U MINGW64 ~
$ git clone https://github.com/particle-iot/google-cloud-datastore-tutorial.git
Cloning into 'google-cloud-datastore-tutorial'...
remote: Enumerating objects: 74, done.
remote: Total 74 (delta 0), reused 0 (delta 0), pack-reused 74
Unpacking objects: 100% (74/74), done.

USUARIO@DESKTOP-BPJE88U MINGW64 ~
$

USUARIO@DESKTOP-BPJE88U MINGW64 ~
$
```

Figura 78: Clonación de Repositorio con GIT
Elaboración propia

3.4.3. PROCESAMIENTO DE DATOS

3.4.3.1. THINGSPEAK

Creamos cuenta en Thingspeak a partir de su página

[“https://thingspeak.com/login”](https://thingspeak.com/login)

Para lo cual necesitaremos una cuenta de Google

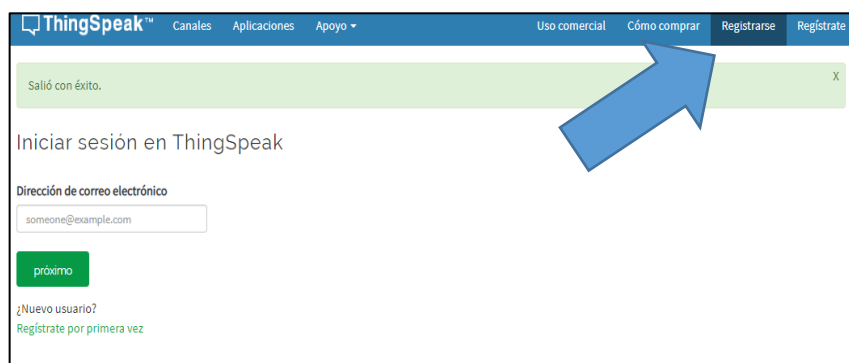


Figura 79: Creación de cuenta Thingspeak
Elaboración propia

Una vez creada la cuenta lo que se tiene que hacer es crear un canal para ellos nos vamos a New Channel, llenamos los datos que indican en la Figura N° 78.

Name	tempambientesuelo	
Description	se va a <u>monitorear</u> los datos de suelo, la temperatura y la humedad	
Field 1	temperatura °C	<input checked="" type="checkbox"/>
Field 2	humedad %	<input checked="" type="checkbox"/>
Field 3		<input type="checkbox"/>
Field 4		<input type="checkbox"/>
Field 5		<input type="checkbox"/>
Field 6		<input type="checkbox"/>
Field 7		<input type="checkbox"/>
Field 8		<input type="checkbox"/>
Metadata		

Figura 80: Creación de Canal y campos
Elaboración propia

Nombre	Creado	Actualizado
tempambientesuelo Privado Público Configuración Compartir API Claves Importar / Exportar datos	2019-09-26	2019-11-28 05:28
Monitoreo Quinua Privado Público Configuración Compartir API Claves Importar / Exportar datos	2019-11-27	2019-11-27 15:10

Figura 81: Visualización de los canales creados
Elaboración propia

Nos dirigimos a la opción de configuración de canal, para visualizar el IDE del canal, recordemos que ese ID tiene que ir en la programación de nuestro dispositivo Argon

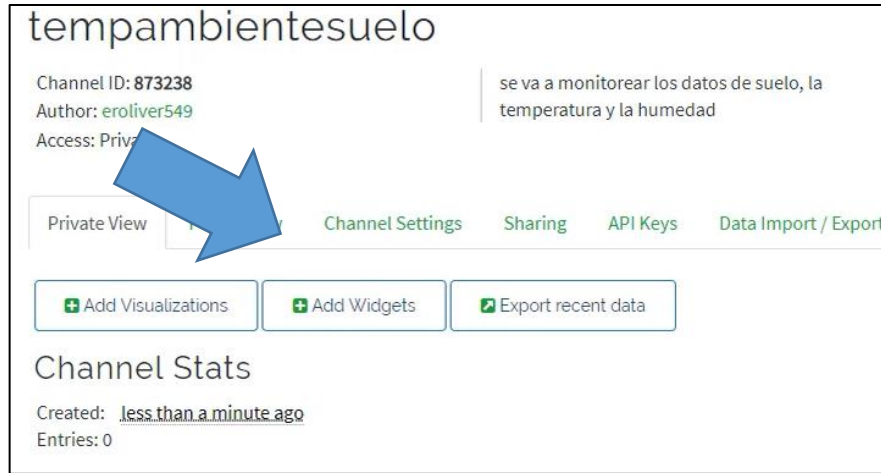


Figura 82: Visualización de ID del canal
Elaboración propia

Luego nos dirigimos a la opción de API Keys para visualizar el Write API Key, teniendo en cuenta también que dicho key se tiene que colocar en la programación del Argon,

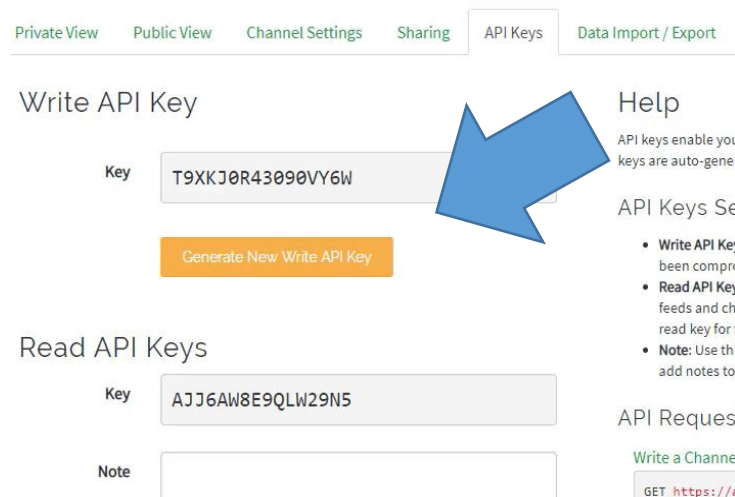


Figura 83: Código API
Elaboración propia

3.4.4. CONSULTA DE DATOS

3.4.4.1 PUSHBULLET

Creación de cuenta de PushBullet a partir de su página principal
“<https://www.pushbullet.com/>” para lo cual solo se necesita tener una cuenta Google.

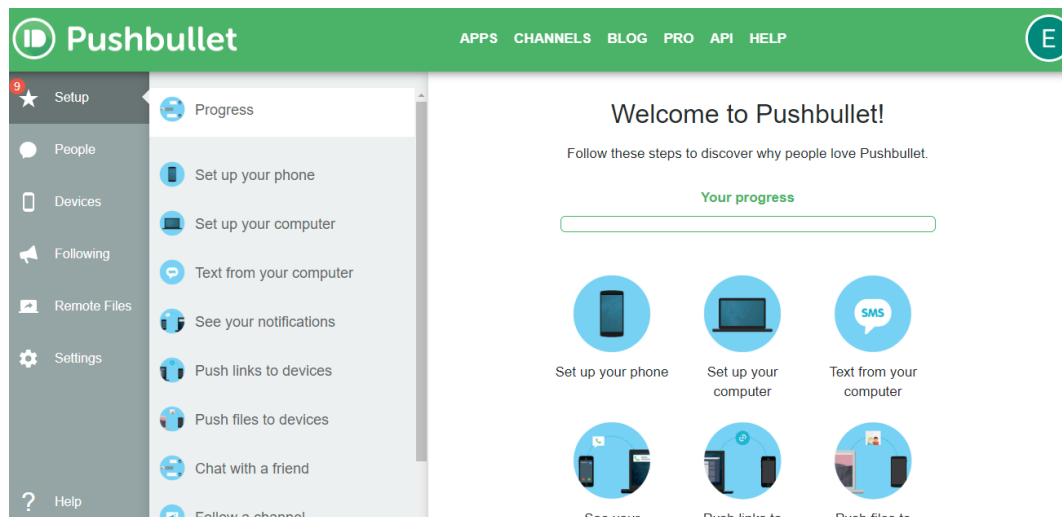


Figura 84: Cuenta creada de PushBullet
Elaboración propia



Figura 85: Creación de nuevo canal
Elaboración propia



Figura 86: Creación de Acces token
Elaboración propia

Luego nos dirigimos a la consola de Particle para crear una nueva integración o WEBHOOK

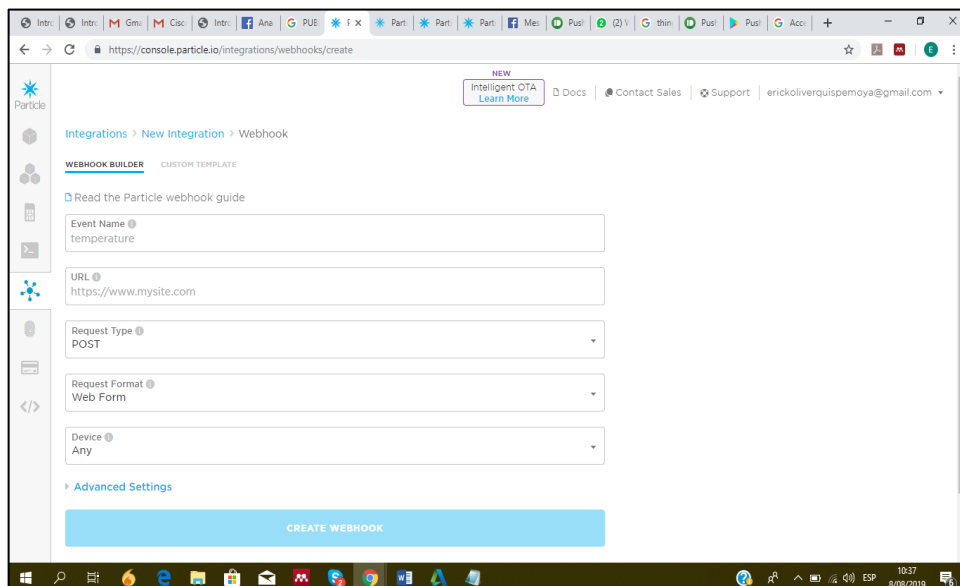


Figura 87: Creación de Webhook
Elaboración propia



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 sincronizarán, se crea un canal de comunicación y en el Webhook colocamos lo siguiente

```
{
  "event": "monitoreopb",
  "url": "https://api.pushbullet.com/v2/pushes",
  "requestType": "POST",
  "noDefaults": true,
  "rejectUnauthorized": true,
  "headers": {
    "Authorization": "Bearer o. PzBBryL51CMUZWoCfVyMr4MDjLj4tiD5",
    "Content-Type": "application/json"
  },
  "json": {
    "event": "{{{PARTICLE_EVENT_NAME}}}",
    "data": "{{{PARTICLE_EVENT_VALUE}}}",
    "coreid": "{{{PARTICLE_DEVICE_ID}}}",
    "published_at": "{{{PARTICLE_PUBLISHED_AT}}}",
    "type": "note",
    "title": "{{{SPARK_EVENT_VALUE}}}",
    "body": ""
  }
}
```

Elaboración propia

Luego nos quedara algo parecido a esto:

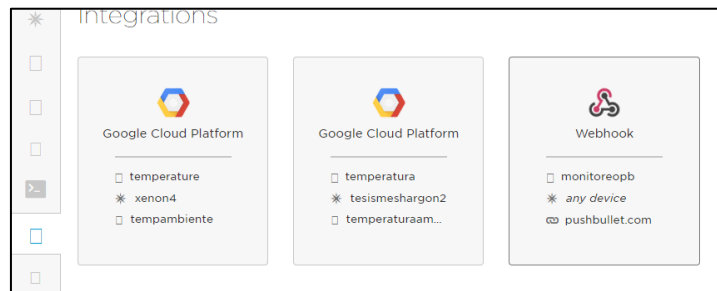


Figura 88: Creación de Webhook
Elaboración propia

3.4.4.2. APP PUSHBULLET

Descargamos la aplicación de Play Store de nuestro Smartphone y le damos permiso de Acceso a tarjeta SD

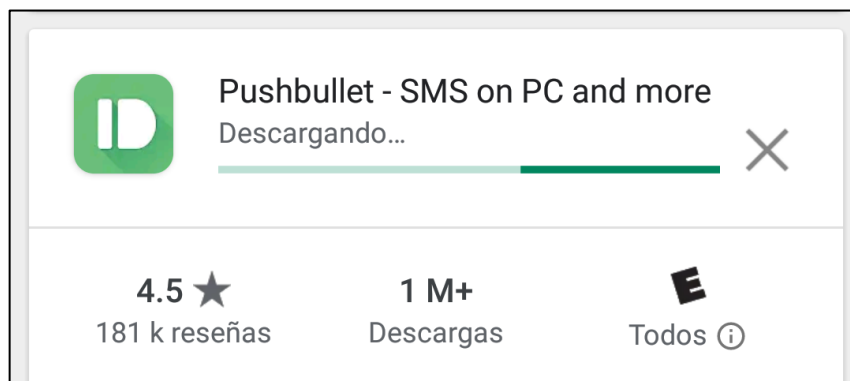


Figura 89: Descarga de PushBullet
Elaboración propia

Se tiene que dar acceso a SD, Contactos y Llamadas telefónicas

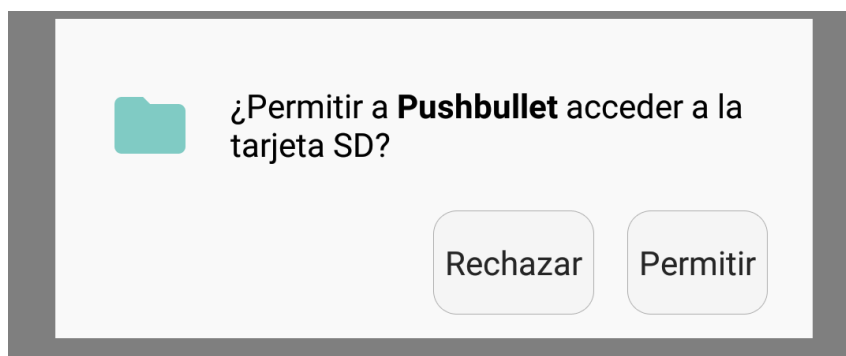


Figura 90: Permiso de Acceso a SD
Elaboración propia

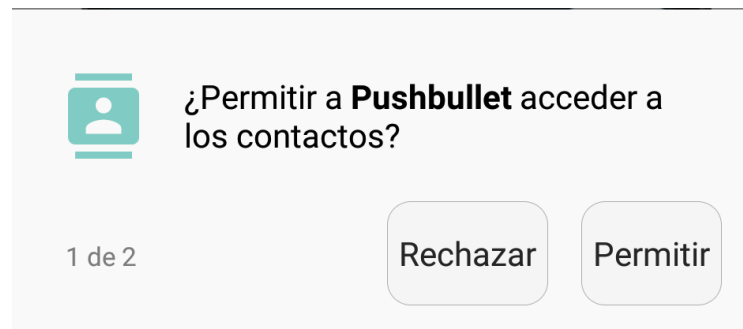


Figura 91: Permiso de Acceso a Contactos
Elaboración propia

3.4.4.3. PLATAFORMA BLYNK.

Para tener acciones remotas, sin necesidad de contacto directo con la agricultura se utiliza esta plataforma para activar y desactivar tanto un sistema de riego como y led o foco de alta potencia.

Es un servicio que nos permite tomar control de nuestro dispositivo a través de internet mediante una interacción directa con el Access Token del mismo servicio, funciona a través de tableros que generan botones virtuales que a través de la programación se pueden ejecutar de forma física con nuestros dispositivos, esto se da a travez de:

```
char auth[] = "_a81xTH2ITOY7NE0gsG42jjXMmeEVZ2g";  
Blynk.begin(auth);  
Blynk.run();
```

Elaboración propia

Y posteriormente se nos notifique cuando el sistema de riego se haya activado con:

```
Blynk.notify (" Se inició el proceso de riego ")
```

De esta forma se puede interactuar de forma remota siempre que lo vea necesario.

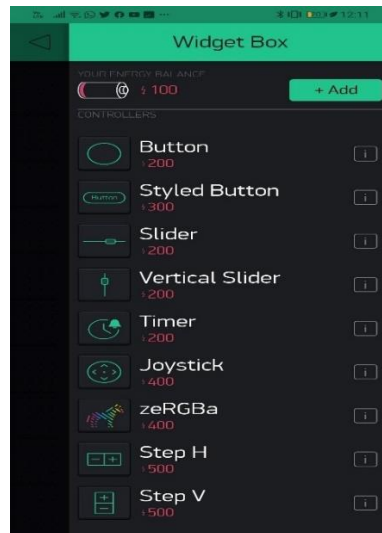


Figura 92: Herramientas de Blynk
Elaboración propia

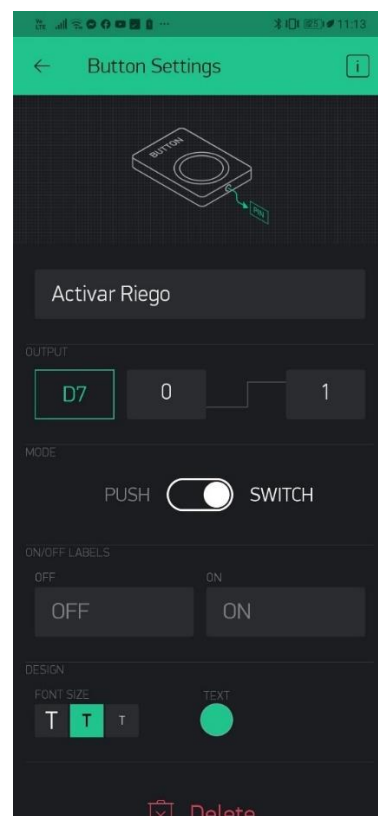


Figura 93: Creación de un botón Virtual.
Elaboración propia

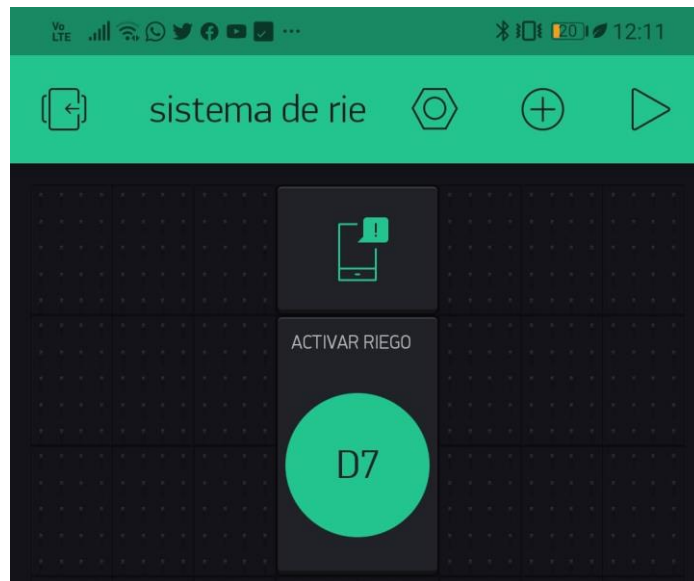


Figura 94: Activación de sistema de riego
Elaboración propia

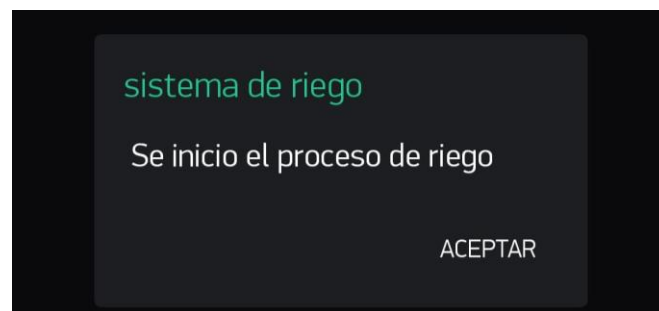


Figura 95: Notificación de riego activado
Elaboración Propia

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. RESULTADOS DE CAPTURA DE DATOS CON LA PLATAFORMA PARTICLE

Se lo logro crear una red Mesh con el dispositivo Argon, Comportándose como Gateway o punto de conexión a internet,

Name	Total device count	Gateway count	Network type
monitoreoquinua	2	0	Micro Network (Wi-Fi)
quinua	3	1	Micro Network (Wi-Fi)

Figura 96: Creación de la red Mesh
Elaboración propia

En la Figura N° 95 se visualiza la creación de la red MESH con nombre “monitoreioquinua” la cual nos servirá para que otros dispositivos Particle se puedan Conectar a ella.

Al estar creada la red Mesh también se añadieron a ella los dispositivos XENON, consolidándose la red Mesh con comunicación inalámbrica.

ID	Type	Name	Last Handshake
e00fce68dcc648b033bd55f3	Argon	tesismeshargon2	11/28/19 at 4:13pm
e00fce68e1fdcdf44d0345a9	Xenon	xenon4	11/27/19 at 10:39pm
e00fce6865eefc5686d30c2b	Xenon	xenon1	11/27/19 at 11:20am
e00fce686c4dc1640c950cdc	Xenon	xenon3	8/25/19 at 9:47pm
e00fce680f503110ce656ab9	Xenon	xenon2	8/25/19 at 9:40pm

Figura 97: Dispositivos conectados a la Red Mesh
Elaboración propia

Se logró la captura de datos de la temperatura- humedad ambiente gracias a la comunicación Mesh entre el Argon y el Xenon (sensor dth22)

Event	humedad	
Data	77.699997	
Time	nov. 26, 2019 21:00:27	
Event	tesisamb	
Data	la temperatura es 11.600000°C	
Time	nov. 26, 2019 21:00:26	
Event	temperatura	
Data	11.600000	
Time	nov. 26, 2019 21:00:21	
Event	tesisamb	
Data	la humedad es 78.000000%se necesita	
Time	nov. 26, 2019 21:00:20	

Figura 98: Captura de datos del sensor DTH22
Elaboración propia

Se logró la captura de datos del sensor de Movimiento

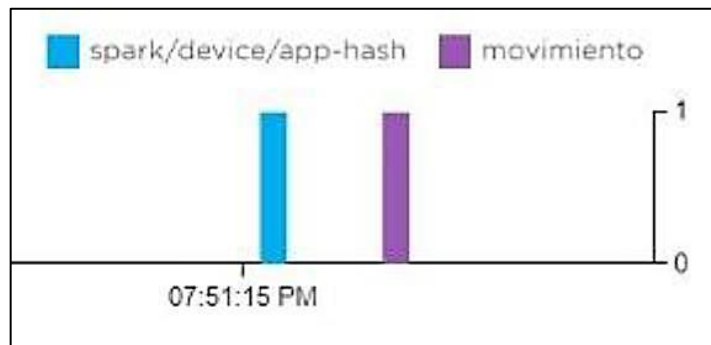


Figura 99: detección de Movimiento con el sensor PIR
Elaboración propia

4.1.1.1. TIEMPO DE RESPUESTA, LATENCIA Y ADQUISICIÓN DE DATOS

Para generar tiempos de latencia y adquisición de datos, Particle tiene la facilidad de darle alcance a toda esa información de datos, para eso ingresamos a la plataforma Particle en en CONSOLA nos dirigimos a “Dispositivos”, como se muestra en la figura 100.

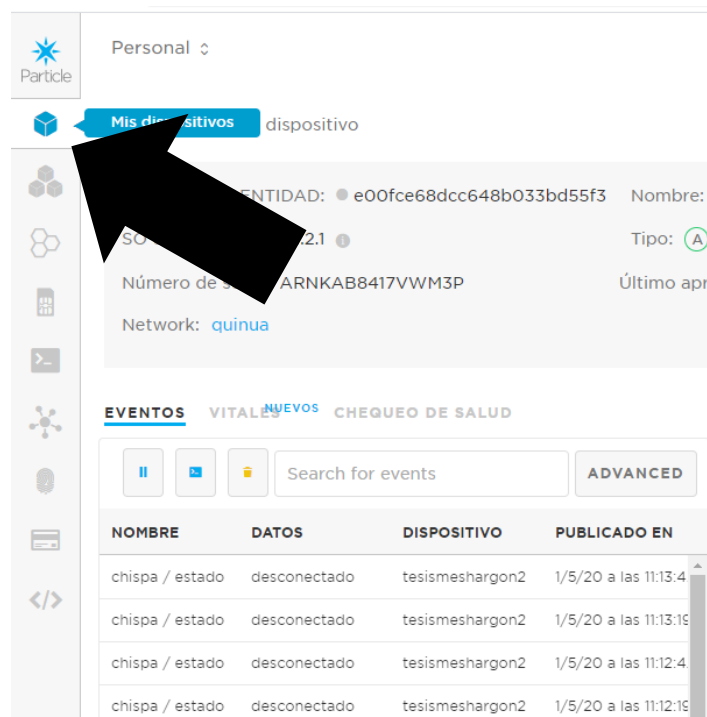


Figura 100: Opciones de "dispositivos" en la consola de Particle

Fuente: Particle (2019)

El máximo tamaño de un arreglo de caracteres (char) que se pueda enviar entre los dispositivos Mesh es de 244 Bytes siendo lo recomendable 236 bytes.

Esto nos da la siguiente velocidad de transferencia máxima por cantidad de Bytes

Tabla 3 Velocidad de transferencia entre dispositivos Mesh,

Tamaño característico	Velocidad máxima de transferencia de datos
20 bytes	222 bytes / seg.
100 bytes	1100 bytes / seg.
200 bytes	2210 bytes / seg.
236 bytes	2186 bytes / seg.
237 bytes	1753 bytes / seg.
244 bytes	1793 bytes / seg.

Fuente: Iot Republic (2018)

En promedio las conectividades BLE tienen una latencia entre 5 a 20 ms, esta puede variar según las condiciones del área de conectividad.

Por otra parte, está la latencia del Gateway que está conformada básicamente por dos aspectos, el primero, por la latencia propia del servidor (ver figura 101), y segundo por el algoritmo y la comunicación del Gateway en sí. Ver Figura 102

Latencia del servidor

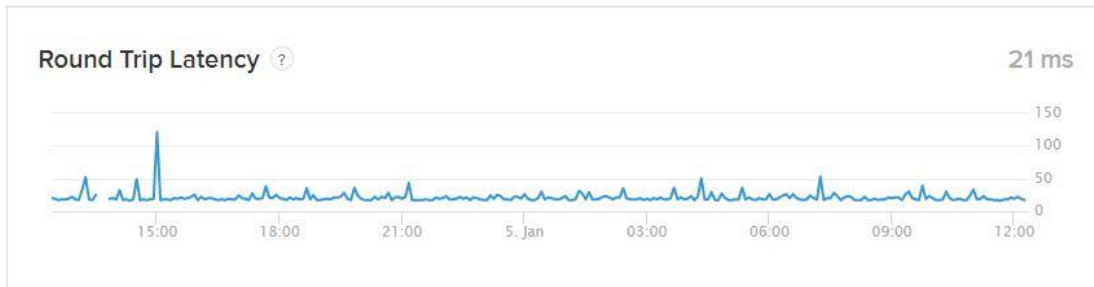


Figura 101: latencia del servidor,

Fuente: Particle (2019)



Figura 102: Latencia del algoritmo y la programación del Gateway.

Elaboración propia

4.1.2. ALMACENAMIENTO DE DATOS CON GOOGLE CLOUD

En la imagen 97 se muestra gracias al código “gcloud beta pubsub subscriptions create test_sub --topic [topic_name]” La llegada de datos de particle a Google Cloud

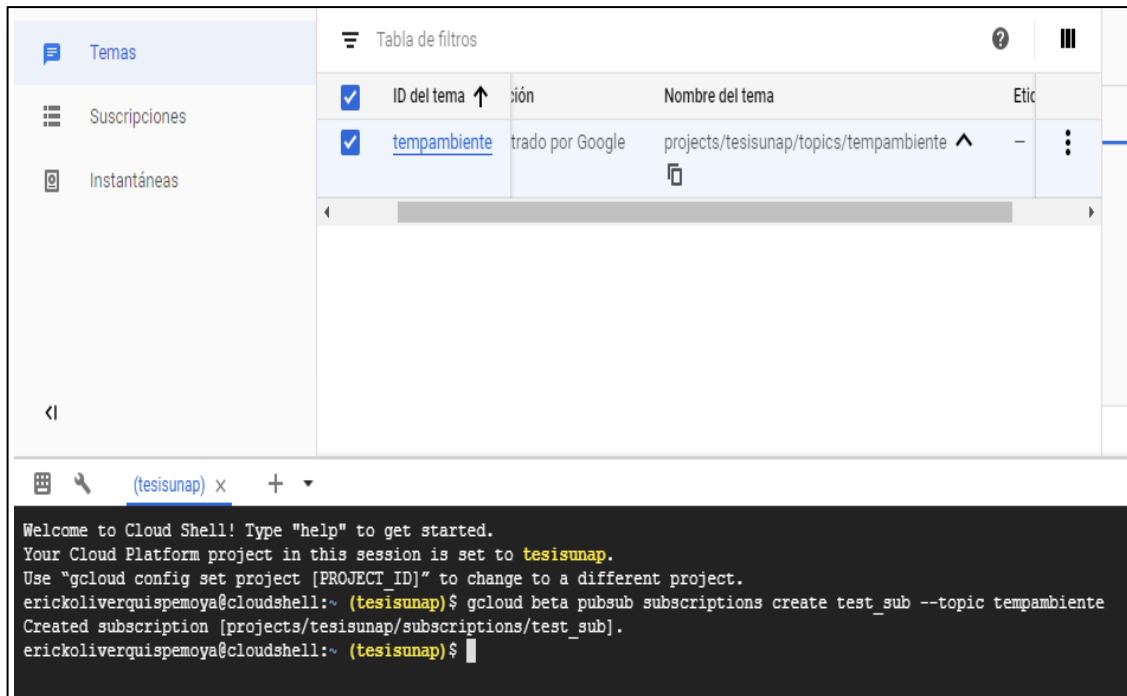


Figura 103: Verificación de llegada de datos

Elaboración propia

Se visualiza la temperatura que el sensor está emitiendo en el momento que tipiamos “gcloud beta pubsub subscriptions pull test_sub --auto-ack --limit 100 En el CLI de Google Cloud”



Figura 104: Llegada de datos de Google Cloud

Elaboración propia

Ya con los datos almacenados en Google, le damos respaldo a nuestra información, teniendo en cuenta que tendremos los datos almacenados el tiempo que Google cloud nos brinda con su modo gratuito, 12 meses.

4.1.3 ANÁLISIS DE DATOS CON THINGSPEAK

Para tener parámetros de cultivos controlados es necesaria la presencia de internet de las cosas, es por eso que la plataforma **ThingsSpeak** es vital para en análisis de los parámetros con las que esta nuestro sembrío de quinua se está cultivando, estos parámetros variaran de acuerdo a los cambios que los sensores capten.



Figura 105: Medición de la Temperatura del suelo
Elaboración propia

Histograma con “Matlap Visualizations”

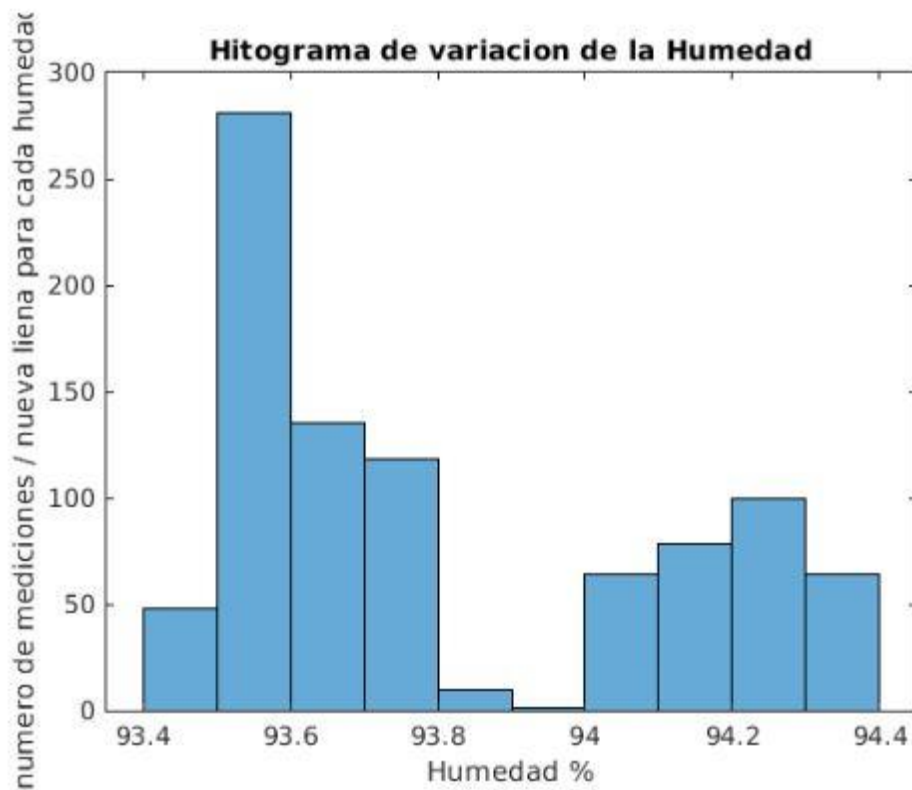


Figura 108: Histograma de la Humedad

Elaboración propia

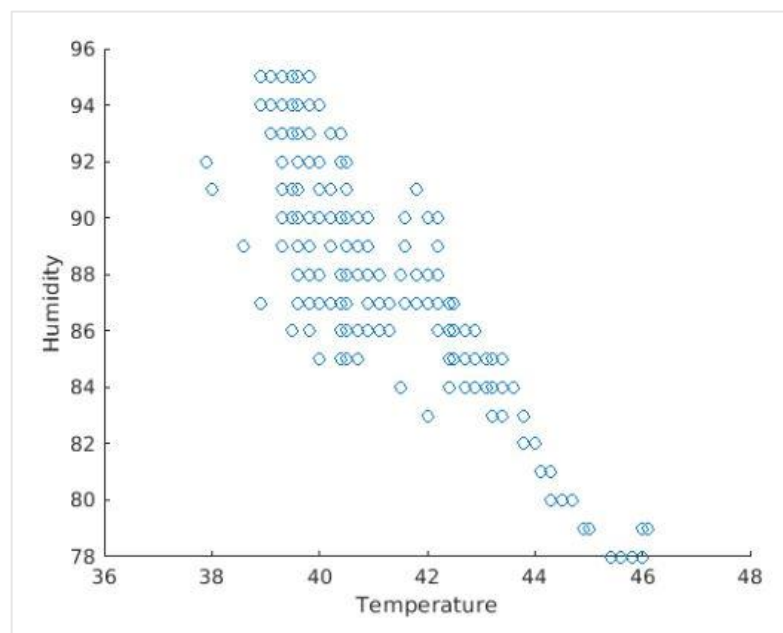


Figura 109: Correlación entre la Humedad y la temperatura ambiente del suelo

Elaboración propia

En la figura N^a 110 se visualiza los pulsos captados por el sensor de moviente graficado en la barra morada y dicho grafico lo realiza el Thingspeak.

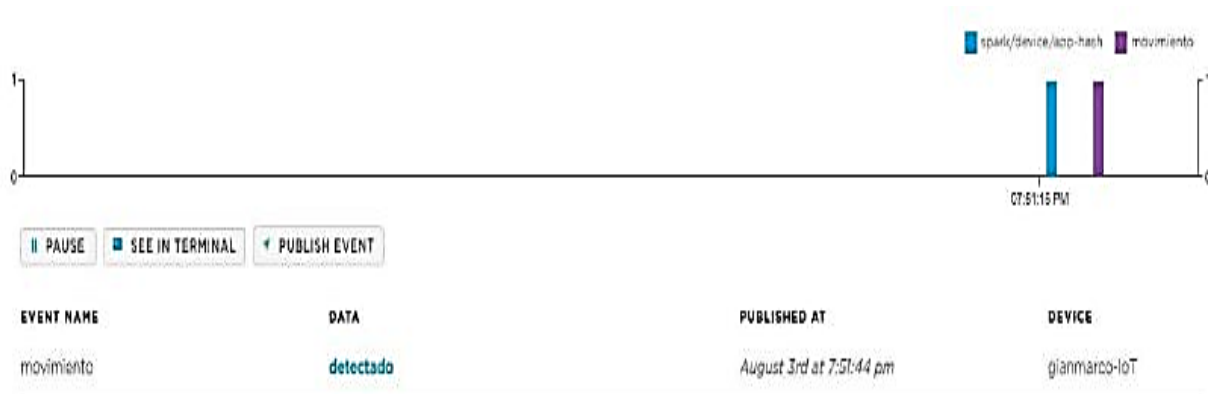


Figura 110: Detección de Movimiento

Elaboración propia

4.1.4. CONSULTA DE DATOS CON PUSHBULLET Y BLYNK

Se logró la recepción de datos de la aplicación PushBullet mediante su app instalado en el dispositivo Smartphone.

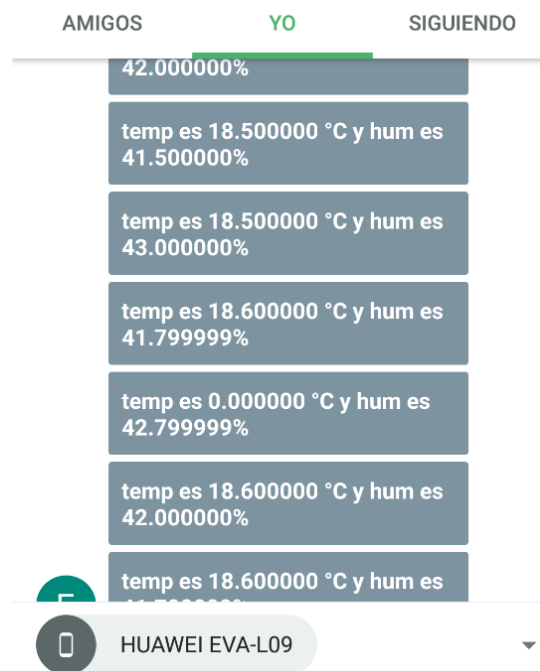


Figura 111: Recepción de temperatura y humedad

Elaboración propia

Se logró la activación de luz con la aplicación Blynk por lo que también se logró la activación de motor para el sistema de riego

Teniendo como Pin de prueba el D7 de nuestro dispositivo Paricle.

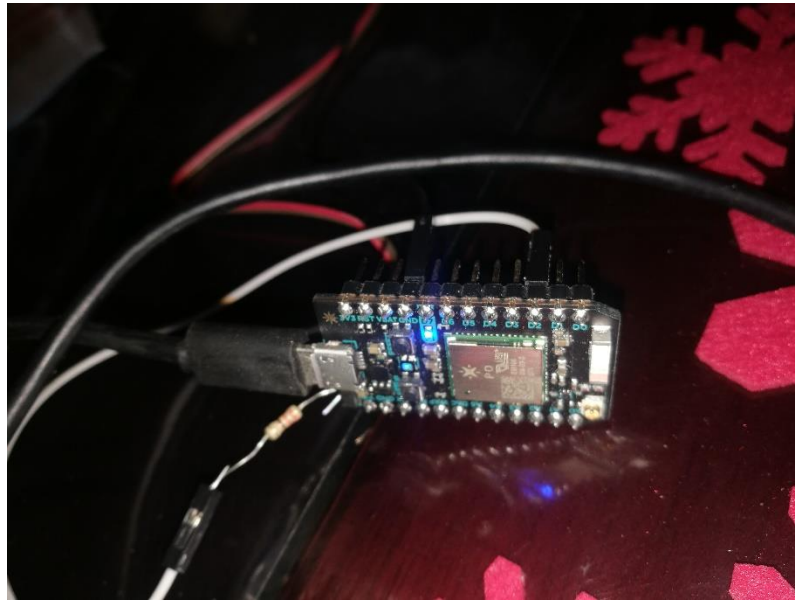


Figura 112: Pin de encendido
Elaboración propia

Mensaje de Blynk de encendido de LED o Motor

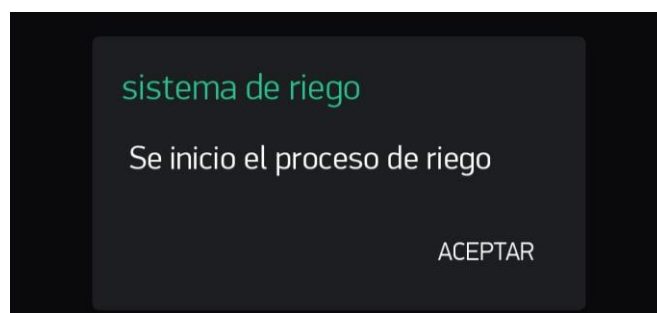


Figura 113: Mensaje de Activación
Elaboración propia

Una vez enviado el mensaje, el LED del Pin D7 encenderá y apagará gracias a la aplicación de Blynk



Figura 114: Led encendido
Elaboración propia

Tabla 4: En la siguiente tabla podemos observar los resultados obtenidos en base a los objetivos planteados donde en algunos casos fueron óptimos y el otros no tan favorables.

Continuará (...)

Hipótesis	Resultados
Es posible mejorar la calidad de los sembríos de la Quinoa a través de la implementación de un sistema de monitoreo, utilizando IoC y redes Mesh	Si es posible debido a que una cosecha será optima siempre en cuando tenga los parámetros correctos de cultivo, y (como se muestra en la figura 98), esto apoyado de la agricultura de precisión, método que va de la mano con la tecnología, por lo cual si nosotros somos capaces de mantener estables dichos parámetros estaremos mejorando la calidad de los sembríos



<p>Es conveniente el uso de un sistema basado en una red Mesh e internet de las cosas, para mejorar la calidad de los sembríos de quinua</p>	<p>Si es conveniente, según las necesidades ya que en mi caso la mejor opción fue el uso de una red Mesh, a través de la tecnología BLE, ya que estos demandan bajo costo, bajo consumo energético y fácil escalabilidad que son idóneos para cultivos de producción media que se tiene en la Región, sin embargo, para cultivos mucho más grandes y con mayor costo de producción, seguiría siendo una alternativa fuerte pero competiría con otras soluciones (LORA, WIFIDIRECT, etc.)</p>
<p>Es posible Implementar un sistema de monitoreo utilizando IoC y redes Mesh, para mejorar la calidad de producción en los sembríos de Quinua</p>	<p>Si es posible, debido a que actualmente el uso de software y hardware libre crea una comunidad de soporte muy grande lo cual permite tener acceso a muchas soluciones, en el presente proyecto se usó una solución basada en la facilidad de uso y comunicación, sin embargo existen muchas otras formas de afrontar estos retos de implementación que con el tiempo se están haciendo visibles.</p>
<p>Es posible Monitorizar los cultivos para mejorar la calidad de producción de la Quinua.</p>	<p>Si es posible monitorizar, ya que las herramientas actuales nos proporcionan capacidades que hasta pocos años no se</p>



	tenían y que requerían de codificaciones complicadas.
Es posible lograr la comunicación ente la agricultura y la Tecnología	Si es posible lograr una comunicación de este tipo, lo que actualmente se le denomina “Industria 4.0” es decir, donde la industria está en un marco de referencia guiadas por las Tecnologías de la Información donde la conectividad en tiempo real y el diagnostico remoto nos ayudaran a predecir más que a reaccionar, lo cual da una ventaja competitiva frente a las anteriores revoluciones.

Elaboración propia

4.2. DISCUSIÓN.

La mayoría de antecedentes de monitoreo de cultivos de quinua están basadas en el empirismo, lo cual no nos da con exactitud datos que nos puedan ayudar a prevenir perdidas en los cultivos, por lo cual implementar de forma tecnológica nos da una precisión que puede eliminar y evitar estos problemas en los cultivos de Quinua y otros productos.

Implementar y diseñar un monitoreo basándose en la arquitectura Lambda la cual consta de cuatro componentes a saber: Captura de Datos, capa de Velocidad o Procesamiento, la Capa de Almacenamiento o Seguridad y la Capa de Consulta como lo plantea en su artículo de investigación (Quiroga et al., 2017),

Si es posible implementar estas 4 capas de la arquitectura Lambda ya que funciona tal y como se muestra en la parte de resultados, para la captura de datos se visualizan en las

Fig. 95, Fig. 96, Fig. 97 y Fig. 98, para la capa de Almacenamiento o seguridad se visualizan en las Figuras Fig. 99 y Fig. 100, para la capa de análisis de datos de visualiza en las Fig. 101, Fig. 102, Fig. 103, Fig. 104, Fig. 105 y Fig. 106 para la capa de consulta de datos se visualiza en las Fig. 107 y Fig. 109.

Compara la tesis **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL MONITOREO DE CULTIVOS NATIVOS UTILIZANDO INTERNET DEL TODO Y REDES FOG** de (Palacios, 2017), en mi caso fue más idóneo la implementación de una red Mesh que una red fog, puesto que la arquitectura de una red MESH es más sostenible en lugares donde las condiciones económicas y energéticas son escasas

En la investigación “UBICACIÓN DE RECURSOS DE LA PLATAFORMA DE ANALISIS DE DATOS EN FOG COMPUTING, se puede constatar que la implementación del FOG COMPUTING ayuda muchísimo en el procesamiento de la información, sin embargo, su infraestructura podría no ser adecuado para ciertas áreas, donde las redes Mesh tienen un mejor desempeño

De acuerdo con En la Investigación “SgreemH-Idc: Plataforma Idc para la agricultura de precisión” en esta investigación se presenta SgreemH-Idc una plataforma Idc a bajo costo y consumo energético, tercerizar el uso de servicios nos facilita la implementación de este tipo de proyectos, sin embargo para etapas mucho más grandes, es necesario empezar a hacer estos servicios propios, y formar tener un mayor control de nuestro sistema, y no depender de terceros, por lo que en esta fase de desarrollo el soporte que brindan estos servicios de terceros ayuda a poder implementar con seguridad dichos sistemas.

En la investigación Nuevo Protocolo Thread 1.2 de IdC y Hogares Inteligentes Habla de la versión de protocolo de comunicación inalámbrica “thread 1.2”, este protocolo está



siendo adoptado por empresas, donde una de ellas es Particle dando así la posibilidad de que más adelante este tipo de comunicación ya no se hagan de forma exclusiva.

De acuerdo con Quiroga, “Propuesta de una Arquitectura para Agricultura de Precisión Soportada en IoT”, el software y hardware libre se desarrolla cada vez más dando la opción de nuevos módulos, protocolos, etc. Que cada vez son más accesibles y económicos atractivos para la implementación en todo tipo de industrias, por lo cual, así como se puede implementar en la agricultura también se ve el desarrollo a nivel global.

4.2.1. DISEÑO DE MONITOREO ECONÓMICO Y SOSTENIBLE CON REDES MESH.

En el pasado, construir e implementar más allá de una conexión de un solo punto de red, requería un alto costo de hardware y software. A medida que las plataformas IdC han madurado, han comenzado a adoptar una alternativa de bajo costo y baja potencia que puede cerrar las brechas de su uso: **las redes de malla inalámbricas.**

4.2.2. PORQUE UTILIZAR UNA RED MESH INALÁMBRICA PARA IDC.

Si bien las tecnologías de redes de malla inalámbrica han existido durante algún tiempo, solo recientemente el poder de la malla ha alcanzado un punto de madurez junto con la alta disponibilidad de los proveedores de chips y silicio. Con nuevos costos accesibles, la red de malla inalámbrica se ha convertido en ideal para los desarrolladores de IdC. Y con el aumento de los hogares conectados y el apoyo de la industria en recursos de código abierto como Thread, Mesh ahora es verdaderamente accesible y tiene un costo lo suficientemente bajo como para escalar para la producción. Como tal, la red de malla inalámbrica se está convirtiendo en una opción mucho más viable para aplicaciones industriales y comerciales.

Tabla 5: Presupuesto Mínimo para la implementación del sistema de Monitoreo

Descripción	Unidad	Costo Unitario (s/.)	Cantidad	Costo Total
Argon	Kits	94.78	01	94.78
Xenon	Kits	71.08	01	71.08
Sensores de temperatura humedad ambiente	01	25.00	01	25.00
Sensor de Movimiento PIR	01	7.00	01	7.00
Sensores de temperatura – humedad del suelo	01	155.40	01	155.40
Power Bank	01	36.00	01	36.00
Modem Wi-fi	Mb/mes	39.00	01	39.00
Otros Accesorios		100.00		100.00
TOTAL				528.26

Fuente: Propia

4.2.3. INCONVENIENTES CON EL USO DEL ARGON.

Si bien la implementación de una red Mesh Inalámbrica es muy económica por el ahorro de batería y la adquisición de los dispositivos Particle, se tiene que tener en cuenta la Ubicación en donde se quiera aplicar este proyecto, siendo un problema zonas Urbanas que no cuentan con tecnología 3G o 4G, en caso único la Red Mesh será implementada

en Afueras de ciudad de Juliaca donde esta zona ya cuenta con conexión 4G (ver Figura N° 109)

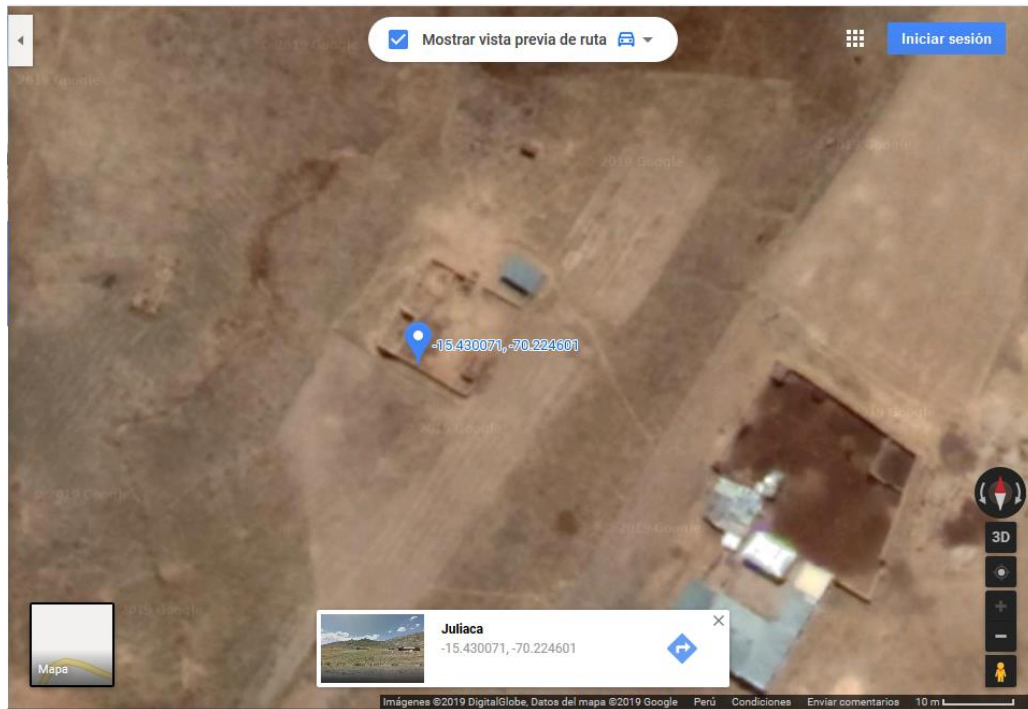


Figura 115 Ubicación de aplicación de Monitoreo

Elaboración propia

4.2.4. CONVERGENCIA

Hablar de convergencia en la tecnología nos lleva a la innovación de la misma, así como lo hace la Plataforma Particle, lanzando productos y/o dispositivos que ya vienen incorporados con toda la tecnología necesaria, en caso de este proyecto se unifican diversos servicios los cuales son:

- Uso de dispositivos Particle, que vienen con tecnología todo en uno para el ahorro de adquisición de más dispositivos.
- Uso de la programación, como medio para salir de la red hacia Internet.
- Uso de las API, como herramienta para integrar 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 servicios de grandes empresas como Google Cloud o Microsoft Azure, que nos permiten mejorar y dar escalabilidad a nuestro sistema.



V. CONCLUSIONES

PRIMERO: Se puso diseñar e implementar sistema de monitoreo de cultivos de Quinoa, basado en una red Mesh a bajo costo, utilizando tecnología BLE, ya que esto nos permite implementar una infraestructura económica a comparación de otras soluciones basadas en la misma topología, puesto que el consumo energético y el costo económico de Bluetooth low energi (BLE) es mucho menor, y también el uso de IdC como una solución, ya que todo este procesamiento de datos se converge en la Plataforma Particle y de esa forma poder hacer integraciones con terceros de forma práctica y fiable como lo son Thingspeak, Google Cloud, PushBullet y Blynk entre otros.

SEGUNDO: Se logró el diseño de una red Mesh utilizando IdC, mediante un proceso de reconocimiento de los pilares que cada elemento conforma el concepto de IdC, y de esta forma se crea la solución en la cual los objetos, datos, personas y procesos logran interactuar de forma eficiente, para así lograr una agricultura de Precisión, y de esta forma controlar los parámetros de cultivos de Quinoa logrando un mejor producto y a la vez generando datos que nos permitan entender el comportamiento de los cultivos, ya con esos datos, analizarlos y prevenirlos de amenazas que puedan afectar a los mismos, por consiguiente también tener una mejora constante de producción.

TERCERO: Se logró la implementación del sistema de monitoreo utilizando una red Mesh e IdC, con los dispositivos Argon y Xenón, comunicándose a través de una Red de malla Inalámbrica (WMN) de forma segura, ya que estos dispositivos están basados en OPENTHREAD lo que nos permite una comunicación segura y encriptada bajo el protocolo IPV6 (802.15.4-2016) que está hecha para Internet de las Cosas de bajo consumo energético y a la vez en el estándar lanzado por NEST que es una marca de Google que nos permite conectar las cosas, como infraestructura está basado en software



y hardware libre lo que nos permite desarrollar de forma abierta los códigos para obtener una escalabilidad sencilla

CUARTO: Es posible el monitorio de un cultivo de quinua a través del diseño e implementación de una red Mesh utilizando IoC, gracias a las integraciones que nos permite el desarrollo de software a través de métodos como: Pub/Sub, Webhooks y APIs; ya que estos métodos son independientes del tipo de hardware que se esté utilizando y están directamente relacionados con los servicios de terceros como Google, PushBullet, Thingspeak, Blynk y muchos otros más, dándonos la posibilidad de siempre tener la alternativa más adecuada a nuestras necesidades.

QUINTO: Se logró la comunicación ente la agricultura y el usuario final con la ayuda de la tecnología, a través de aplicaciones de terceros mediante notificaciones, mensajes Push, email, Visualización de página Web, y de esta forma también tener conocimiento y poder manipular nuestros procesos de forma remota.

VI. RECOMENDACIONES

PRIMERO: En caso de la conexión wifi, se recomienda utilizar un modem con línea Bitel, o Entel, ya que estas líneas tienen amplia cobertura en la zona donde se aplicará el proyecto, no siendo necesario utilizar conexión TCP ya que el Wifi será ilimitado.

SEGUNDO: Teniendo en cuenta que la aplicación de este proyecto se hizo en una zona con cobertura 4g, no se tuvo inconvenientes en cuanto a la comunicación del Gateway(argón) con la nube de internet, en caso de que el proyecto se aplique a zonas donde no cuenten con tecnología 4g, se recomienda hacer un cambio de dispositivo de particle, es recomendable usar el dispositivo Boro que tiene la tecnología (ver Tabla N° 44

Tabla 6 Enfoque para selección de rol de dispositivos

Tableros	Wifi	Celular	Malla
Argon	x		x
Xenon			x
Boro		x	x

Elaboración propia

TERCERO: Para el almacenamiento de datos recibidos de Particle en Google Cloud, se recomienda descargar el programa “GIT” ya que Windows no es compatible con el comando de Clonación de Repositorio (Ver fig. 75).



CUARTO: Tener en cuenta la actualización de esta tecnología, ya que mucho antes no era para nada rentable, por el tema de la infraestructura el costeo y la escalabilidad, pero con la actualización de su firmware y la creación de nuevos protocolos como el Thread hacen que la tecnología Mesh sea la más asequible a comparación de otras redes.



VII. REFERENCIAS

- Alandia, S., Otazu, V., & Salas, B. (1979). *Enfermedades. En: Quinoa y Kañiwa Cultivos Andinos* (IICA).
- Antevenio. (2019). *¿Qué es el internet de las cosas? todo lo que necesitas saber.*
<https://www.antevenio.com/blog/2019/09/que-es-el-internet-de-las-cosas/>
- Bonifacio, A., & Saravia, R. (1999). *Evaluación de la resistencia al mildiu en quinua. In Tercer Taller de Preduza en Resistencia Duradera en Cultivos Altos en la Zona Andina.*
- Danielsen, S., & Ames, T. (2000). *El mildiu (Peronospora farinosa) de la quinua (Chenopodium quinoa) en la zona andina* (Centro Int).
- El Peruano. (2017). *Agricultura de Precisión: desafío de los productores.*
<https://elperuano.pe/noticia-agricultura-precision-desafio-de-productores-60185.aspx>
- FAO, & CIRAD. (2013). *State of the Art Report of Quinoa in the World in 2013.*
- Fisher, T. (2019). *¿Qué es una red de malla?* <https://www.lifewire.com/mesh-network-4175202>
- Fundación de la Innovación Bankinter. (2011). *El Internet de las Cosas - En un mundo conectado de objetos inteligentes.*
http://www.belt.es/expertos/%0Aimagenes/XV_FTF_El_internet_de_las_cosas.pdf
- Gomez, P. L., & Aguilar, C. E. (2016). *Guia De Cultivo De Quinoa. In Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering, JCSSE 2017* (Universida).
<https://doi.org/10.1109/JCSSE.2017.8025923>



- Guerrero, I. J. A., Estrada, G. F. P., Medina, T. M. A., Rivera, G. M. G., Alcaraz, A. J. M., Maldonado, M. C. A., Toledo, Z. D., & Lopez, G. V. I. (2017). SGreenH-IoT: Plataforma IoT para Agricultura de Precisión. *Sistemas, Cibernética e Inform{á}tica*, 53–58.
[http://www.iiisci.org/journal/CV%7B\\$%7D/risci/pdfs/CA544SI17.pdf](http://www.iiisci.org/journal/CV%7B$%7D/risci/pdfs/CA544SI17.pdf)
- Hyea-Won, L., & Vikas-Choudhary. (2017). *Agricultura 2.0: cómo la “Internet de las cosas” puede revolucionar el sector agrícola*.
<https://blogs.worldbank.org/es/voices/agricultura-como-internet-puede-revolucionar-el-sector-agricola>
- Ibañez, J. (2009). *Agricultura de Precisión y Suelos*.
- Iot Republic. (2018). *Particle*. <https://www.iotrepublic.mx/articles/particle-una-plataforma-para-el-desarrollo-de-iot>
- Lee, J. (2018). *Cómo construir una red de malla inalámbrica*.
<https://blog.particle.io/2018/04/28/how-to-build-a-wireless-mesh-network/>
- León-lobos, P., Morales, A., Ruf, K., Zurita, A., & Alfaro, C. (2018). *Catálogo de Variedades Locales de Quínoa*.
- Llamas, L. (2019). *Protocolos de comunicación para IoT*.
<https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>
- Manzo, J. (2017). *API vs Webhooks*. <https://medium.com/@jdevmanzo/api-vs-webhooks-4745bffcfa65>
- Palacios, F. G. (2017). DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL MONITOREO DE CULTIVOS NATIVOS UTILIZANDO INTERNET DEL TODO Y REDES FOG. *Repositorio Institucional UNA-PUNO*.



- Particle. (2019). *Partícula de xenón*. <https://docs.particle.io/xenon/>
- Particle. (2019). *ParticlaPARTICLE ARGON: WI-FI + BLUETOOTH + MESH*.
<https://docs.particle.io/argon/>
- Peña, M. (2019). *Qué es el Internet de las Cosas y cómo afecta tu vida diaria* .
<https://es.digitaltrends.com/tendencias/que-es-el-internet-de-las-cosas/>
- Quiroga, M. E. A., Jaramillo, C. S. F., Campo, M. W. Y., & Chanchí, G. G. E. (2017).
Propuesta de una Arquitectura para Agricultura de Precisión Soportada en IoT.
RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao, 24, 18.
<https://doi.org/10.17013/risti.24.39-56>
- Ramirez, I. (2019). *Aplicaciones con las que flipaste en su día: Pushbullet*.
<https://www.xatakandroid.com/aplicaciones-android/aplicaciones-que-flipaste-su-dia-pushbullet>
- RS Components. (2016). 11 redes inalámbricas fundamentales para Internet de las Cosas.
737, 62–65. [http://www.sherpa.ac.uk/romeo/issn/0482-6396/%0Ahttps://polibuscador.upv.es/primo-explore/openurl?url_ver=Z39.88-2004&rft_val_fmt=info:ofi%2Ffmt:kev:mtx:journal&rft.date=2016&rft.issue=737&rft.epage=65&rft.genre=journal&rft.jtitle=Revista española de elec](http://www.sherpa.ac.uk/romeo/issn/0482-6396/%0Ahttps://polibuscador.upv.es/primo-explore/openurl?url_ver=Z39.88-2004&rft_val_fmt=info:ofi%2Ffmt:kev:mtx:journal&rft.date=2016&rft.issue=737&rft.epage=65&rft.genre=journal&rft.jtitle=Revista%20espa%C3%B1ola%20de%20elecc)
- SAS THE Power to Know. (2017). *Online guide to the Internet of Things*. 27.
<http://www.i-scoop.eu/internet-of-things/>
- Schutter, L. (2018). *9 cosas que debe saber sobre las redes de malla: por*.
<https://hackernoon.com/9-things-you-need-to-know-about-mesh-networks-f61a77e5751a>
- Secureweek. (2019). *Nuevo Protocolo Thread 1.2 de IoT y Hogares Inteligentes*.



<https://www.secureweek.com/nuevo-protocolo-thread-1-2-de-iot-y-hogares-inteligentes/>

Tam, M. J. (2008). *Tipos, métodos y estrategias de investigación científica*.

Taneja, M., & Davy, A. (2016). Resource Aware Placement of Data Analytics Platform in Fog Computing. *Procedia Computer Science*, 97, 153–156.
<https://doi.org/10.1016/j.procs.2016.08.295>

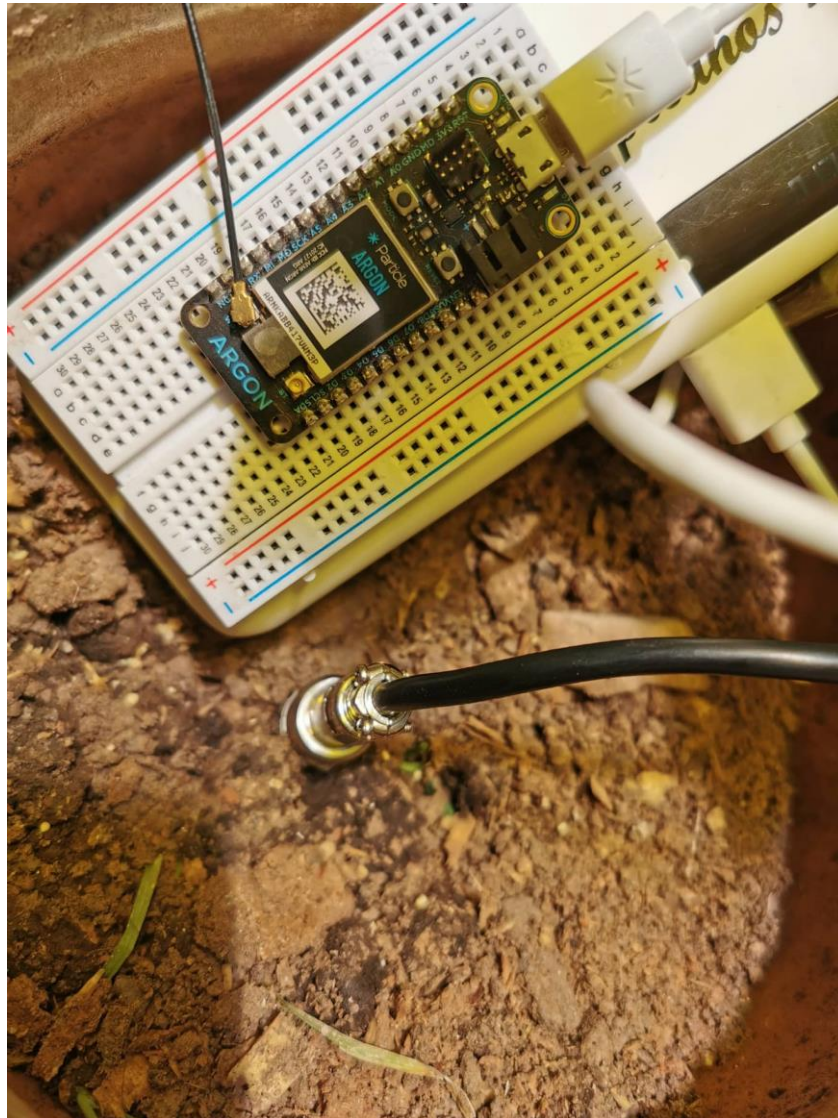
The Particle Team. (2019). *La ultima lista de verificación de seguridad de IoT*.
<https://blog.particle.io/2019/06/27/iot-security/>

Yachay, T. D. P., Castro, S. J. A., Espinosa, L. L. H., & Anaya, I. A. J. (2017). *Sistema de Riego Basado En La Internet De Las Cosas (IoT)*. *April*, 1–9.

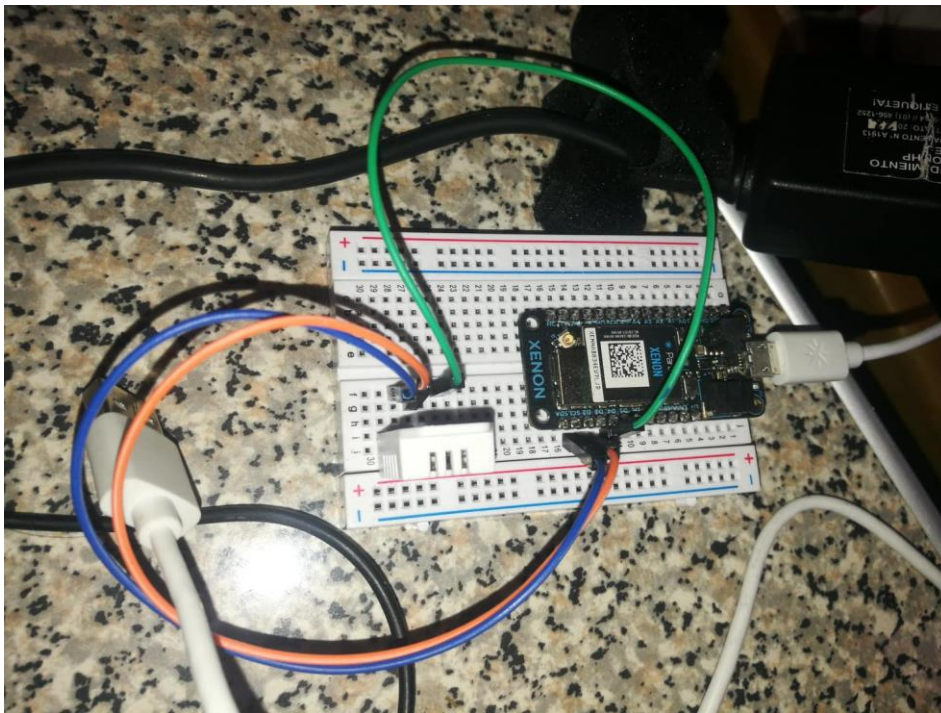
ANEXOS

Anexo 1: Argon (Particle).

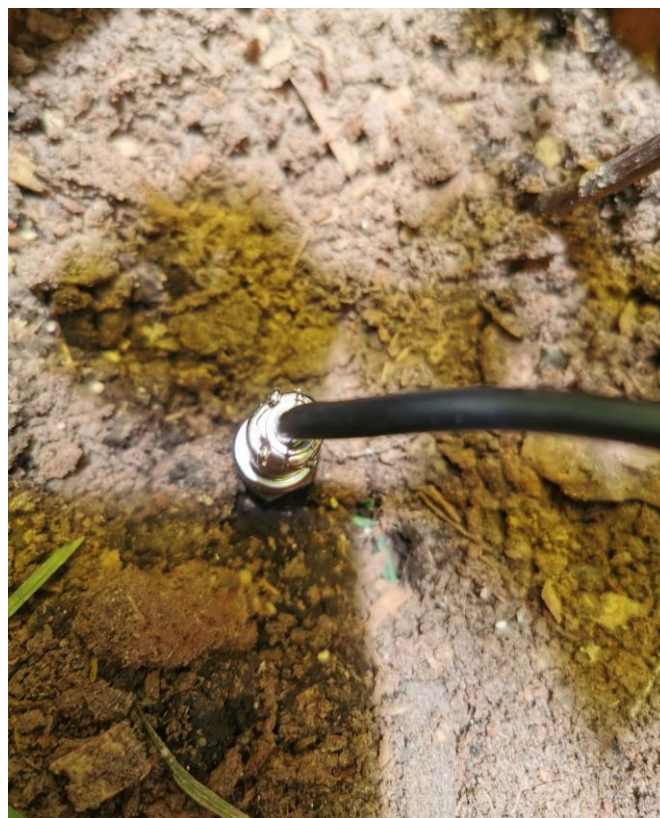
Anexo A 1 Dispositivo Argon (Particle).



Anexo 2 Dispositivo Xenon (Particle).



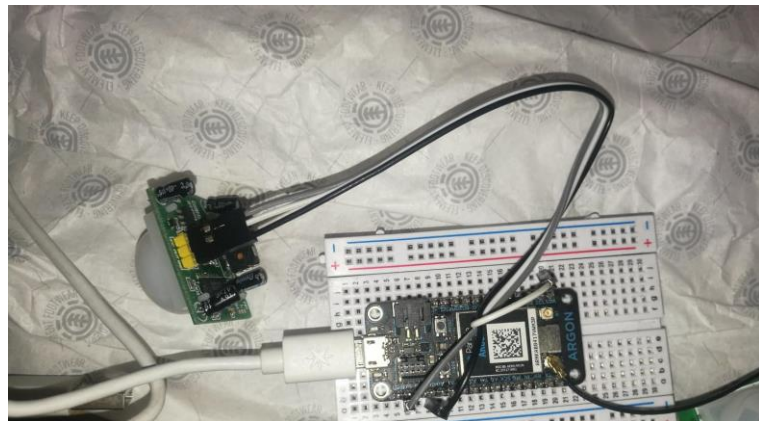
Anexo 3 Sensor de Temperatura-Humedad (Suelo).



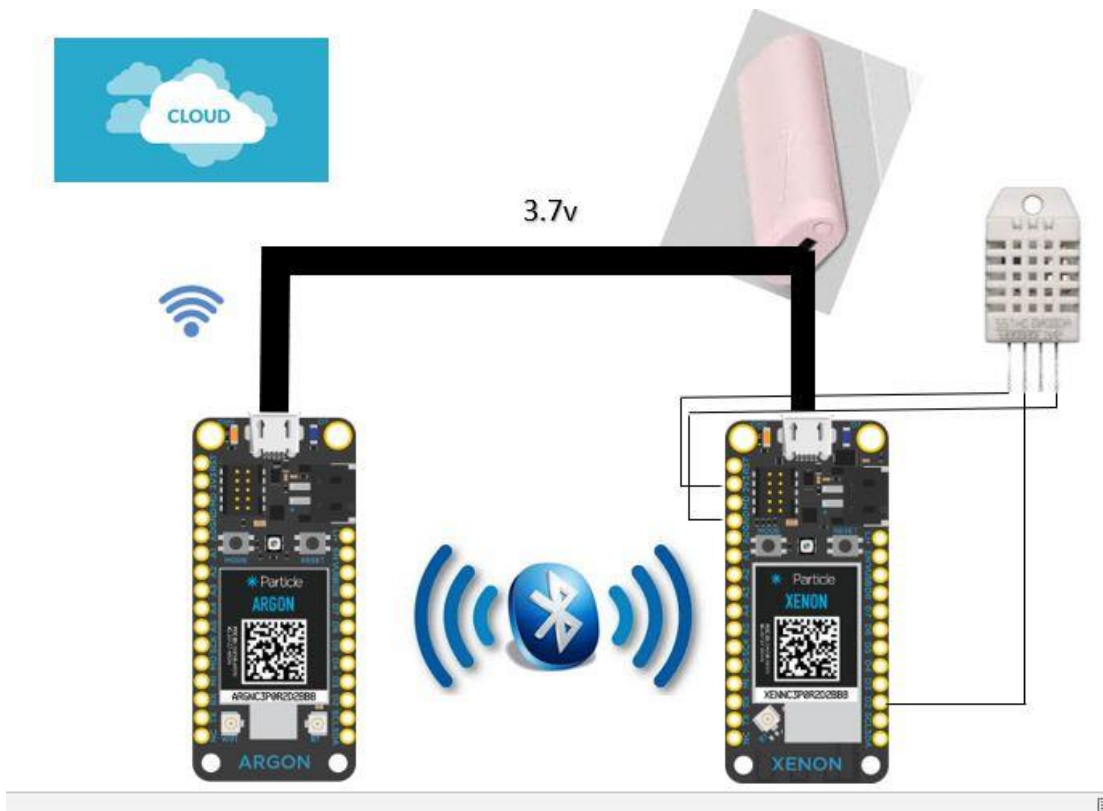
Anexo 4 Sensor de Temperatura-humedad (Ambiente-DHT22).



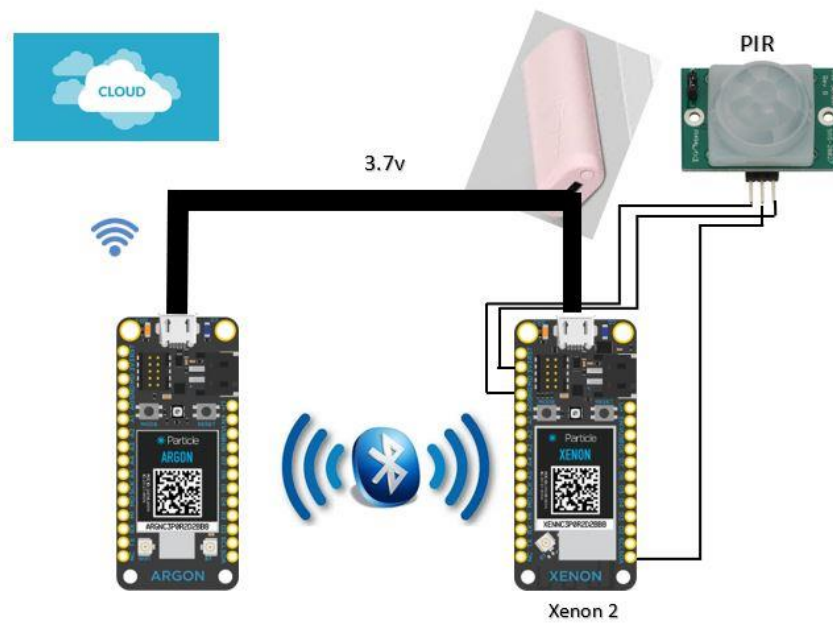
Anexo 5 Sensor de Movimiento (PIR).



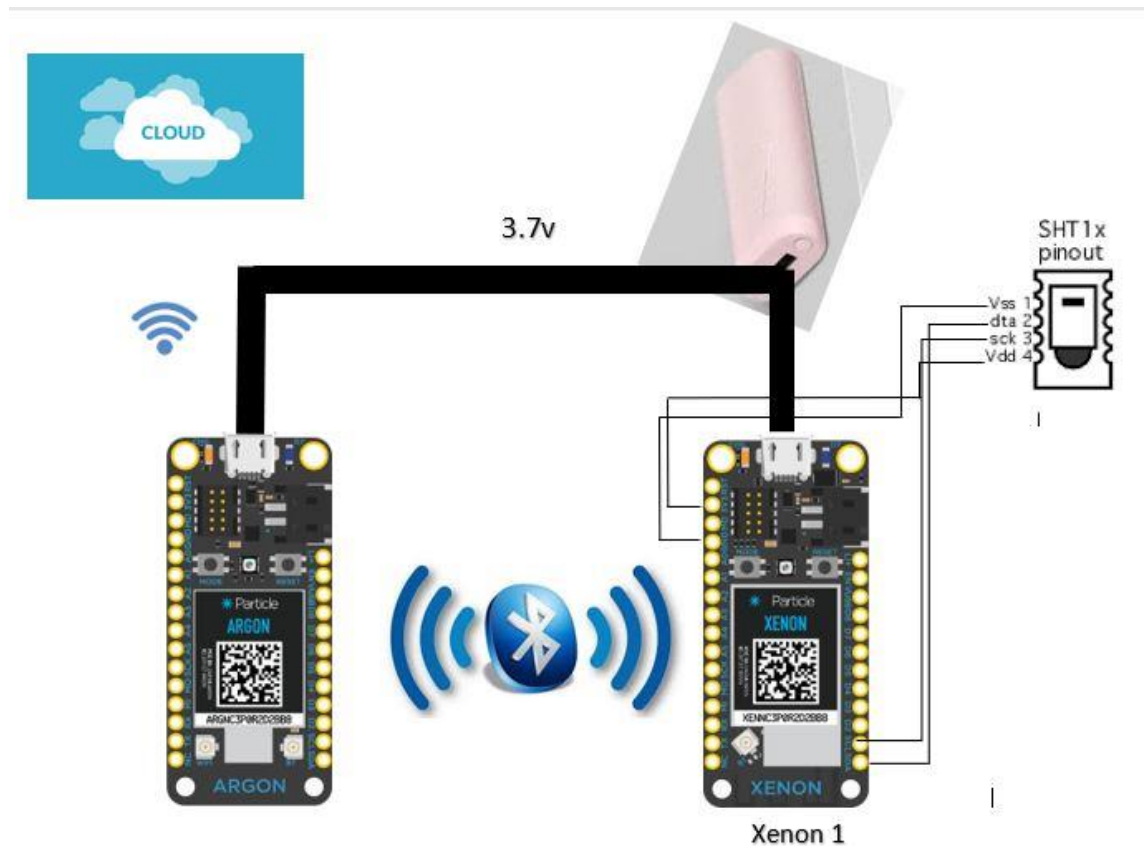
Anexo 6 Diagrama de Conexión del Xenon con el sensor DHT22



Anexo 7 Diagrama de Conexión del Xenon con el sensor PIR



Anexo 8 Diagrama de conexión del Xenon con el sensor SHT11



Anexo 9: Programación para la Publicación a internet (Argon).

```
#include <ThingSpeak.h>
```

```
TCPCClient client;//conectar con thingspeak
```

```
unsigned int myChannelNumber = 922765; // replace with your ChannelID
```

```
const char * myWriteAPIKey = "KV2YB81JG5GIHT19"; // replace with your
```

```
WriteAPIKey
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  ThingSpeak.begin(client);//iniciamos thingspeak
```

```
  Mesh.subscribe("humsuelo", humidity);
```

```
  delay(2000);
```

```
  Mesh.subscribe("tempsuelo", temperature);
```



```
        delay(2000);
    Mesh.subscribe("tempaire", temp);
        delay(2000);
    Mesh.subscribe("humaire", hum);
        delay(6000);
    Particle.subscribe("hook-response/temperatura", myHandler, MY_DEVICES);
}
void myHandler(const char *event, const char *data) {
}
void loop ()
{
    humidity(const char *event, const char *humsuelo);
}
void humidity(const char *event, const char *humsuelo)
{
    Particle.publish("humedadsuelo", String(humsuelo));
    ThingSpeak.setField(4,String(humsuelo));
    ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
}
void temperature(const char *event, const char *tempsuelo)
{
    Particle.publish("temperaturesuelo", String(tempsuelo));
    ThingSpeak.setField(3,String(tempsuelo));
    ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    delay(3000);
}
void temp(const char *event, const char *tempambiente)
{
    Particle.publish("temperatura", String(tempambiente));
    ThingSpeak.setField(1,String(tempambiente));
    ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    delay(3000);
    Particle.publish("tesisamb", "la temperatura es " + String(tempambiente) +
    "°C", 600 ,PRIVATE); //publish an event
```



```
        delay(1000);
    }
    void hum(const char *event, const char *humambiente)
    {
        Particle.publish("humedad", String(humambiente));
        ThingSpeak.setField(2,String(humambiente));
        ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
        delay(3000);
        {if(abs(int(humambiente)) > 10)
        {
            Particle.publish("tesisamb", "la humedad es " + String(humambiente) + "% "
+ "se necesita", 600 ,PRIVATE); //publish an event
            delay(1000);
        }}
    }
```

Anexo 10: Programación del sensor de temperatura y humedad Ambiente.

(Xenon 4)

```
// This #include statement was automatically added by the Particle IDE.
#include <Adafruit_DHT.h>
#include "math.h"
// Definimos el tipo de sensor o el modelo
#define DHTTYPE DHT22          // DHT 22 (AM2302)

// DHT22 sensor pinout/salidas:
// Pin 1 (on the left/izquierda): +3.3V
// Pin 2: output/salida
// Pin 4 (on the right/derecha): GROUND
#define DHT_5V_PIN D1
#define DHT_SENSOR_PIN D2
#define DHT_GROUND_PIN D4

DHT dht(DHT_SENSOR_PIN, DHTTYPE);
```



```
/* Thingspeak */
//TCPClient client;
//unsigned long myChannelNumber = 184077;
//const char * myWriteAPIKey = "39662E74UQB6RK0D";

void setup() {

    pinMode(DHT_5V_PIN, OUTPUT);
    pinMode(DHT_GROUND_PIN, OUTPUT);
    digitalWrite(DHT_5V_PIN, HIGH);
    digitalWrite(DHT_GROUND_PIN, LOW);

    // esperando a que el sensor se estabilice
    delay(1000);

    // Inicialización del sensor
    dht.begin();

    // Lectura del sensor
    double temp = dht.getTempCelcius();
    double hum = dht.getHumidity();
}

void loop() {
    delay(1000);
    double temp = dht.getTempCelcius();
    double hum = dht.getHumidity();
    //double lastTemperature;

    //if (digitalRead(D2) == HIGH) ...
    {
        //digitalWrite(D7,HIGH); //show the activity of the sensor with the onboard

```

D7 LED



```
//if(abs(Temperature-lastTemperature)>2)
{
//Particle.publish("Temp", String (temp) + " °C" , 600 ,PRIVATE); //publish
an event
delay(1000);//Particle.publish("Hum", String (hum) + " %" , 600 ,PRIVATE);
Mesh.publish("tempaire", String(temp));
Mesh.publish("humaire", String(hum));

delay(10000);
// while (digitalRead(D0) == HIGH); // wait for the sensor to return back to
normal
//System.sleep(SLEEP_MODE_DEEP, 15 * 60);
}
// digitalWrite(D7,LOW);
}
// This will run because the system is sleeping
```

Anexo 11: Programación del sensor de movimiento PIR.(xenón 2)

```
int inputPin = D0;          // elige el pin de entrada
int ledPin = D7;           // pin LED
int pirState = LOW;        // comenzamos asumiendo que no se detecta
movimiento
int val = 0;               // variable para leer el estado de PIN

int calibrateTime = 5000;  // Espera a que la cosa se calibre

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT); // declara el sensor como entrada
```



```
Particle.subscribe("hook-response/pushbulletHOME", myHandler,
MY_DEVICES);
}

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

// manejar la respuesta de integración
}

void loop() {

// si el sensor esta calibrado
if (calibrated()) {
// Obtener datos del sensor
readTheSensor();

// report it out, if the state has changed
reportTheData(); ...
}
}

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

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

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

void reportTheData() {
if (val == HIGH) {
// si el estado actual no es movimiento
// es decir solo ah cambiado
// anuncia este cambio publicando en evento
```



```
    if (pirState == LOW) {  
        // acabamos de encender  
        Particle.publish("movimiento", "detectado", PRIVATE);  
        // actualizar el estado actual  
        pirState = HIGH;  
        setLED(pirState);  
    }  
}  
}
```

Anexo 12: Programación para sensor de temperatura y humedad del suelo (Xenon 1)

```
#include "application.h"  
#include "SHT1x.h"  
  
//SYSTEM_MODE(MANUAL);  
  
//Connections:  
// SHT_RED    ELECTRON_3V3  
// SHT_BLACK  ELECTRON_GND  
// SHT_YELLOW ELECTRON_D1 CLOCK  
// SHT_GREEN  ELECTRON_D0 DATA  
  
// Specify data and clock connections and instantiate SHT1x object  
#define dataPin D0  
#define clockPin D1  
SHT1x sht1x(dataPin, clockPin);  
  
#define MAX_TEMP    100.0    ...  
#define MIN_TEMP    40.0  
#define MIN_HUMIDITY 30.0
```




```
uint32_t previousMillis = 0;
const uint32_t interval = 1000; //milliseconds
bool state = 0;

int soilPublish(String command);

void setup()
{

// Particle.function("soil", soilPublish);
//Serial.begin(9600); // Open serial connection to report values to host
//Serial.println("Starting up");
//pinMode(D7,OUTPUT);

//leer el sensor
//float temperature = sht1x.readTemperatureC();
//float humidity = sht1x.readHumidity();

// Update the 2 ThingSpeak fields with the new data
//ThingSpeak.setField(1, (float)temperature);
//ThingSpeak.setField(2, (float)humidity);

//Mesh.publish("humsuelo", String(humidity));
// delay(3000);
// Mesh.publish("tempsuelo", String(temperature));
// Write the fields that you've set all at once.
//ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

// Give time for the message to reach ThingSpeak
// delay(3000);

// Sleep for 15 minutes to save battery
//System.sleep(SLEEP_MODE_DEEP, 15 * 60);
```



```
}

void loop()
{

    float temperature = sht1x.readTemperatureC();
    float humidity = sht1x.readHumidity();

    Mesh.publish("humsuelo", String(humidity));
    delay(3000);
    Mesh.publish("tempsuelo", String(temperature));
    ...

    // Write the fields that you've set all at once.
    //ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

    // Give time for the message to reach ThingSpeak
    delay(3000);
```

Anexo 13: Programación para sistema de riego y encendido de led con Blynk

```
// This #include statement was automatically added by the Particle IDE.
#include <blynk.h>

#define BLYNK_PRINT Serial // Set serial output for debug prints
// #define BLYNK_DEBUG // Uncomment this to see detailed prints

// #include <blynk.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
int status = 0;
int controlriego = D7;
char auth[] = "_a81xTH2IT0Y7NE0gsG42jjXMmeEVZ2g";
```



// Attach a Button widget (mode: Switch) to the Digital pin 7 - and control the built-in blue led.

// Attach a Graph widget to Analog pin 1

// Attach a Gauge widget to Analog pin 2

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  delay(5000); // Allow board to settle
```

```
  pinMode(controlriego, OUTPUT);
```

```
  Blynk.begin(auth);
```

```
}
```

```
void loop()
```

```
{
```

```
  Blynk.run();
```

```
  riego();
```

```
}
```

```
void riego()
```

```
{
```

```
  status = digitalRead(controlriego);
```

```
  if (status == 1){
```

```
    delay(1800);
```

```
    Blynk.notify (" Se inicio el proceso de riego ");
```

```
  }
```

```
}
```