



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

TEMA:

**Diseño de una red de sensores inalámbricos para la
horticultura climáticamente inteligente para un invernadero
de rosas mediante la plataforma Ubidots**

AUTOR:

Chaucala Escalante, Joel Alexander

**Trabajo de titulación previo a la obtención del título de
INGENIERÍA EN TELECOMUNICACIONES**

TUTOR:

Mgs. Vélez Tacuri, Efraín Oliverio

Guayaquil, Ecuador

09 de septiembre del 2022



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

**FACULTAD EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación, fue realizado en su totalidad por **Chaucala Escalante, Joel Alexander**, como requerimiento para la obtención del título de **Ingeniería en Telecomunicaciones**.

TUTOR

f. _____
Mgs. Vélez Tacuri, Efraín Oliverio

DIRECTOR DE LA CARRERA

f. _____
MSc. Bohórquez Escobar, Celso Bayardo

Guayaquil, a los 09 del mes de septiembre del año 2022



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

**FACULTAD EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Chaucala Escalante, Joel Alexander**

DECLARO QUE:

El Trabajo de Titulación, **Diseño de una red de sensores inalámbricos para la horticultura climáticamente inteligente para un invernadero de rosas mediante la plataforma Ubidots**, previo a la obtención del título de **Ingeniería en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 09 del mes de septiembre del año 2022

EL AUTOR

f. Joel Chaucala E.
Chaucala Escalante, Joel Alexander



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

**FACULTAD EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, **Chaucala Escalante, Joel Alexander**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **Diseño de una red de sensores inalámbricos para la horticultura climáticamente inteligente para un invernadero de rosas mediante la plataforma Ubidots**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 09 del mes de septiembre del año 2022

EL AUTOR

f. Joel Chaucala E.
Chaucala Escalante, Joel Alexander

REPORTE DE URKUND



Document Information

Analyzed document	Chaucala Joel - Trabajo Titulacion.docx (D143635306)
Submitted	9/5/2022 11:01:00 PM
Submitted by	
Submitter email	joel.chaucala@cu.ucsg.edu.ec
Similarity	1%
Analysis address	efrain.velez.ucsg@analysis.orkund.com

Sources included in the report

SA	TEISIS UNO -4.docx Document TEISIS UNO -4.docx (D29503993)		3
SA	AVANCE DEL DOCUMENTO (modificado).docx Document AVANCE DEL DOCUMENTO (modificado).docx (D71147874)		1
SA	Proyecto tecnico Calva.docx Document Proyecto tecnico Calva.docx (D25683986)		1

Reporte Urkund del trabajo de titulación de la Carrera ingeniería en telecomunicaciones denominado: DISEÑO DE UNA RED DE SENSORES INALÁMBRICOS PARA LA HORTICULTURA CLIMÁTICAMENTE INTELIGENTE PARA UN INVERNADERO DE ROSAS MEDIANTE LA PLATAFORMA UBIDOTS, del estudiante CHAUCALA ESCALANTE JOEL ALEXANDER se encuentra al 1% de coincidencias.

TUTOR

ING. VELEZ TACURI EFRAIN

DEDICATORIA

Este trabajo de titulación está dedicado para mi padre, Edmundo Jarama y mi madre, Jaqueline Escalante, quienes han sido mis maestros de toda la vida, pues con ellos he aprendido a vivir de la mejor forma y me han inculcado valores y virtudes que pongo en práctica todos los días. Siempre han sido mi apoyo fundamental durante las situaciones buenas y malas que se han presentado. Nunca han dejado de apoyarme económicamente y siempre han hecho un esfuerzo incalculable por mi bienestar. Estoy muy orgulloso de ellos, y mediante mi esfuerzo, quiero que sepan que han formado en mí, una persona profesional y de buenos principios.

A mi tío materno, Xavier Escalante, quien ha sido mi padre de crianza desde niño, me ha enseñado a ser constante y responsable con mis tareas. Me ha demostrado que una persona nunca se rinde y siempre tiene mejorar porque no existe obstáculos, solo nosotros nos imponemos el límite, además de que no hay tiempo para dudar de nuestras capacidades. A mi abuela materna, Maria Consuegra, quien junto a mi tío materno, me acogieron de pequeño y me criaron con toda su paciencia y amor. Al igual que mis padres, me han dado lo mejor en lo sentimental, educativo y económico. Me siento muy afortunado en tenerlos, y por eso, les dedico este trabajo de titulación como muestra de mi amor, respeto, y admiración.

EL AUTOR

f. Joel Chaucala E.
Chaucala Escalante, Joel Alexander

AGRADECIMIENTOS

Agradezco a Dios, por la salud y sabiduría que me ha otorgado para llegar a la culminación de mi etapa universitaria. Agradezco las bendiciones que me ha entregado, sus enseñanzas y por hacerme sentir que todos tenemos una oportunidad para mejorar y seguir adelante pese a las derrotas que hemos tenido a lo largo de la vida. Su infinita bondad y amor permite que me llene de vida y motivación para encontrar mi verdadera identidad y deseo de superación.

Agradezco a mi enamorada, Anggie Avilés, por enseñarme la calidez y el amor de compartir con otras personas. Por ser la pareja con quien quiero desarrollar mi vida. Por las atenciones y cuidados que me brinda, y por su ayuda en momentos buenos y malos. Me ha demostrado ser una mujer ideal y el tiempo que compartimos es invaluable para mí.

A mi hermana, Leslie Jarama, quien pese a todo siempre ha sabido ayudarme y es una persona querida para mí. Las diferencias que tenemos me hacen entender que somos dos personas opuestas con metas distintas, pero con el camino común de velar por el bienestar mutuo.

Agradezco a mi primo, Kevin Lázaro, quien es como uno hermano para mí porque me ha ayudado siempre a continuar con mis metas y siempre me brinda su ayuda en cualquier situación. Admiro su personalidad honesta y compañerismo como amigo y como familia. Que este trabajo de titulación sirva como un ejemplo de superación para sus metas personales.

Agradezco a mi familia paterna de apellido Jarama y familia materna de apellido Escalante, porque cada uno de ellos me entregaron un poco de su tiempo, confianza y amor para ser la persona que soy en la actualidad. Nunca me apartaron de su calidez. Sus acciones y palabras me hicieron sentir que tengo un lugar al cual pertenezco.

Agradezco al ingeniero Bayardo Bohórquez, director de carrera, y al ingeniero, Efraín Vélez, coordinador de titulación, por ser las personas más importantes en mi vida universitaria. Gracias a ellos por la ayuda que me brindaron en las gestiones académicas y por depositar su confianza en mí, como un estudiante capaz de cumplir sus responsabilidades. Les agradezco de corazón porque nunca me dejaron desamparado y por enseñarme que tengo derechos como cualquier otro estudiante.

EL AUTOR

f. Joel Chaucala E.
Chaucala Escalante, Joel Alexander



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**
FACULTAD EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

f. 

MSc. Bohórquez Escobar, Celso Bayardo
DIRECTOR DE CARRERA

f. 

Mgs. Vélez Tacuri, Efraín Oliverio
COORDINADOR DEL ÁREA DE TITULACIÓN

f. 

M. Sc. Bohórquez Heras, Daniel Bayardo
OPONENTE

ÍNDICE GENERAL

ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS.....	XIII
RESUMEN.....	XV
ABSTRACT.....	XVI
Capítulo I: Descripción General del Trabajo de Titulación	2
1.1 Introducción	2
1.2 Antecedentes.....	2
1.3 Definición del problema	3
1.4 Justificación del problema.....	4
1.5 Objetivos del problema de investigación.....	4
1.5.1 Objetivo General	4
1.5.2 Objetivos Específicos	4
1.6 Hipótesis.....	5
1.7 Metodología de la investigación.....	5
Capítulo 2: Fundamentación Teórica	6
2.1 Tecnología de los sensores	6
2.2 Clasificación de los sensores.....	7
2.2.1 Sensores Resistivos.....	8
2.2.2 Sensores Capacitivos.....	8
2.2.3 Sensores Inductivos	9
2.2.4 Sensores Ultrasónicos	10
2.2.5 Sensores Electroquímicos.....	11
2.2.6 Sensores Térmicos	12
2.2.7 Sensores Fotoeléctricos	14
2.2.8 Sensores Piezoeléctricos	15
2.2.9 Sensores Piezorresistivos	16
2.3 Redes de Sensores Inalámbricos	17
2.4 Estándares de las redes de sensores inalámbricos.....	19

2.5	Topologías de redes	20
2.5.1	Topología de red en anillo	20
2.5.2	Topología de red en bus	21
2.5.3	Topología de red en estrella.....	22
2.5.4	Topología de red en árbol	22
2.5.5	Topología de red en malla.....	23
2.6	Microcontroladores	23
2.6.1	Arduino.....	23
2.6.2	PIC	24
2.7	Plataformas IoT	25
2.7.1	Ubidots	26
Capítulo 3: Diseño. Simulación y Resultados.		28
3.1	Selección de sensores para el cultivo de rosas	28
3.2	Estructura de la red basada en Arduino y xBee.....	29
3.2.1	Alimentación de la red de sensores inalámbricos	29
3.2.2	Configuración de los módulos de RF xBee como enlaces inalámbricos.....	32
3.2.3	Configuración del Arduino Uno como nodo esclavo.....	35
3.2.4	Configuración del Arduino Uno Wifi como nodo maestro	40
3.2.5	Diseño de la red de sensores inalámbricos.....	44
3.3	Comunicación entre la red de sensores inalámbricos y la plataforma IoT Ubidots.....	47
3.4	Costo detallado del proyecto	51
CONCLUSIONES		52
RECOMENDACIONES.....		53
Anexo 1: Programación del nodo 1		54
Anexo 2: Programación del nodo 2.....		56
Anexo 3: Programación del nodo 3.....		57
Anexo 4: Programación del nodo Coordinador		59
BIBLIOGRAFÍA.....		64

ÍNDICE DE TABLAS

Capítulo 2:

Tabla 2.1 Especificaciones sobre los estándares basados en la normativa IEEE 802.15.x.....	20
--	----

Capítulo 3:

Tabla 3. 1 Selección de sensores de acuerdo con las variables establecidas	28
Tabla 3. 2 Interconexión de pines entre microcontrolador y módulo RF	33
Tabla 3. 3 Parámetros Técnicos de la red inalámbrica de sensores.....	34
Tabla 3. 4 Parámetros técnicos del Arduino UNO	36
Tabla 3. 5 Asignación de sensores a microcontroladores.....	37
Tabla 3. 6 Especificaciones técnicas de los módulos RF	45
Tabla 3. 7 Conexiones de pines para los sensores según la programación	47
Tabla 3. 8 Costo total del diseño de red de sensores y su plataforma IoT ..	51

ÍNDICE DE FIGURAS

Capítulo 2:

Figura 2.1 Sensor DHT11	6
Figura 2.2 Diagrama de un Sensor Inteligente	7
Figura 2.3 Diagrama eléctrico de un potenciómetro	8
Figura 2.4 Partes de un sensor capacitivo	9
Figura 2.5 Partes de un sensor capacitivo	9
Figura 2.6 Sensor Inductivo Blindado	10
Figura 2.7 Sensor Inductivo no Blindado	10
Figura 2.8 Tipos de sensores ultrasónicos	11
Figura 2.9 Estructura de un sensor electroquímico de gas	12
Figura 2.11 Efecto Seebeck en dos conductores metálicos	13
Figura 2.10 Efecto Peltier a distintas temperaturas	13
Figura 2.12 Efecto Peltier Generación y Absorción de calor	14
Figura 2.13 Sensor fotoeléctrico de tipo reflexivo	15
Figura 2.14 Efecto piezoeléctrico producido por una corriente eléctrica	16
Figura 2.15 Estructura de un sensor Piezorresistivo(Izquierdo) y su diagrama eléctrico(Derecho).....	16
Figura 2.16 Estructura de un nodo de sensor	17
Figura 2.17 Estructura de una red de sensores inalámbricos	18
Figura 2.18 Topología de red en anillo	21
Figura 2.19 Topología de red en bus	21
Figura 2.20 Topología de red en estrella	22
Figura 2.21 Topología de red en bus	22
Figura 2.22 Topología de red en bus	23
Figura 2.23 Encapsulado Atmega 328P	24
Figura 2.24 Encapsulado PIC18F4550	25
Figura 2.25 Arquitectura de la plataforma IoT ThingsBoard	25
Figura 2.26 Arquitectura de la plataforma IoT Ubidots.....	27

Capítulo 3:

Figura 3. 1 Módulo interruptor automático de emergencia.....	30
Figura 3. 2 Módulo regulador de voltaje multicanal.....	30
Figura 3. 3 Conexión entre fuentes y modulo interruptor	31
Figura 3. 4 Alimentación de respaldo en funcionamiento	32
Figura 3. 5 Esquema gráfico de conexión entre Arduino y xBee	32
Figura 3. 6 Conexión física para la configuración del módulo.....	33
Figura 3. 7 Configuración y verificación mediante software XCTU	34
Figura 3. 8 Tarjetas Arduino Uno Rev.3.....	35
Figura 3. 9 Diagrama de entradas y salidas del Arduino Uno.....	36
Figura 3. 11 Conexión entre 4 sensores y el primer nodo mediante Shield. 38	
Figura 3. 10 I/O Expansion Shield V7.1, de la compañía DFROBOT	38
Figura 3. 12 Conexión entre 3 sensores y el segundo nodo mediante Shield	39
Figura 3.14 Tarjeta Arduino Uno Wifi Rev.2	40
Figura 3. 13 Conexión entre 4 sensores y el tercer nodo mediante Shield..	40
Figura 3. 16 Librería “WiFiNINA”.....	41
Figura 3. 15 Librería “Arduino AVR Boards”	41
Figura 3. 18 Mensaje de placa desactualizada.....	42
Figura 3. 17 Proyecto “CheckFirmwareVersion”	42
Figura 3. 20 Interfaz para actualizar firmware de la placa	43
Figura 3. 19 Proyecto “FirmwareUpdater” y su complicación	43
Figura 3. 21 Interfaz para actualizar firmware de la placa	44
Figura 3. 22 Arquitectura de la red de sensores inalámbricos	46
Figura 3. 23 Creación del dispositivo en la plataforma Ubidots	48
Figura 3. 24 Creación del dispositivo en la plataforma Ubidots	48
Figura 3. 25 Sketch programado en Arduino Wifi Rev.2.....	49
Figura 3. 26 Interfaz gráfica destinada al monitoreo de las variables	50
Figura 3. 27 Datos almacenados de la variable “hum_suel” enviados desde el nodo coordinador.....	50

RESUMEN

El presente trabajo de titulación plantea un diseño de red de sensores inalámbricos que permite el monitoreo de los variables climáticas y recursos presentes en un invernadero de cultivo de rosas, con la finalidad de comprender la incidencia del entorno en el comportamiento de las rosas para identificar y proponer mejoras en el cuidado y crecimiento, asegurando así la calidad de la producción. El primer capítulo describe la situación del cultivo de rosas en regiones que no son óptimas para su desarrollo, con enfoque en la horticultura de un invernadero, y su necesidad de implementar sensores que genere información para la gestión de respuestas adecuadas frente a un requerimiento ambiental. Por ello, se plantea el problema y su solución mediante el objetivo general y objetivos específicos, así como el tipo de investigación implementado en el trabajo de titulación. En el segundo capítulo se encuentra la fundamentación teórica con la finalidad de comprender en detalle las partes del diseño de red mediante recursos académicos y científicos. En el tercer capítulo se determina las variables que se definieron para el desarrollo del trabajo de titulación, así como el diseño de los nodos, la programación de cada uno en función a los sensores seleccionado, la configuración de los módulos radiofrecuencia utilizados, la comunicación del nodo principal y la plataforma IoT remota, y el diseño de la interfaz en la plataforma IoT para la visualización de datos en tiempo real.

Palabras Claves: Zigbee, Ubidots, IoT, Sensores, Radiofrecuencia, Redes, Invernadero, Rosas, Arduino.

ABSTRACT

This degree work proposes a wireless sensor network design that allows the monitoring of climatic variables and resources present in a rose growing greenhouse, to understand the impact of the environment on the behavior of roses to identify and propose improvements in the care and growth, thus ensuring the quality of production. The first chapter describes the situation of rose cultivation in regions that are not optimal for its development, with a focus on greenhouse horticulture, and the need to implement sensors that generate information for the management of appropriate responses to an environmental requirement. Due to this, the problem and its solution are presented through the general objective and specific objectives, as well as the type of research implemented in the degree work. In the second chapter is the theoretical foundation to understand in detail the parts of the network design through academic and scientific resources. The third chapter determines the variables that were defined for the development of the degree work, as well as the design of the nodes, the programming of each one according to the selected sensors, the configuration of the radio frequency modules used, the communication of the main node and the remote IoT platform, and the design of the interface in the IoT platform for the visualization of data in real time.

Keywords: Zigbee, Ubidots, IoT, Sensors, Radio Frequency, Networks, Greenhouse, Roses, Arduino.

Capítulo I: Descripción General del Trabajo de Titulación

1.1 Introducción

La tecnología tiene un gran adaptación para los diferentes campos de investigación y producción de tal manera que otorga muchos beneficios y genera un abanico muy amplio de posibilidades con respecto a su aplicación. Gracias a esto tenemos procesos automatizados, equipos de gestión, comunicaciones remotas, almacenamiento de datos, y estructuras híbridas que brindan apoyo en el desarrollo industrial. El desarrollo de la tecnología del internet de las cosas ha despertado el interés en el sector industrial al considerar que la información que se gestiona y se obtiene de forma precisa, permite que los recursos sean aprovechados de forma óptima e incrementa la productividad.

La horticultura es uno de los campos que se benefician con el uso adecuado de herramientas tecnológicas. Una de estas herramientas son los sensores con distintas aplicaciones, ya que cada sensor cumple con un propósito específico al recolectar información sobre una variable. Por ello, se pretende demostrar a través de un diseño de red de sensores que la gestión remota de los parámetros requeridos para el caso que se ha planteado cumple con los objetivos propuestos. Esta red podrá mantener informado al administrador mediante una plataforma remota, considerando que la consulta podrá ser ejecutada mediante cualquier dispositivo que tenga acceso a internet. Este modelo es oportuno para que las personas formen parte de la red digital a través de la adquisición de datos y reducción de tiempos en gestión.

1.2 Antecedentes

López (2021) explica que la horticultura en el Ecuador se viene desarrollando desde los años 90 y con el paso del tiempo se ha convertido en una actividad muy importante ya que las exportaciones del Ecuador respecto a las rosas son el 77% de las flores totales exportadas anualmente. Dado que esta actividad resulta ser provechosa, se tiene especial cuidado al cultivar para mantener su calidad y cantidad de producción. Se toma muy en cuenta

el lugar donde se cultivarán, y la región sierra del Ecuador no deja de ser la principal sección geográfica ideal para el cultivo en espacio abierto o cerrado de variedades de flores.

La producción de flores, especialmente de rosas, permite a los productores campesinos obtener una fuente alternativa económica, al mismo tiempo de crear una sinergia empresarial para el desarrollo en las exportaciones, pero la inversión del productor campesino se convierte en riesgo cuando no cuenta con un sistema de monitoreo adecuado. De acuerdo con Velasco (2019), existen condiciones ambientales óptimas para el cultivo de flores tales como:

- Temperaturas ambientales dentro de 15 °C a 25 °C, siendo 25 °C en la mañana y 15 °C en la noche.
- Humedad ambiental entre 70% y 80%
- Humedad del suelo superior al 60%
- Presencia de dióxido de carbono aproximadamente de 1.000 ppm.
- Nivel de pH del suelo constante a 6.
- Nivel de iluminación no superior a 3.000 lux.
- Contaminación con óxidos de nitrógeno limitada a 60ppb.

La exigencia de estos parámetros específicos causa que la región costa no sea un lugar óptimo para el desarrollo y producción de las rosas, ni de otras flores. Sin embargo, son pocos los productores costeños, que realizan una inversión en invernaderos climatizados para producir flores en general. De esta manera, los productores interesados no cuentan con presupuesto ni herramientas alternativas para crear su propio espacio de producción. El presente trabajo se encargará de diseñar un sistema que permita solventar dichos inconvenientes.

1.3 Definición del problema

La horticultura de plantas que no son propias de una región, tiende a no desarrollarse adecuadamente, además del poco aprovechamiento de herramientas de medición. Esto ha llevado a muchos horticultores a estar en situaciones de mucha presión para crear espacios cerrados con una atmósfera artificial e idónea para cultivar diferentes tipos de plantas. La

aplicación de tecnología para administración remota en este ámbito pretender solventar algunas problemas y ser un apoyo durante todo el proceso de producción de los horticultores con información relevante para la inspección de su producción y calidad. De esta forma, permite mejorar las cosechas y su calidad frente a cambios ambientales controlados y monitorizados. Por lo tanto, se origina la siguiente interrogante: ¿Cómo se puede justificar el uso de sensores inalámbricos en la integración de la horticultura de un invernadero?

1.4 Justificación del problema

En la actualidad, la capacidad de adaptación de la tecnología en otros campos de investigación y desarrollo ha permitido generar un gran rendimiento en procesos y producción de artículos y servicios que son destinados al consumo de las personas. La facultad de gestión a través de interfaces que brindan datos precisos y permiten un control oportuno de un ambiente se ha convertido en una necesidad indispensable en casi cualquier infraestructura. Dentro de la horticultura también se pretende establecer un modelo de infraestructura que permita la gestión no solo presencial, sino remota con la finalidad de aprovechar todos los recursos. el presente modelo permitirá conocer todo el proceso de diseño de la red y su comunicación remota entre el microcontrolador principal Arduino y la plataforma Ubidots, además de la comunicación interna entre los sensores inalámbricos y el microcontrolador.

1.5 Objetivos del problema de investigación

1.5.1 Objetivo General

Diseñar una red de sensores inalámbricos que permitan la gestión y control de los parámetros de clima mediante una plataforma IoT para un invernadero de rosas.

1.5.2 Objetivos Específicos

- Explicar la selección de sensores inalámbricos en la medición de variables climáticas en cultivo de rosas.
- Estructurar una red basada en microcontroladores Arduino y módulos RF, además de su programación.

- Usar la plataforma Ubidots en la medición y control de las variables climáticas establecidas a través de una interfaz gráfica.

1.6 Hipótesis

El diseño de red de sensores permite mejorar las capacidades de gestión de los variables establecidas gracias a una comunicación remota y en tiempo real, permitiendo monitorear y solventar las necesidades requeridas y obtener un mejor tiempo de respuesta y cuidado durante la etapa de desarrollo de las rosas dentro del invernadero.

1.7 Metodología de la investigación

El presente trabajo de titulación es una investigación experimental y explicativa, ya que pretende estructurar un sistema de monitoreo capaz de brindar datos de manera remota y oportuna acerca de los diferentes variables que se pueden manejar dentro del campo de cultivo de flores, ya que la demanda climática que exige este proceso, debe mantener en un rango ideal para el crecimiento y desarrollo. Por la recopilación, almacenamiento y presentación de datos a través de la plataforma web, se considera una investigación cualitativa, porque los datos son capaces de obtenerse como resultado de la comunicación entre la red de sensores y la plataforma. El trabajo de titulación también será de tipo descriptivo, ya que se pretende documentar el proceso de funcionamiento y desarrollo estructural de la red de sensores.

La metodología consiste en definir las variables climáticas que se pretende gestionar en función al invernadero de rosas con la finalidad de desarrollar un esquema de comunicación entre sensores y el microcontrolador encargado de unificar los datos y enviarlo a la plataforma de Ubidots. Después se desarrollará la programación de los sensores utilizando componentes físicos que permitan una lectura precisa del ambiente real, permitiendo así que el trabajo sea aprovechado en todos los recursos que entregue. Por último se realizará la interfaz y su comunicación con el microcontrolador para realizar la obtención y control de datos verificando su integridad con respecto a su aplicación y uso cumpliendo así con los objetivos establecidos.

Capítulo 2: Fundamentación Teórica

2.1 Tecnología de los sensores

La necesidades de automatización de procesos ha permitido interconectar un gran número de dispositivos, formando así un sistema de dispositivos capaz de satisfacer las necesidades en distintas aplicaciones, mejorando la capacidad de control y supervisión en tareas continuas de los usuarios. Estos dispositivos capaces de obtener y entregar información de un parámetro definido lo realizan a través de una unidad de control principal encargada de recolectar y proveer los datos a un módulo de visualización o software final. De acuerdo con Corrêa (2021), el sensor se puede definir como un dispositivo electrónico capaz de generar una respuesta a través de una señal eléctrica frente a la exposición de un fenómeno físico o químico. Un ejemplo es el sensor de humedad y temperatura mostrado en la figura 2.1.

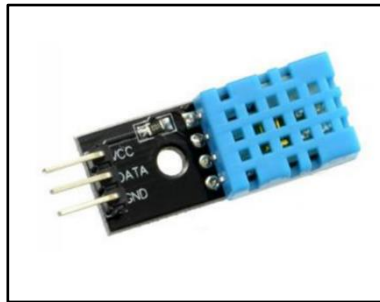


Figura 2.1 Sensor DHT11
Fuente: (Guasch & Calleja, 2019)

La evolución de los sensores inteligentes ha permitido estructurar un sistema de control para mejorar la estimación de la cantidad censada a través de protocolos, convertidores y microcontroladores. Hidalgo (2019) indica que los sensores complejos y de alta estimación cuenta con un esquema con un microcontrolador y memoria que permite obtener datos de una forma precisa, al contrario de los sensores que carecen de estas mejoras técnicas. El esquema varía según la necesidad del parámetro a medir, pero se lo puede representar tal como en la figura 2.2.

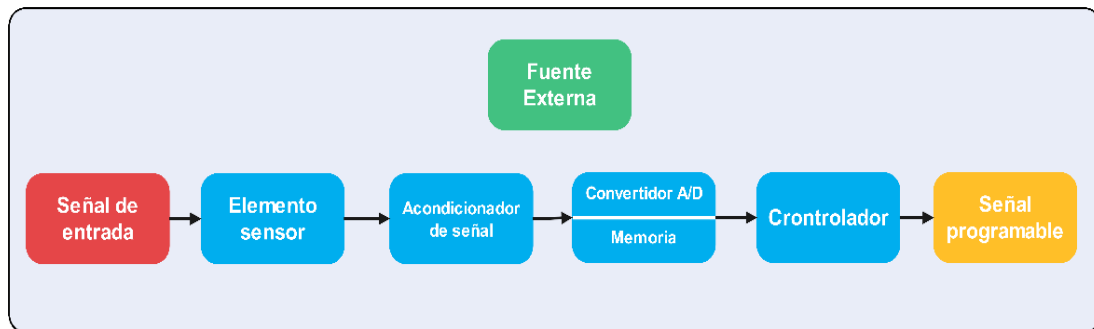


Figura 2.2 Diagrama de un Sensor Inteligente

Fuente: Autor

Dentro del diagrama existe una señal, por lo general eléctrica, que será recogida por el elemento sensor. Corrêa (2021) explica que el acondicionador de señal puede ser utilizado para ajustar la señal con respecto a su amplificación o atenuación, además de eliminar frecuencias no deseadas y realizar la compensación de cantidades, este último se lo utiliza más en los sensores de temperatura. El convertidor A/D realiza la transformación de la señal analógica a digital y posteriormente es almacenada en la memoria. El controlador hace uso de esa señal digital para adaptarla según las necesidades del usuario.

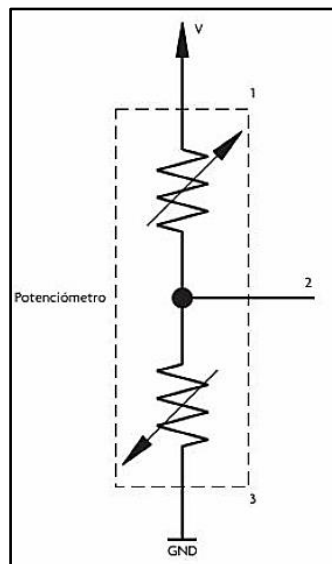
2.2 Clasificación de los sensores

La clasificación de los sensores es amplia ya que puede ser definidas y agrupadas bajo diferentes criterios. Para este caso, la clasificación de los sensores se definirá por el principio de transducción presentado por Corona et al. (2014) de la siguiente manera:

- Resistivo
- Capacitivo
- Inductivo
- Ultrasónico
- Electroquímico
- Térmico
- Fotoeléctrico
- Piezoeléctrico
- Piezorresistivo

2.2.1 Sensores Resistivos

Los dispositivos resistivos tienen la particularidad de que sus electrones fluyan de manera aleatoria de acuerdo con el principio de los cambios que ocurren en la resistencia eléctrica. De acuerdo con la ley de Ohm, la resistencia eléctrica no solo depende de la resistividad del material, sino de su geometría como lo puede ser su sección del área y su longitud. De acuerdo con Hering & Schönfelder (2022), la aplicación de este tipo de sensor tiene como finalidad la posición angular, debido a que incorpora un terminal que recorre el dispositivo resistivo y registra el cambio en la resistencia. Usualmente se compone de tres terminales como se observa en la figura 2.3.



*Figura 2.3 Diagrama eléctrico de un potenciómetro
Fuente: (Corona et al., 2014)*

2.2.2 Sensores Capacitivos

Los dispositivos capacitivos son ideales para analizar la permitividad dieléctrica y factor de pérdida del material, puesto que genera un cambio en la capacitancia a través de los electrodos del sensor (Lang, 2021). Esto permite que estos dispositivos sean diseñados para la detección de la proximidad de un objeto, ya que no requieren contacto, similar a los sensores inductivos y ultrasónicos. De acuerdo con Corona et al (2014), puede trabajar sobre la mayoría de los materiales tanto conductores como no conductores,

sin embargo, en los objetos no conductores se debe de tener en cuenta factores como el tamaño del objeto y la constante dieléctrica.

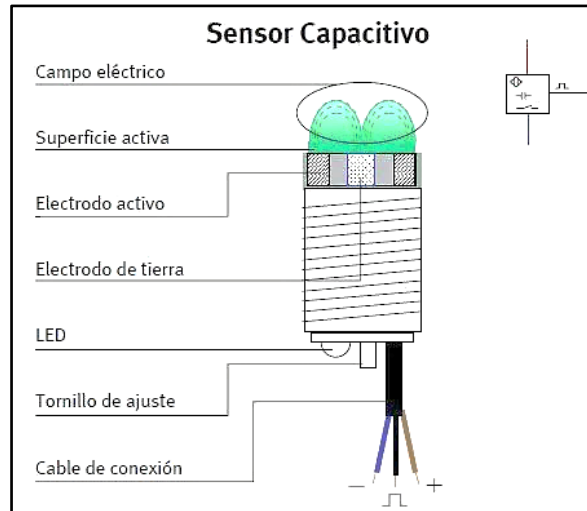


Figura 2.4 Partes de un sensor capacitivo

Fuente: (Astudillo, 2016)

2.2.3 Sensores Inductivos

Este tipos de dispositivos trabaja de forma similar que los sensores capacitivos, sin embargo, solo pueden ser utilizados con materiales conductores. Hering & Schönfelder (2022) explican que el principio de transducción de este sensor se basa en el efecto Hall, el cual un campo magnético generado por una corriente eléctrica que fluye en un conductor, ejerce una fuerza transversal y produce un voltaje en los extremos del conductor. Su aplicación más común es como sensor de proximidad.

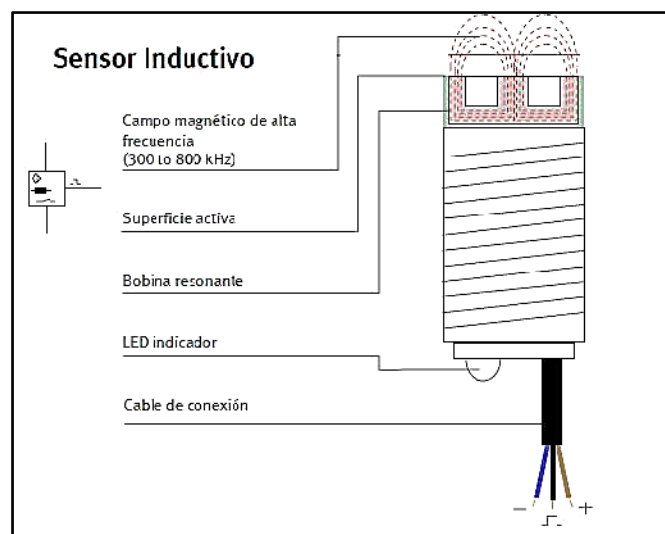


Figura 2.5 Partes de un sensor capacitivo

Fuente: (Castaño, 2016)

Fericean (2018) indica que existen dos tipos de sensores inductivo; blindado y no blindado. La diferencia radica en que los sensores blindados tienen la particularidad de bloquear el campo electromagnético que genera el sensor en los laterales gracias al revestimiento del núcleo de ferrita conformado por un anillo metálico, el cual permite trabajar el sensor al mismo nivel que la superficie como se observa en la figura 2.6.

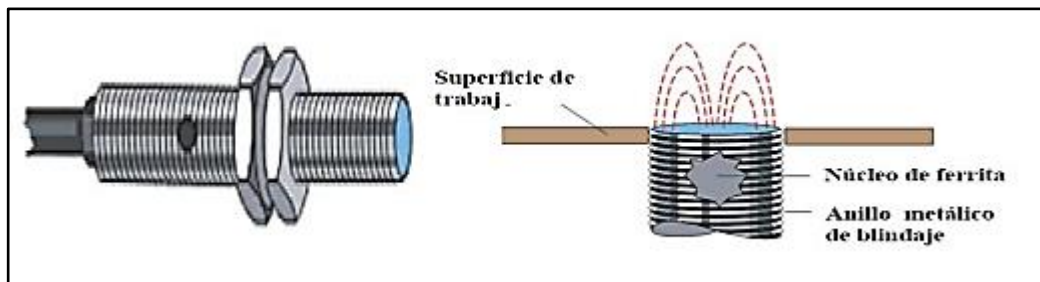


Figura 2.6 Sensor Inductivo Blindado

Fuente: (Astudillo, 2016)

Por otro lado, en la figura 2.7, el sensor no blindado necesita estar a una altura superior de la superficie ya que puede sufrir alteraciones en la detección de objetos. Sin embargo, la particularidad de no estar blindado lo hace un 50% más eficiente en la detección. (Castaño,2016)

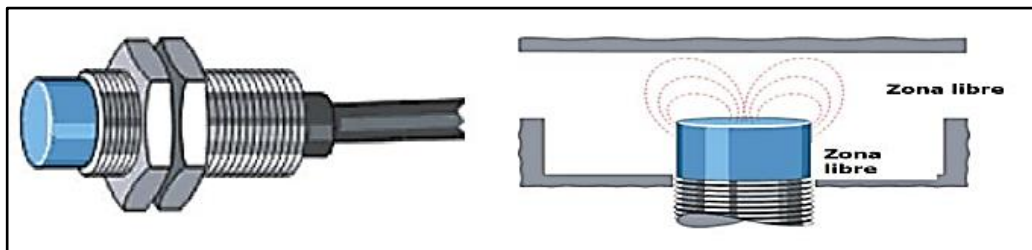


Figura 2.7 Sensor Inductivo no Blindado

Fuente: (Astudillo, 2016)

2.2.4 Sensores Ultrasonicos

Los sensores ultrasónicos tienen la capacidad de medir distancias entre dos puntos. Las ondas son emitidas a través de un transmisor y las ondas reflejadas son recibidas por un receptor. El transmisor convierte un voltaje en una onda sonora que se desplaza por el aire, esta es reflejada por una superficie que devuelve la onda para que el receptor pueda convertirla en voltaje. El material piezoeléctrico es deformado por la variación de tensión del entorno, por lo que produce una tensión. (Corrêa, 2021)

Existen tres tipos de configuración en los sensores ultrasónicos como se observa en la figura 2.8:

- La primera es el transmisor y receptor separado, con lo cual los sensores deben estar alineados cara a cara.
- La segunda es el transmisor y receptor lado a lado, lo cual permite que el receptor reciba la onda reflejada de la superficie.
- La tercera es el transmisor y receptor integrados en un solo transductor.

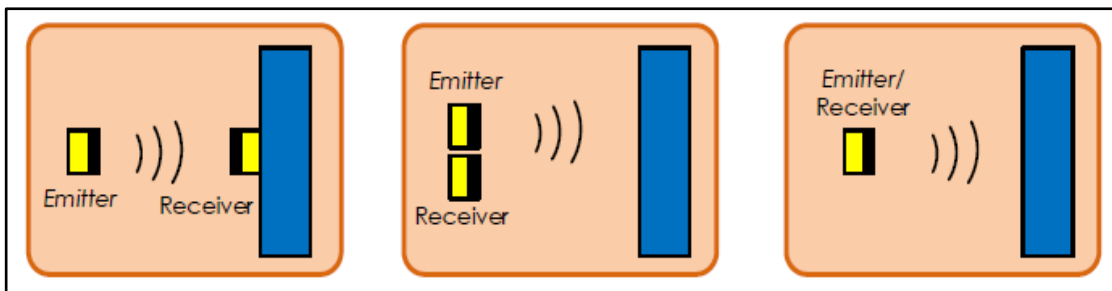


Figura 2.8 Tipos de sensores ultrasónicos

Fuente: (Muñoz & Lesmes, 2016)

2.2.5 Sensores Electroquímicos

Los sensores electroquímicos poseen la característica de reaccionar a ciertos componentes químicos para medir su concentración. De acuerdo con Chandra et al., (2019), La reacción química que genera este sensor está relacionada con la corriente eléctrica, ya que existe una combinación entre el transporte de la masa y la carga presente. Además posee tres tipos de derivación en los sensores electroquímicos los cuales son: sensores potenciométricos, amperométricos e impedimétricos o conductométricos. En la figura 2.9 se muestra un ejemplo de sensor electroquímico de gas.

Para Hering & Schönfelder (2022), los sensores de gas, al igual que otros sensores electroquímicos, realizan un calentamiento en su superficie catalítica conocido como micro calefactor. Una vez caliente, el gas presente en el ambiente se introduce en el sensor y forma una reacción química con la película del óxido semiconductor, lo cual produce un cambio en la resistividad del sensor. De esta manera, el sensor registra el cambio de la resistencia con respecto a la concentración del gas presente en su circuito de lectura.

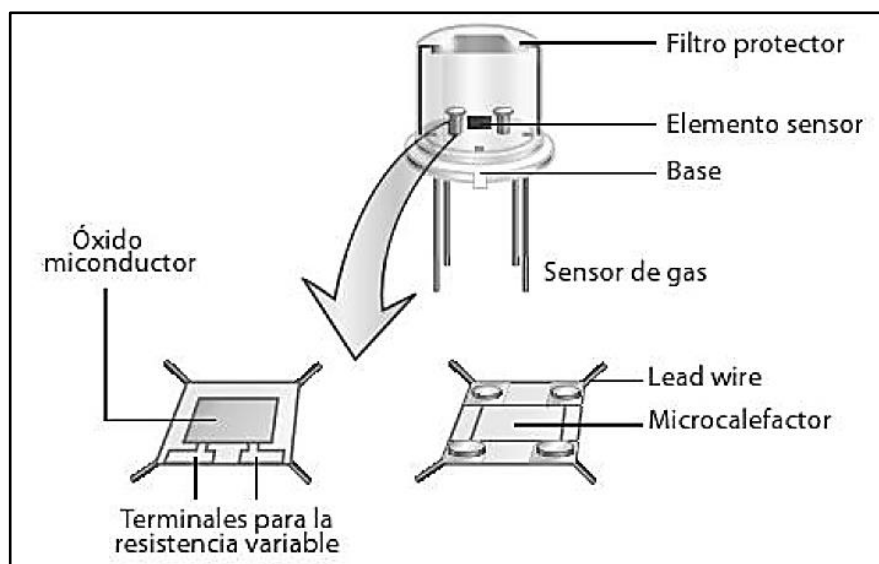


Figura 2.9 Estructura de un sensor electroquímico de gas

Fuente: (Corona et al., 2014)

2.2.6 Sensores Térmicos

Los sensores térmicos permiten registrar las señales térmicas en forma de cambios en la resistencia o voltaje de salida del componente en diferentes niveles de temperatura. Estos sensores cuentan con un alto nivel de precisión, por lo que son costosos y su calibración es necesaria de forma periódica. Según Lang (2021), este tipo de sensor se puede clasificar por sus tres efectos termoeléctricos en su fabricación; efecto Peltier, efecto Seebeck y efecto Thomson. Los tres efectos son producidos por la energía cinética en los electrones del componente, lo que resulta en una corriente termoeléctrica.

El efecto Peltier es el resultado de la unión de dos conductores de metal diferentes, a temperaturas distintas, en un circuito que absorbe o emite calor creando un valor proporcional a la corriente eléctrica que circula en dicho circuito hasta conseguir un equilibrio dinámico. El esquema del efecto Peltier se muestra en la figura 2.10. (Ida, 2020)



Figura 2.11 Efecto Peltier a distintas temperaturas

Fuente: (Corona et al., 2014)

El efecto Seebeck es producido por la unión de dos conductores diferentes que se encuentran a una misma temperatura utilizando una fuerza electromotriz para producir una corriente eléctrica en el circuito. Por lo tanto, si uno de los conductores cambia de temperatura, entonces la corriente eléctrica cambiará. La estructura de funcionamiento se presenta en la figura 2.11 (Hering & Schönfelder, 2022).

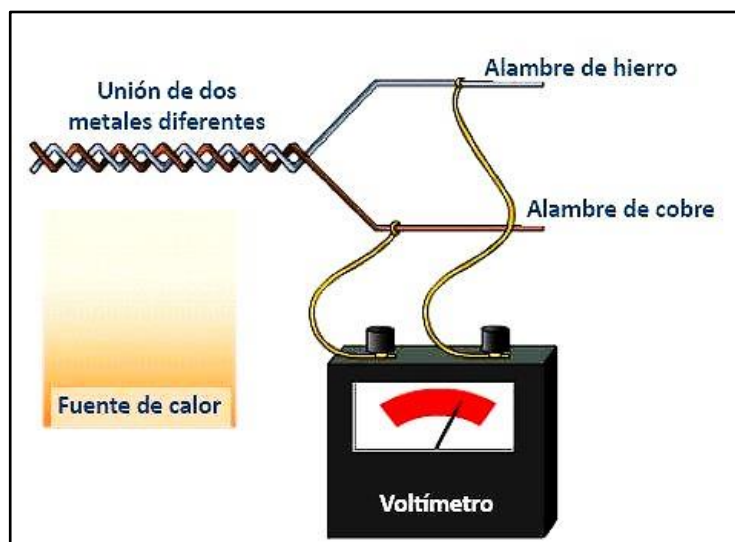


Figura 2.10 Efecto Seebeck en dos conductores metálicos

Fuente: (Astudillo, 2016)

El efecto Thomson establece que la generación o absorción de calor producido por la corriente eléctrica que circula a través del circuito está condicionada por el diferencial de temperatura en los extremos del conductor. El efecto de generación de calor ocurre cuando la corriente eléctrica fluye en sentido antihorario del reloj. Por el contrario, el efecto de absorción de calor se presenta cuando la corriente eléctrica fluye en sentido horario del reloj a través del circuito. Estos dos tipos de fenómenos se muestran en la figura 2.12 (Novák, 2020)

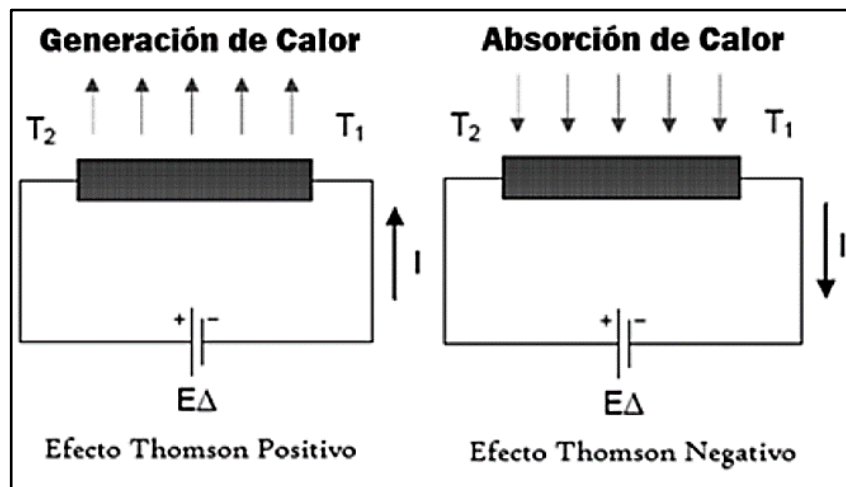


Figura 2.12 Efecto Peltier Generación y Absorción de calor
Fuente: (Corona et al., 2014)

2.2.7 Sensores Fotoeléctricos

Los sensores fotoeléctricos están constituidos como semiconductores, capaces de reaccionar a la luz o emitir luz. Posee la característica de interactuar con el rango visible o infrarrojo de la luz, siendo este ideal para aplicaciones industriales y ambientales. El principio de este sensor se basa en la interacción de un fotón que incide con una cantidad de energía a través de un material que requiere cierta cantidad de energía para enviar un electrón, puesto que realiza una conversión de luz a señal eléctrica. (Bernstein, 2022).

En la figura 2.13 se muestra un sensor de tipo reflexivo. Este sensor está conformado dos componentes; un emisor y receptor del mismo lado. El emisor emite un haz de luz hacia una superficie refleja, y posteriormente es reflejada al receptor, similar al principio de los sensores ultrasónicos.

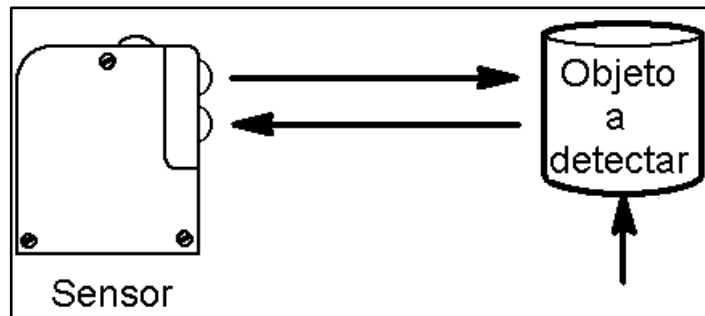


Figura 2.13 Sensor fotoeléctrico de tipo reflexivo

Fuente: (Muñoz & Lesmes, 2016)

2.2.8 Sensores Piezoeléctricos

Los sensores piezoeléctricos están diseñados para medir unidades acústicas, eléctricas y mecánicas. Su funcionamiento es posible por la interacción de los estados eléctricos y mecánicos que posee, y gracias a esto genera una carga eléctrica cuando se aplica una presión externa (Rupitsch, 2019). En detalle, cuando el material recibe una presión que lo deforma, se genera un cambio de polarización y esta magnitud física es convertida a una señal eléctrica. También existen los sensores piezoeléctricos que trabajan de manera inversa, es decir, que el material se encuentra en un estado de reposo y puede ser inducido con una corriente eléctrica para alterar su estado en contracción o expansión como se muestra en la figura 2.14. (Fericean, 2018)

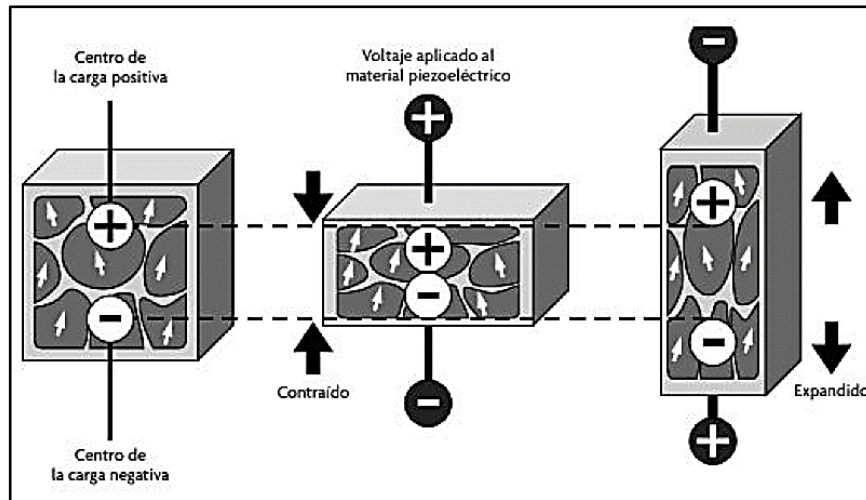


Figura 2.14 Efecto piezoeléctrico producido por una corriente eléctrica
Fuente: (Corona et al., 2014)

2.2.9 Sensores Piezorresistivos

Los sensores Piezorresistivos operan de forma similar que los sensores piezoeléctricos, sin embargo, la interacción se produce entre la resistencia eléctrica y el estado mecánico. La acción mecánica que se presenta es una presión que crea una elongación en el material. Esta elongación produce un cambio en la resistencia eléctrica en el material del sensor. (Lang, 2021)

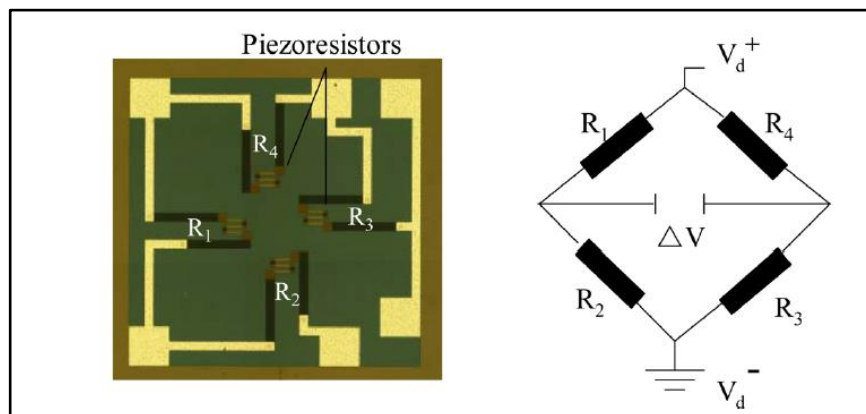


Figura 2.15 Estructura de un sensor Piezorresistivo(Izquierdo) y su diagrama eléctrico(Derecho)

Fuente: (Novák, 2020)

En la figura 2.15, se muestra la estructura de un sensor Piezorresistivo estructurados por una estructura mecánica con 4 galgas extensométricas que actúan como resistencias sobre un material de soporte que suele ser papel o poliamida, además de los elementos de fijación que son muy importantes en

la fabricación puesto que ayudan a obtener una correcta medición. (Chandra et al., 2019)

2.3 Redes de Sensores Inalámbricos

Dentro de la naturaleza de los sensores, existen la carencia de la propiedad inalámbrica. Esto demuestra que un sensor debe convertirse en un “sensor inteligente” para crear un enlace inalámbrico que permita el envío de datos de manera directa, ya que los sensores comunes no pueden por sí solos. Dependiendo de la necesidad de uso o aplicación, el sensor podrá crear un enlace de comunicación mediante microonda, radiofrecuencia o satélite. (Ida, 2020)

Los nodos de sensores están por uno o más dispositivos que conforman una variedad de medidas de propiedades del entorno. Singh et al. (2020), explican que una característica muy importante en su energía, pues son de bajo consumo y una desventaja de estos es su poca capacidad de memoria. Poseen un sistema complejo que les permite ser autónomo, adaptativo, operativo en altas densidades y de bajo costo de producción.

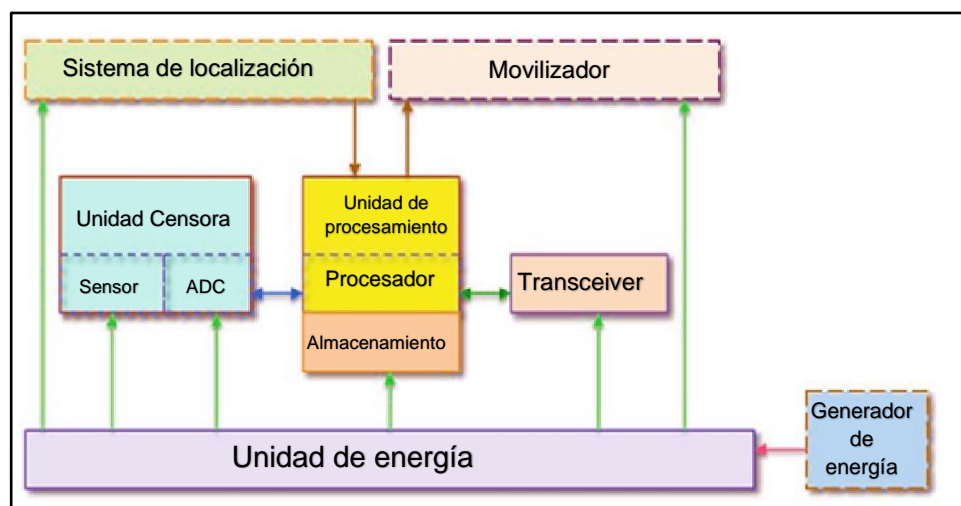


Figura 2.16 Estructura de un nodo de sensor
Fuente: Adaptado de Ahmad, 2021

En la figura 2.16 se muestra un diagrama de componente de un nodo de sensor. La unidad de energía está encargada de suministrar energía a los módulos del nodo. La unidad censora está constituida por los sensores del nodo y convertidores Analógico-Digital y viceversa. La unidad de procesamiento contiene el procesador central del nodo y, en ocasiones, una

pequeña unidad de almacenamiento para la gestión de tareas asignadas a otros módulos. La unidad transceptora o “Transceiver”, tiene como finalidad la conexión alámbrica o enlace inalámbrico a la red principal. De la misma forma, existen módulos complementarios que se utilizan para distintas aplicación como pueden ser los sistemas de localización para precisar georreferencias de alta precisión, movilizados que permiten ejecutar tareas de movimiento, también conocido como “actuadores” y sistemas alternos o de respaldo de energía como generadores o paneles solares.

El surgimiento de las redes de sensores inalámbricos, también conocidas en inglés como “Wireless Sensor Networks(WSN)”, son resultados del desarrollo tecnológico de procesadores, memorias, radio y componentes de bajo consumo. Estas redes son un conjunto de nodos de sensores dotados de capacidades para comunicarse de manera inalámbrica, censar el entorno y recolectar y enviar datos. Generalmente, poseen una estructura de red inalámbrica de topología red de área personal y son diseñadas con la finalidad de aplicaciones IoT y sistemas basados en web. (Gupta et al., 2021)

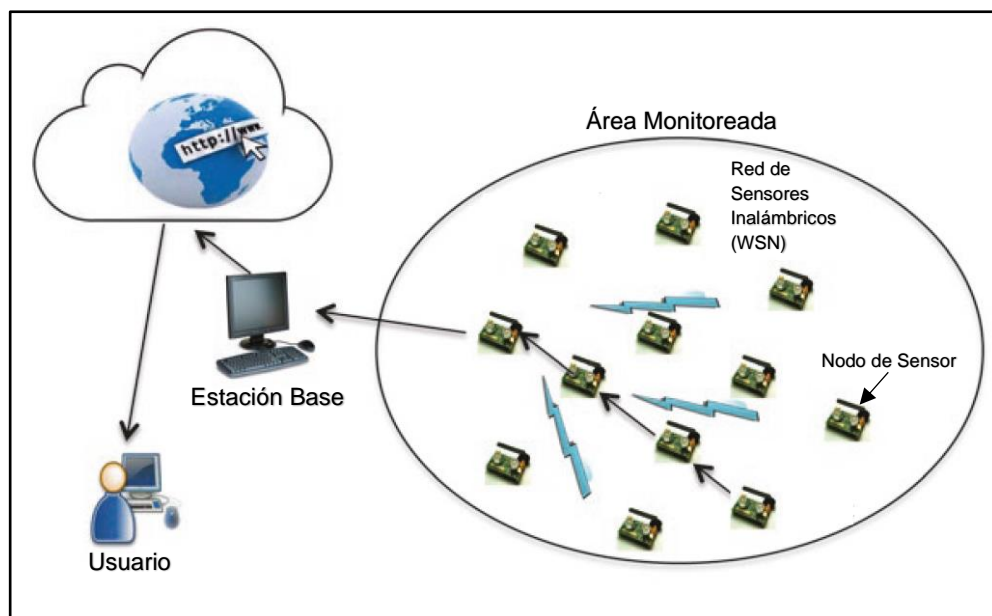


Figura 2.17 Estructura de una red de sensores inalámbricos

Fuente: Adaptado de Dumka et al., 2019

En la figura 2.17 se muestra un ejemplo de una arquitectura WSN. Comienza con un conjunto de nodo de sensores comunicados de forma inalámbrica, esta se denomina el área monitoreada. La gestión y recopilación de datos de los nodos está encargado por una estación base o servidor local,

además de establecer una comunicación a la internet o “nube”. El usuario que desea obtener los datos de las variables administradas deberá acceder por medio de un terminal con acceso a internet para establecer una comunicación remota con su estación base.

2.4 Estándares de las redes de sensores inalámbricos

De acuerdo con Singh et al. (2020), los estándares de la WSN son las mismas normas establecidas en la Unión Internacional de Telecomunicaciones(UIT), puesto que esta entidad regulatoria está encargada de normalizar el uso de espectro eléctrico, siendo las bandas de radio industrial, científica y médica(ISM), la primera norma en ser establecida en el año 1947. En la tabla 2.1, se enlistan las normas y sus características que han sido definidas para el uso en las redes de sensores inalámbricos basados en la normativa IEEE 802.15.x: Redes de área personal inalámbricas(WPAN´s).

Protocolo	Zigbee	Z-Wave	EnOcean	UWB	Wibree, BLE	Insteon	Wavenis
Estándar	IEEE 802.15.4	Propio	Propio	IEEE 802.15.4a	IEEE 802.15.1	Propio	Propio
Banda de frecuencia	868/915 MHz, 2.4 GHz	868/908 MHz	868 MHz	3.1-10.6 GHz	2.4 GHz	904 MHz	433/868/915 MHz
Velocidad de datos	20/40/250 Kbps	9.6/40/200 Kbps	120 Kbps	850 Mbps	1 Mbps	38.4 Kbps	19.2 Kbps
Modulación	BPSK/ BPSK/ O- QPSK	FSK /GFSK	ASK	BPSK, QPSK	GFSK	FSK	GFSK/PSK
Riesgo de interferencia	Muy alto - 2.4 GHz	Medio	Muy bajo	Muy bajo	Bajo	Medio	Bajo
Rango de comunicación	30 m interior, 100 m - 1.6 km exterior	30 m interior, 100 m exterior	30 m interior, 300 m exterior	10 m interior - exterior	10 m interior - exterior	50 m interior - exterior	200 m interior, 1 km exterior

Requerimiento de energía	Medio – 90 mW	Bajo – 30 mW	Muy Bajo – 15 mW	Alto – 140 mW	Muy Bajo – 15 mW	Medio – 75 mW	bajo – 30 mW Muy alto – 500 mW
Nodos en red	6500	232	2 ³²	8	8	256	Sin límites, teórico

Tabla 2.1 Especificaciones sobre los estándares basados en la normativa IEEE 802.15.x

Fuente: Adaptado de Ahmad (2021)

2.5 Topologías de redes

Las topologías de redes definen la arquitectura de los enlaces que permiten que los nodos estén interconectados. Estas topologías son esenciales para la descripción de los elementos que conforman la red, además de ofrecer un detalle físico para implementar técnicas que mejoren los problemas que se puedan presentar. Las topologías de red también son aplicables a las WSN, simplemente cambia el medio de transmisión físico a inalámbrico. Existen 5 modelos fundamentales dentro de las redes de datos; red en anillo, red en bus, red en estrella, red en árbol y red en malla. (Liberatori, 2018)

2.5.1 Topología de red en anillo

Para Kuzmiakova (2021), la topología en anillo presenta una red en bucle, donde el primer nodo estará conectado al último nodo, formando una red circular interconectada. Dentro de la red, existirá un nodo encargado de la gestión interna de la red. Esta configuración es muy útil para estructurar redes de nodos a largas distancias y provee redundancia en la red. Sin embargo, existe la posibilidad de una pérdida de comunicación si un nodo presenta problemas, además de la disminución de rendimiento con cada nodo en red.

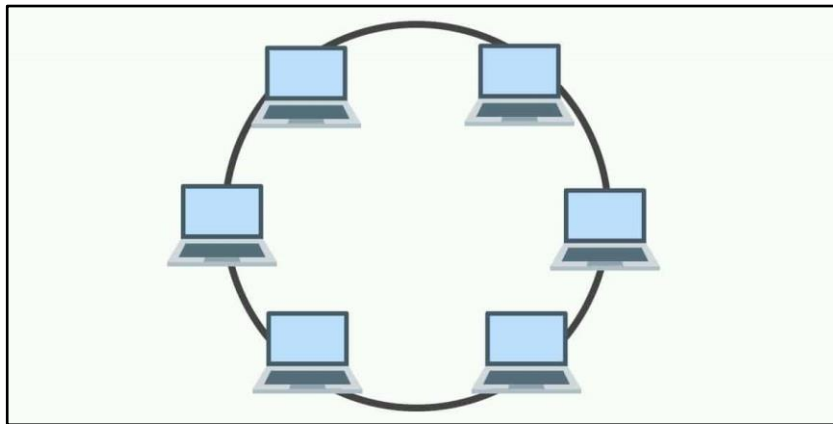


Figura 2.18 Topología de red en anillo

Fuente: (Kurose & Ross, 2021)

2.5.2 Topología de red en bus

La topología en bus está conformada por un enlace principal en donde los nodos estarán vinculados para formar parte de la red. Esta configuración solo permite que la comunicación fluya por una sola dirección, es decir, que si se producen dos comunicaciones al mismo tiempo, creará un conflicto en el flujo de la comunicación o un bloqueo. Para evitar estos inconvenientes, se utiliza en los extremos del cable principal un dispositivo denominado “terminador”, que destruye los datos que rebotan en el flujo de comunicación principal. Esta topología está obsoleta, ya que tiene muy bajo rendimiento en alto tráfico de datos, además de la comunicación en un solo sentido, por lo que se considera una red semidúplex. (Peterson & Davie, 2022)

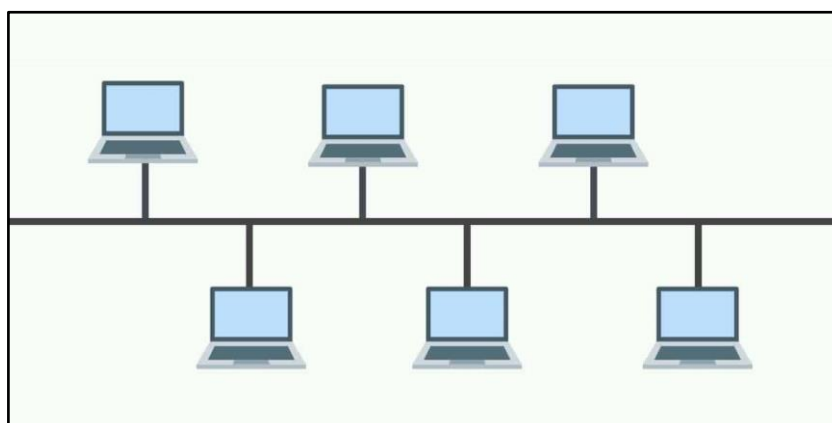


Figura 2.19 Topología de red en bus

Fuente: (Kurose & Ross, 2021)

2.5.3 Topología de red en estrella

En esta configuración, la gestión se centraliza en un nodo o concentrador principal. Los dispositivos o terminales se transforman en clientes porque la transacción del tráfico de datos tiene que pasar por el nodo principal. Esta topología es comúnmente usada debido a que el diseño de la red entrega un alto rendimiento, la distribución de carga de trabajo es dinámica y los fallos solo afectan a los terminales. (Kuzmiakova, 2021)

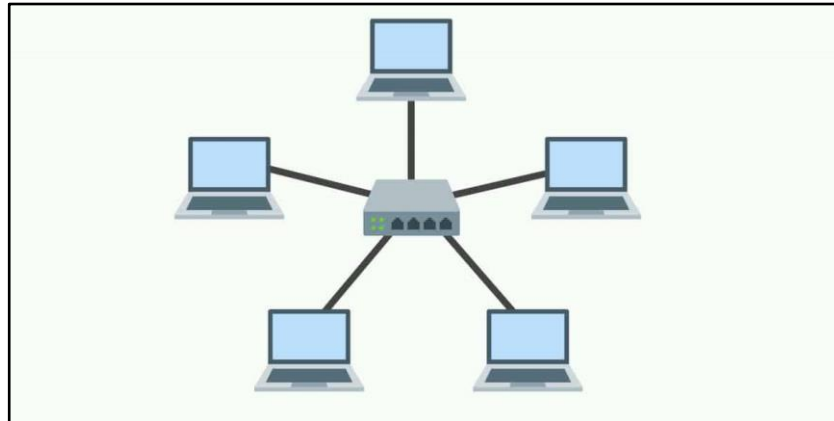


Figura 2.20 Topología de red en estrella
Fuente: (Kurose & Ross, 2021)

2.5.4 Topología de red en árbol

Lederkremer (2019) explica que la topología en árbol desarrolla un modelo jerárquico en forma de raíz donde existe un nodo predominante. Este encaminamiento permite un sistema heredado con relación maestro-esclavo en la red. La red puede ser expandible, por lo que permite crear sistemas complejos que detecten fallas de manera eficiente. Por el contrario, si el nodo principal falla, las ramas estarán subdivididas.

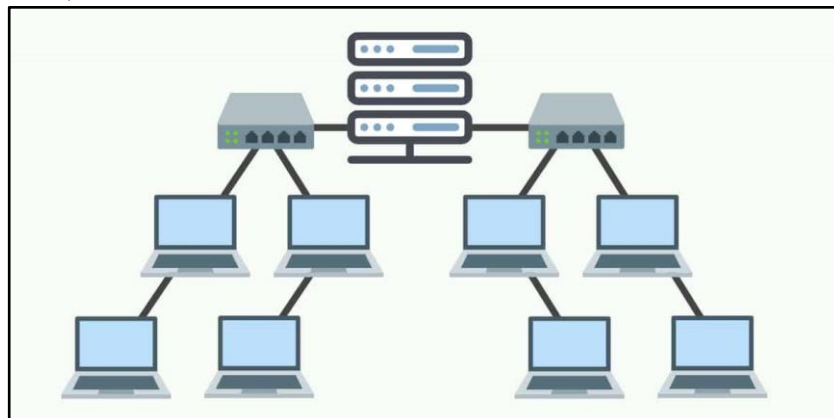


Figura 2.21 Topología de red en bus
Fuente: (Kurose & Ross, 2021)

2.5.5 Topología de red en malla

La topología en malla ofrece una red con nodos interconectados entre sí. Su amplia cantidad de nodos ofrece un alto rendimiento, ya que gestiona el tráfico de datos por medio de enrutadores lógicos que calculan distancias menores de viaje para el destino. Esta configuración punto a punto posee una alta tolerancia a las fallas gracias a la redundancia de conexiones, de tal manera que si el medio de transmisión falla, un nodo A y un nodo B, podrán comunicarse indirectamente por medio de un nodo C. La desventaja de este tipo de configuración reside en el costo de implementación, en el tiempo de configuración y mantenimiento de la red. (Liberatori, 2018)

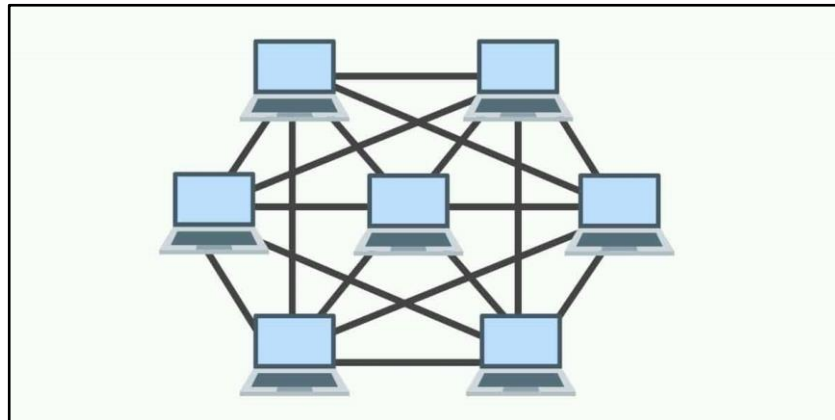


Figura 2.22 Topología de red en bus

Fuente: (Kurose & Ross, 2021)

2.6 Microcontroladores

Méndez (2021), explica que los microcontroladores son dispositivos que integran microcomponentes tales como: unidad de procesamiento, memoria de programa, memoria de datos y puertos de entrada y salida. Estos circuitos integrados tienen como finalidad la adquisición de datos(DAQ) y procesamiento de señales. El bajo costo de producción permite que cualquier persona pueda adquirir uno para el desarrollo y ejecución de programas realizadas por un usuario conteniendo acciones específicas de un proyecto.

2.6.1 Arduino

Es una compañía destinada al desarrollo de hardware libre a través de microcontroladores. Sus microcontroladores están diseñados bajo el nombre "Ametel", con un bajo consumo energético y de código abierto. Existen variedad

en los modelos de las tarjetas Arduino, pero su verdadero cambio reside en el microcontrolador que las compone, puesto que los demás componentes electrónicos forman una expansión que permite utilizar las entradas y salidas del microcontrolador de forma segura.

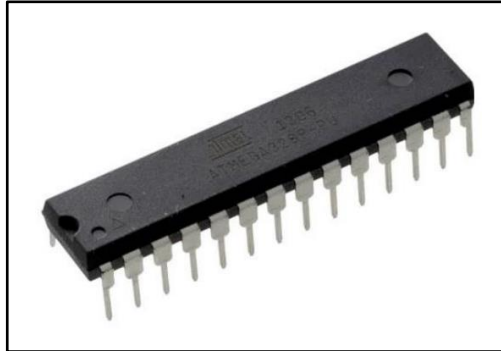


Figura 2.23 Encapsulado Atmega 328P

Fuente: (Méndez, 2021)

En la figura 2.23 se muestra el encapsulado Atmega 328, cuyo voltaje de operación se encuentra en el rango de 1.8 a 5.5 VDC. Cuenta con una arquitectura de 8 bits AVR, una memoria flash de 32 kB, una memoria RAM y EEPROM de 2 kB. Usualmente está destinados para proyectos de gestión IoT mediante el uso de sensores y actuadores. (Hidalgo, 2019)

2.6.2 PIC

De acuerdo con Muñiz (2021) el microcontrolador PIC es desarrollado por la empresa “Microchip”, cuyo objetivo es elevar el rendimiento con los módulos de comunicación de una red mediante protocolos avanzados. Su programación está basada en el lenguaje C, lo que permite que programadores en este lenguaje desarrollen instrucciones y soluciones de forma dinámica y sólida.

En la figura 2.24, se muestra el encapsulado 18f4550. Este encapsulado cuenta con 18 pines, ideales para utilizarse en proyectos de comunicación con Ethernet, Zigbee o USB. Su programación se realiza a través de un “Quemador” y un software de programación que compile archivos

con extensión “.hex”. De preferencia se recomienda utilizar el software “Mikro C”, pero se puede utilizar software de terceros.

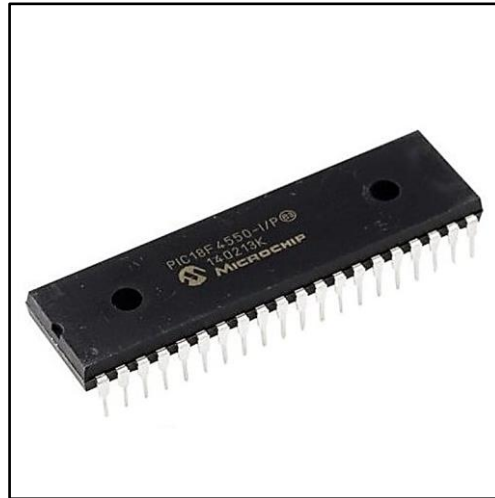


Figura 2.24 Encapsulado PIC18F4550

Fuente: (Muñiz, 2021)

2.7 Plataformas IoT

Estas plataformas son herramientas que facilitan la gestión de datos y proveen un control remoto detallado de la red del internet de las cosas. Su implementación se basa en la configuración de los parámetros y variables que deseamos monitorizar. El estilo de estas plataformas no solo se basa en la observación a través de sensores, sino en el control a través de actuadores para un mejor desempeño. (Velasquí & Fernández, 2020)

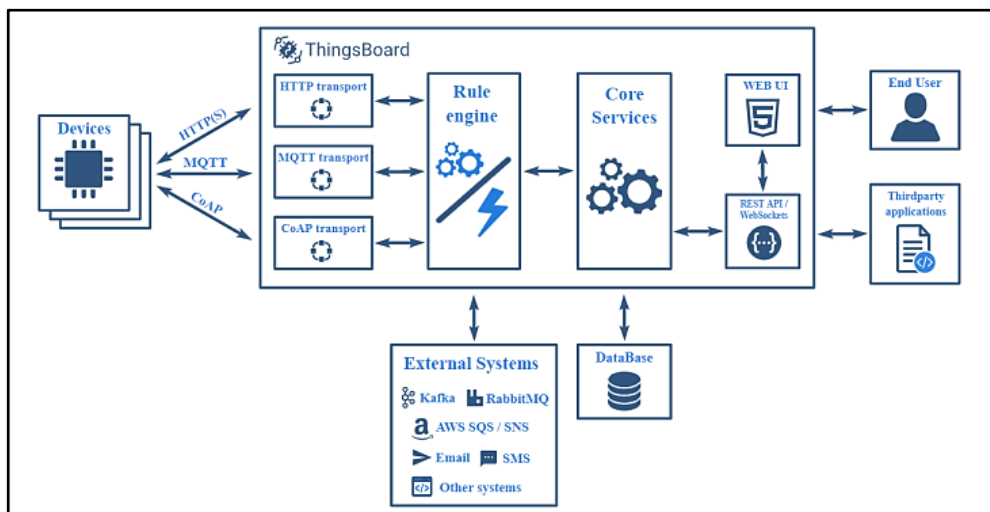


Figura 2.25 Arquitectura de la plataforma IoT ThingsBoard

Fuente: (Velasquí & Fernández, 2020)

En la figura 2.25 se muestra un ejemplo de arquitectura de una plataforma IoT. La mayoría de las plataformas dedicadas al IoT tienen semejanza a este tipo de modelo, por lo que, siempre existirá dispositivos de entrada que se comunicarán mediante protocolos de conexión gestionados independientemente por cada plataforma. Estas plataformas entregan variedades de funciones que permiten centralizar y gestionar los datos de una manera que puedan solventar las necesidades requeridas. Algunas cuentan con extensiones de terceros mediante APIs que permiten trabajar de manera conjuntas con otras plataformas o servicios remotos. También proveen una interfaz gráfica en el sitio web que mostrará al usuario la información traducida de su red en una Web UI personalizable, así como una API con la información de su interfaz para adaptarla a programas de terceros o propios. Este modelo resulta muy conveniente para el desarrollo de un entorno controlado al detalle y la variedad de herramientas permitirán al usuario administrador controlar procesos o variables de su red implementada.

2.7.1 Ubidots

Es una de las múltiples plataformas IoT que existen en la actualidad. Su servicio se caracteriza por entregar capacitaciones mediante prototipos basados en diferentes microcontroladores y módulos que se utilizan mayormente en producción. Cuenta con una amplia librería de dispositivos, variedad de herramientas de personalización visual y gestión de permisos. Su encriptación y cifrado de datos es robusta y su conexión remota se adapta a la mayoría de los sistemas operativos de escritorio y móviles. (Narváez & Contreras, 2020)

En la figura 2.26 se muestra la arquitectura de la plataforma Ubidots. Este tipo de arquitectura se basa en dos tipos de aplicación; desarrollo y producción. En la aplicación de desarrollo se implementa la configuración de los dispositivos y la traducción de datos. Los dispositivos se comunican con la nube mediante SDK y API desarrollados en su biblioteca con más de 200 dispositivos probados por la empresa. De la misma manera, cuenta con complementos para datos de terceros mediante entradas y salidas creadas en APIs personalizadas. El motor de análisis cuenta con dos etapas. La primera

etapa de variables sintéticas permite que los datos sean transformados en variables mediante expresiones estadísticas o matemáticas complejas, mientras que, la segunda etapa codifica las funciones en la nube para el posterior análisis de datos, siendo una parte fundamental el filtrado, limpieza, adición y optimización de datos para su extracción. En la administración de datos se consolida la información que será retenida en los servidores de acuerdo con el plan de pago que se escoja, por lo general, los planes de pago incluyen una retención de datos de 2 años y pueden ser utilizados para informar eventos, generar reportes o estadísticas. Por último, el “Dashboard” o interfaz gráfica permite una gestión visual a través de paneles o widgets que interpretarán las variables controladas, así como el control de acceso de los usuarios que utilicen dicha plataforma.

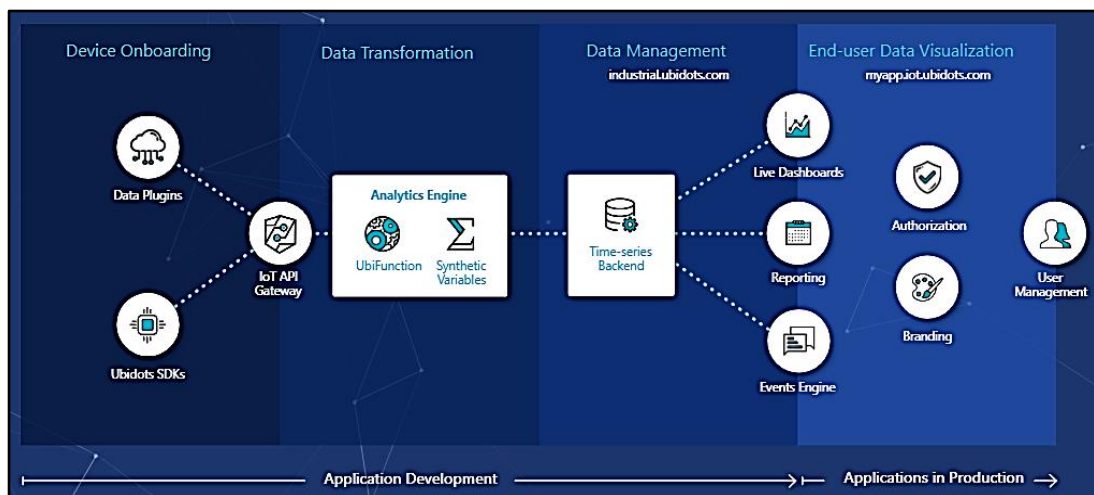


Figura 2.26 Arquitectura de la plataforma IoT Ubidots
Fuente: (Ubidots IoT Platform Features, s/f)

Capítulo 3: Diseño. Simulación y Resultados.

3.1 Selección de sensores para el cultivo de rosas

Tal y como se expuso en los antecedentes, el crecimiento y desarrollo de las rosas es gracias al cuidado de los factores ambientales del lugar donde se cultiva. Por lo tanto, hacer uso de los sensores para registrar cambios en las variables climáticas resulta favorable en la monitorización y corrección de dichos parámetros. Dentro del trabajo de investigación también se consideró variables del entorno comunes como: consumo en las tuberías de riego, calidad del aire, presencia de radiación UV, capacidad en tanques de líquido y capacidad en tanques de sólido. Para cumplir con el registro de cada parámetro, se hace uso de los siguientes sensores por cada variable climática establecida en la siguiente tabla 3.1.

Variable	Sensor	Fabricante	SKU de fabricante
Temperatura Y Humedad Ambiental	Sensor DHT-20	DFROBOT	SEN0497
Humedad del Suelo	Sensor Capacitivo Analógico		SEN0308
Presencia de CO2	Sensor CO2 Infrarrojo		SEN0220
Nivel de pH del Suelo	Sensor pH Analógico		SEN0169-V2
Iluminación	Sensor Luz Ambiental		SEN0390
Consumo de agua	Sensor YF-S201		SEN0217
Calidad del aire	Sensor ENS160		SEN0515
Radiación UV	Sensor UV Analógico		SEN0162
Tanque de Líquidos	Sensor Ultrasónico		SEN0208
Tanque de sólidos	Sensor Capacitivo sin contacto		SEN0204

Tabla 3. 1 Selección de sensores de acuerdo con las variables establecidas

Fuente: Autor

3.2 Estructura de la red basada en Arduino y xBee

En esta sección del presente trabajo de titulación se planteó la configuración paso a paso de la estructura de la red inalámbrica. Se comenzó con la configuración inicial de los microcontroladores. Para este caso, se consideró utilizar 4 microcontroladores; 3 Arduino UNO y 1 Arduino UNO WIFI. Del mismo modo, se definió para cada microcontrolador, un “Shield” o expansión que permitirá a cada microcontrolador albergar un dispositivo RF xBee, se utilizó 4 expansiones y módulos xBee en total. Cada microcontrolador actuó como un nodo que estará conformado por sensores que demostrarán el proceso de comunicación entre nodos.

La programación de cada nodo estuvo definida por la función de cada sensor presente, ya que los datos que se obtuvieron, fueron traducidos por cada nodo esclavo y enviados al nodo maestro, quien estuvo encargado de la comunicación entre la red interna y la plataforma de comunicación. Los módulos xBee son utilizados como un medio de enlace no guiado y la distancia de comunicación está definida por sus parámetros técnicos. Por último, en la alimentación de la red se consideró la adecuación de un módulo de transferencia de energía que permitió alimentar, mediante dos fuentes diferentes, a un regulador de voltaje de múltiples salidas que energiza a los microcontroladores y sensores en de la red.

3.2.1 Alimentación de la red de sensores inalámbricos

Para el diseño inicial, se consideró que la red fuese alimentada mediante dos fuentes de energía en corriente continua, con la finalidad de proveer redundancia y evitar la pérdida de registro datos y comunicación. En la figura 3.1, se observa un módulo interruptor automático de emergencia que permite la alimentación mediante dos entradas en corriente continua. La alimentación de este módulo permite un rango entre 12V a 48V y soporta hasta 10A. Para el presente trabajo de titulación se utilizó un voltaje de 12V

en la alimentación principal y de respaldo, de esta manera la carga emitirá un voltaje de 12V.



Figura 3. 1 Módulo interruptor automático de emergencia
Fuente: Autor

Para alimentar los diversos dispositivos de la red, se consideró el uso de un regulador de voltaje multicanal de 4 salidas; 3.3V, 5V, 12V y ajustable. Puede trabajar en un rango de voltaje entre 5V a 40V, con una corriente hasta 3A en un circuito o salida, y en la capacidad total de carga de corriente puede soportar hasta 5A. La terminal de voltaje ajustable ha sido establecida en 7V, que es el voltaje de operación recomendada para las tarjetas Arduino. En la figura 3.2 se muestra el regulador de voltaje con sus respectivas terminales de voltaje.

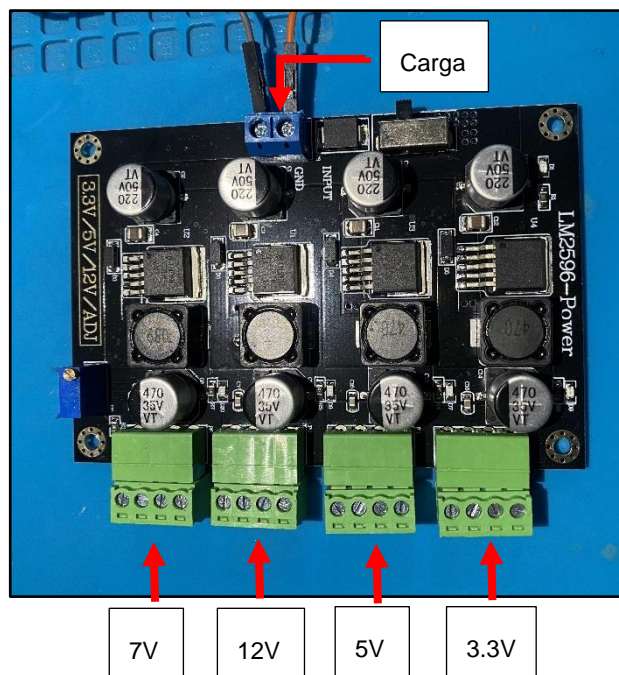
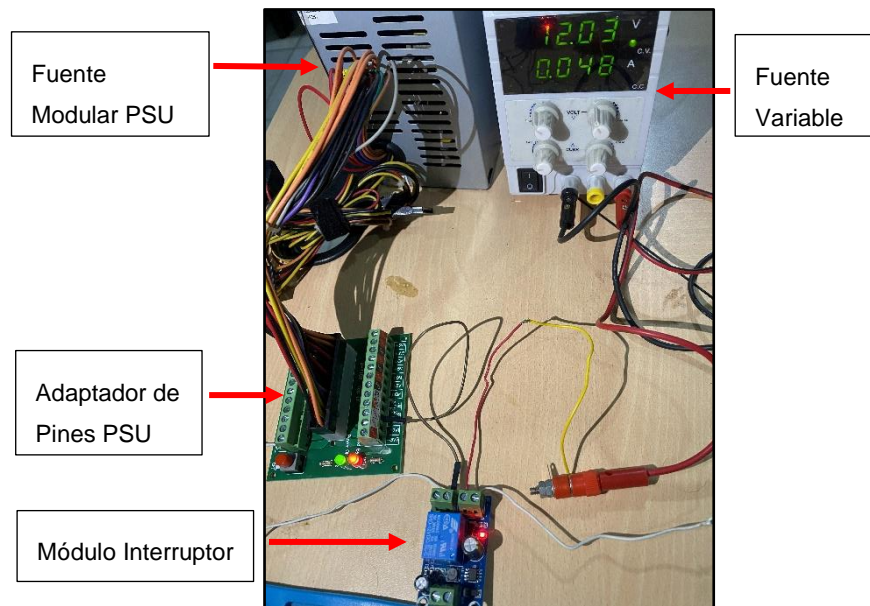


Figura 3. 2 Módulo regulador de voltaje multicanal
Fuente: Autor

Para el diseño de simulación de alimentación por medio de fuentes diferentes, se utilizó una fuente de alimentación variable y una fuente de alimentación modular. Ambas fuentes fueron definidas a 12V y fueron conectadas al módulo de interruptor automático de emergencia. El módulo interruptor entregó un voltaje de carga de 12V al regulador multicanal. En la figura 3.3, la fuente de alimentación variable simula la alimentación principal y la fuente modular la alimentación secundaria. El módulo interruptor indica a través de un led que la alimentación primaria se encuentra en funcionamiento y como consecuencia el regulador multicanal está operativa.



*Figura 3. 3 Conexión entre fuentes y modulo interruptor
Fuente: Autor*

En la figura 3.4, se observa que la luz guía del módulo interruptor está apagada, por lo tanto, no existe suministro en la alimentación primaria confirmada por la fuente variable apagada. Sin embargo, la fuente PSU de la entrada de respaldo sigue suministrando voltaje, y esto es confirmado por el funcionamiento del regulador multicanal. De esta manera se concluye que utilizando dos fuentes de energía; la red pública con un regulador de 12v y una fuente de energía alternativa como paneles solares o baterías o generadores, permiten que la red sigue operando a pesar de los problemas que se generen en la alimentación principal.

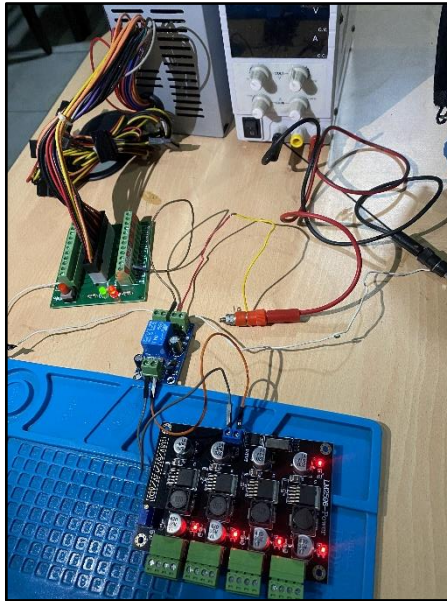


Figura 3. 4 Alimentación de respaldo en funcionamiento

Fuente: Autor

3.2.2 Configuración de los módulos de RF xBee como enlaces inalámbricos

La gestión de la comunicación entre nodos estuvo gestionada por los módulos de radiofrecuencias xBee. Estos módulos están basados en el estándar de redes de sensores inalámbricos Zigbee, cuya topología de red se basa en una malla, donde un nodo será en coordinador de la red, mientras que los demás serán nodos esclavos. Para configurar estos módulos, se hizo una conexión de los pines entre el módulo xBee y un microcontrolador Arduino Uno, siendo representados en la figura 3.5 y los terminales de conexión indicados en la tabla 3.2.

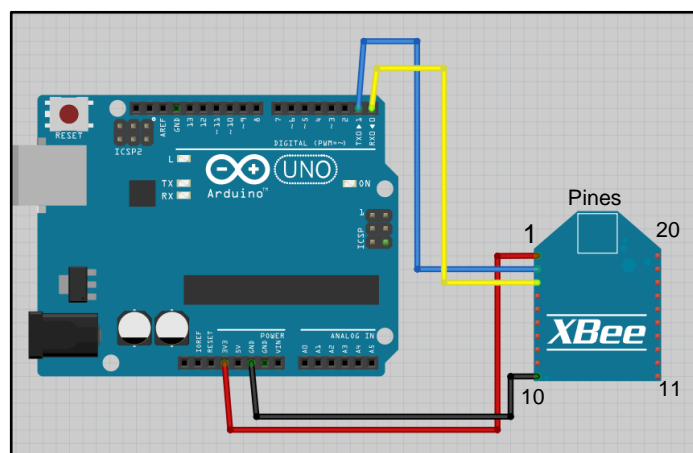


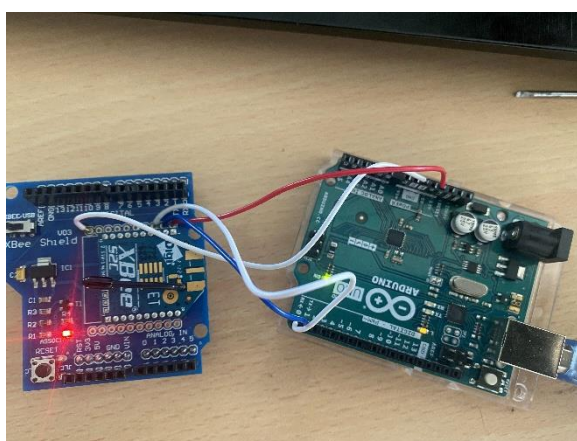
Figura 3. 5 Esquema gráfico de conexión entre Arduino y xBee

Fuente: Autor

Interconexión de Entradas/Salidas		
Pines del Arduino Uno	Pines del módulo xBee	Función
3.3V	1	Alimentación +
GND	10	Alimentación -
0D	3	Rx
1D	2	Tx

*Tabla 3. 2 Interconexión de pines entre microcontrolador y módulo RF
Fuente: Autor*

Este procedimiento se lo realizó en los Arduino Uno y Arduino Uno Wifi. Para el modelo físico se utilizó un adaptador de xBee con los pines correspondientes como se muestra en la figura 3.6. De esta manera, el microcontrolador sirvió como un bypass para la lectura del módulo en el ordenador. El microcontrolador recibe una programación de un sketch vacío para que el módulo obtenga acceso a través de los puertos seriales Rx y Tx.



*Figura 3. 6 Conexión física para la configuración del módulo
Fuente: Autor*

La configuración de los parámetros técnicos se realizó en el software XCTU, de la empresa DIGI, ya que es el software primordial de estos módulos RF. Se definió los mismos parámetros de la red inalámbrica para el nodo maestro y para los nodos esclavos. El único parámetro que varía entre los nodos es el “CE Coordinator Enable”, que indica el nodo asignado como maestro o coordinador. En la siguiente tabla 3.3, se encuentra establecido los parámetros de la red en detalle.

Configuración de la WSN		
Tipo XBee	Coordinador	Esclavo
ID Pan Id	2022	2022
CH Channel	10	10
JV Channel Verification	Enabled(1)	Enabled(1)
NI Node Identifier	COORD	GRUPO 1/2/3
AP API Enabled	TRANSP(0)	TRANSP(0)
PL Tx Power Level	Highest (4)	Highest (4)
PM Power Mode	Boost Enabled (1)	Boost Enabled (1)
BD Baud Rate	9600	9600

*Tabla 3. 3 Parámetros Técnicos de la red inalámbrica de sensores
Fuente: Autor*

En la figura 3.7 se observa la configuración de los módulos xBee como coordinador y grupos. En el software XCTU, se visualiza los 4 módulos conectados en serial a los microcontroladores Arduino. Posteriormente, está demostrado su operatividad en la red PAN del coordinador observando los nodos agrupados debajo del nodo "COORD". Estos nodos son los mismos ya que se pueden verificar por la dirección MAC de cada módulo.

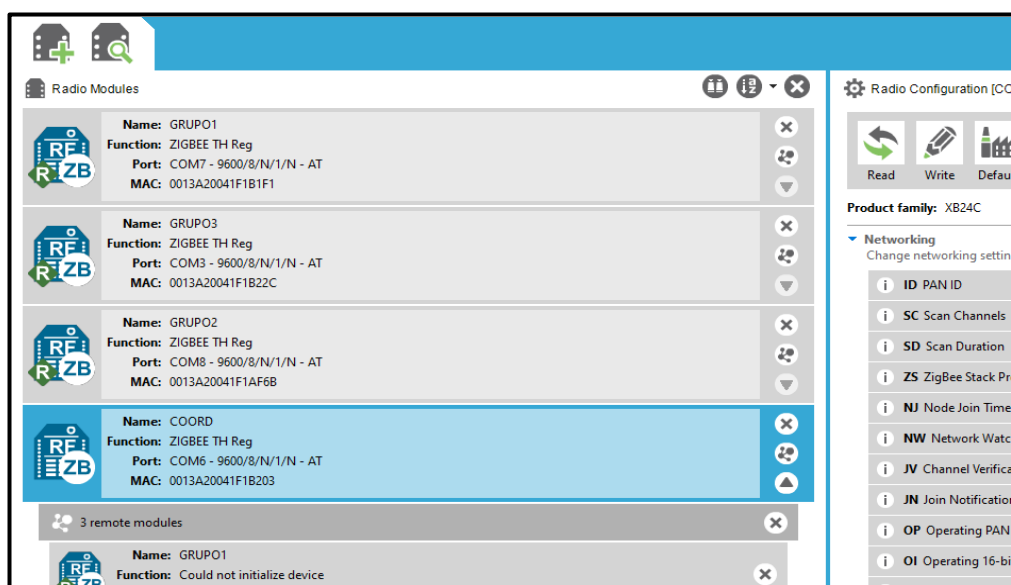


Figura 3. 7 Configuración y verificación mediante software XCTU

Fuente: Autor

3.2.3 Configuración del Arduino Uno como nodo esclavo

Se realizó el diseño de la red desde la parte interna de la red hasta la externa. Por ello, se definió en primera instancia a los microcontroladores que será los encargados de la recolección de datos de los sensores. Dos de los microcontroladores están conformados por 3 sensores y un microcontrolador por 4 sensores. Para el trabajo de titulación, se utilizó 3 microcontroladores Arduino Uno como se muestra en la figura 3.8.

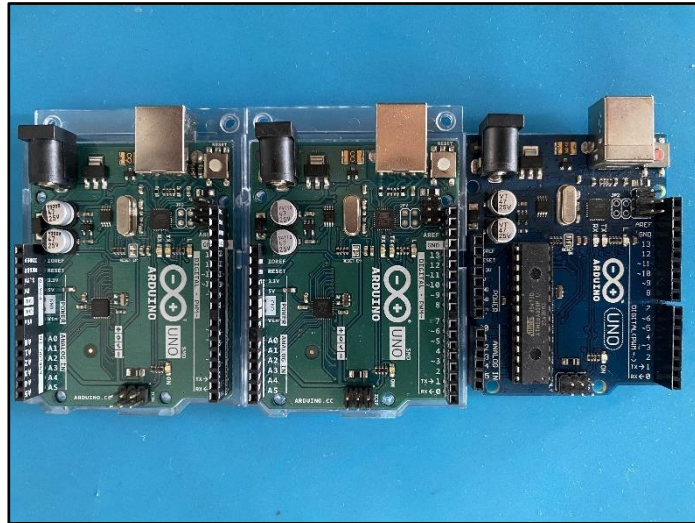


Figura 3. 8 Tarjetas Arduino Uno Rev.3

Fuente: Autor

Existen dos modelos de Arduino Uno; desmontable y SMD. La única diferencia que existe reside en su microcontrolador encapsulado, puesto que el SMD es un dispositivo de montaje superficial y el otro es un encapsulado desmontable para utilizar sin la tarjeta. Sus parámetros técnicos son exactamente iguales y se describen la siguiente tabla 3.4.

Especificaciones Técnicas	Arduino Uno/SMD
Microcontrolador presente	Atmega328P
Voltaje de Operación	5V
Voltaje de Entrada	7V – 12V
Voltaje Límite de Entrada	6V – 20V
Pines E/S Digitales	12 + 6 Salida PWM
Pines Entrada Analógicos	6

Corriente DC por Pin E/S	20 mA
Corriente DC para Pin 3.3V	50 mA
Memoria Flash	32 kB – 0.5 kB Bootloader
SRAM	2 kB
EEPROM	1 kB
Pin LED Incorporado	13
Velocidad del Reloj	16 MHz

Tabla 3. 4 Parámetros técnicos del Arduino UNO

Fuente: (Arduino, s/f)

El diagrama de pines de las tarjetas Arduino Uno son idénticas para la versión desmontable y SMD. Las salidas y algunos indicadores de la tarjeta se encuentran representadas en la figura 3.9.

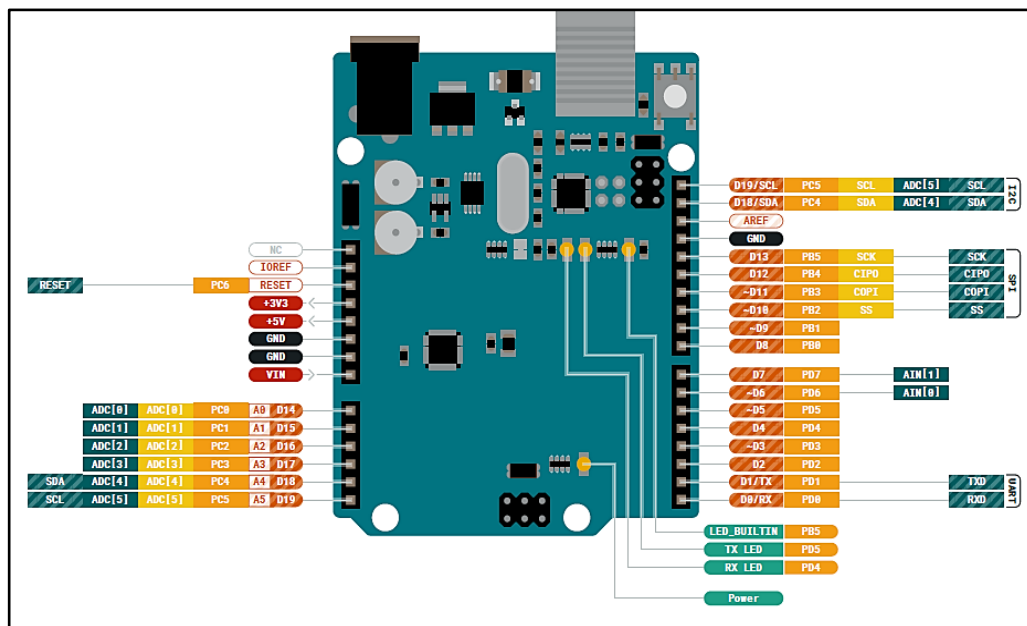


Figura 3. 9 Diagrama de entradas y salidas del Arduino Uno

Fuente: (Arduino, s/f)

La distribución de los sensores por cada microcontrolador para el presente trabajo fue definida en dos grupos de 3 sensores y un grupo de 4 sensores como se muestra en la tabla 3.5.

Microcontrolador	Sensor Asignado
Arduino Uno SMD Grupo 1	Sensor de Temperatura y Humedad
	Sensor de Humedad de suelo
	Sensor CO2
	Sensor Calidad del Aire
Arduino Uno SMD Grupo 2	Sensor pH
	Sensor UV
	Sensor Luz Ambiental
Arduino Uno Grupo 3	Sensor Flujo de Agua
	Sensor Ultrasónico
	Sensor Capacitivo

Tabla 3. 5 Asignación de sensores a microcontroladores

Fuente: Autor

La programación se realizó a través del software Arduino IDE, herramienta complementaria para la creación de instrucciones en los diversos modelos de tarjetas Arduino. La red de sensores interna conformada por 3 nodos esclavos consta de tres programaciones diferentes de acuerdo con las necesidades de cada sensor. Para cada microcontrolador, se implementó una tarjeta de expansión, con la finalidad de implementar un módulo RF que permita un enlace inalámbrico de comunicación. Además, esta expansión brinda una estructura de conexión mejor planteada ya que cuenta con pines de energización en paralelo con los pines nativos del microcontrolador Arduino. Esto supone un gran ahorro de espacio y recursos al no tener que implementar dispositivos electrónicos adicionales. La expansión del Arduino se muestra en la figura 3.10.

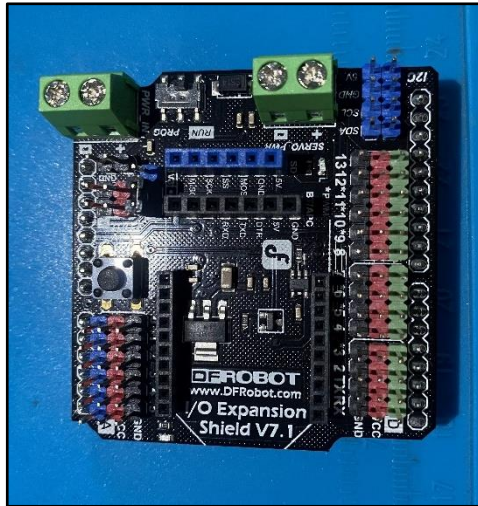


Figura 3. 11 I/O Expansion Shield V7.1, de la compañía DFROBOT

Fuente: (Arduino, s/f)

Para el primer grupo se realizó el archivo de programación en el anexo 1, basados en los sensores de temperatura y humedad del ambiente, humedad de suelo, calidad del aire y presencia de CO₂. Cabe destacar que el sensor de humedad de suelo necesita calibrarse en función de la temperatura ambiente para obtener un valor analógico preciso con la programación. Del mismo modo, el sensor de calidad del aire necesita la temperatura y humedad relativa del entorno para lograr una mayor precisión. La conexión de los sensores físicos se muestra en la figura 3.11.

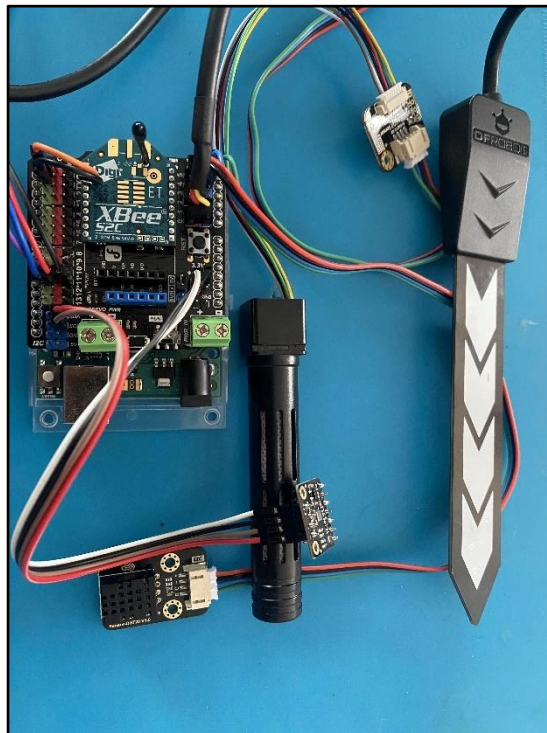


Figura 3. 10 Conexión entre 4 sensores y el primer nodo mediante Shield.

Fuente: Autor

Para el segundo grupo que conforma un nodo de la red, se utilizó los sensores de luz ambiental, pH y sensor UV. Estos sensores actúan en beneficio de la planta a través de un control de datos precisos de las variables escogidas. Su programación se muestra en el anexo 2. Los sensores son conectados a través de una tarjetas de expansión que permiten el enlace de los pines de los sensores y estas tarjetas se encuentran en todos los nodos de la red como se observa en a figura 3.12.

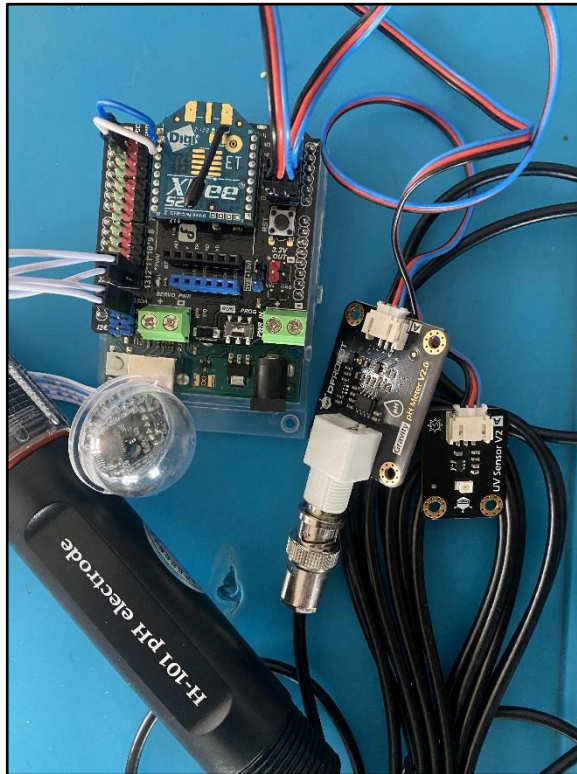


Figura 3. 12 Conexión entre 3 sensores y el segundo nodo mediante Shield
Fuente: Autor

Para el tercer grupo que conforma la red, se realizó el archivo de programación en el anexo 3, basados en los sensores de flujo de agua, ultrasónico y capacitivo. Estos tienen la finalidad de representar un apoyo general en diferentes funciones que se realizan en el invernadero y como intervinientes indirectos en el proceso de desarrollo. En este nodo no se presenta ningún dispositivo sensor que requiera de una calibración. En la figura 3.13 se observa el nodo físico constituido por los elementos sensores y el módulo RF.

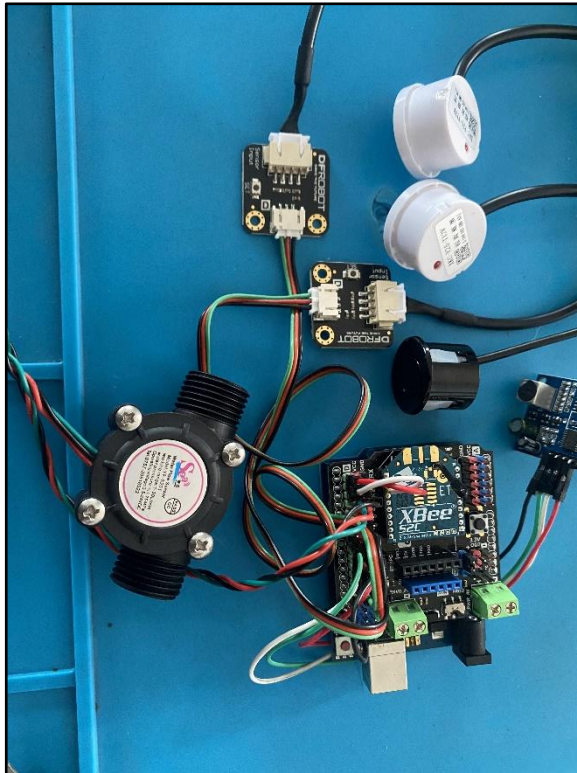


Figura 3. 14 Conexión entre 4 sensores y el tercer nodo mediante Shield
Fuente: Autor

3.2.4 Configuración del Arduino Uno Wifi como nodo maestro

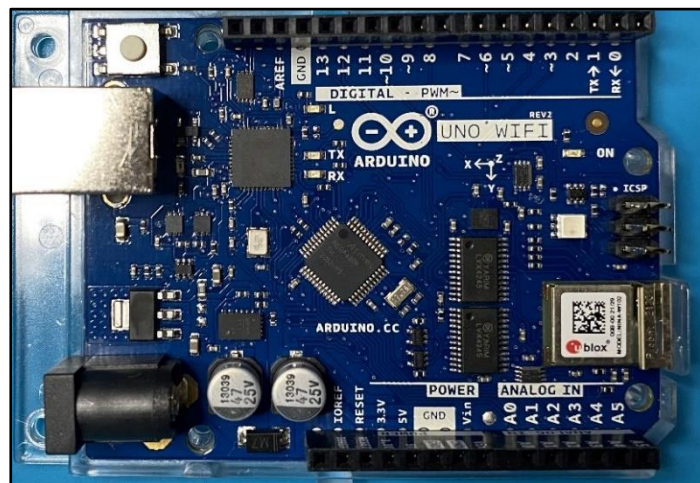


Figura 3.13 Tarjeta Arduino Uno Wifi Rev.2

Fuente: Autor

Para el diseño inicial se utilizó un Arduino Uno Wifi como microcontrolador principal mostrado en la figura 3.14, el cual se encargará de recibir los datos de los demás nodos para su comunicación entre los sensores y la interfaz IoT web. La diferencia entre este Arduino y los demás es que incorpora en su tarjeta un SoC denominado “u-blox NINA-W102”, el cual

permite utilizar redes wifi con una configuración directa a la red wifi de un router, y de esta manera se obtiene un acceso a internet sin necesidad de un ordenador o cable físico.

Antes de programar el microcontrolador, es necesaria la instalación de dos librerías específicas en el programa Arduino IDE para reconocer la tarjeta y utilizar su función wifi. La librería “Arduino megaAVR Boards”, ubicada en el gestor de tarjetas como se observa en la figura 3.15, está destinada reconocimiento y librería de la tarjeta. La librería “WiFiNINA”, ubicada en el gestor de librerías como se observa en la figura 3.16, está orientada al acceso a internet a través del wifi abierto o encriptado.

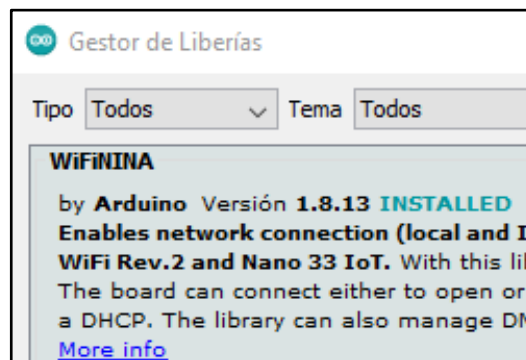


Figura 3. 16 Librería “Arduino AVR Boards”

Fuente: Autor



Figura 3. 15 Librería “Wi-FiNINA”

Fuente: Autor

Se realiza el chequeo del firmware de la placa antes de su programación a través del proyecto “CheckFirmwareVersion” de la librería “Wi-FiNINA” como se muestra en la figura 3.17. Esto tiene como objetivo evitar fallas o experimentar problemas durante la ejecución del programa que se asigne.


```
CheckFirmwareVersion Arduino 1.8.16
Archivo Editar Programa Herramientas Ayuda

CheckFirmwareVersion
*/
#include <SPI.h>
#include <WiFiNINA.h>

void setup() {
  // Initialize serial
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // Print a welcome message
  Serial.println("WiFiNINA firmware check.");
  Serial.println();

  // check for the WiFi module:
  if (WiFi.status() == WL_NO_MODULE) {
    Serial.println("Communication with WiFi module failed!");
    // don't continue
  }
}

Subido
This behaviour is deprecated and will result in an error in future version
You probably want to use 0xcd instead of 0xc9 (double check with your datasheet)
1 Arduino Uno WiFi Rev2, Nene (ATMEGA4809) en COM6
```

Figura 3. 18 Proyecto “CheckFirmwareVersion”

Fuente: Autor

Si la placa no se encuentra actualizada, resultará en un mensaje “NOT PASSED” como se muestra en la figura 3.18. Posteriormente, se tiene que cargar el código del proyecto “FirmwareUpdater” de la librería “WiFiNINA” para actualizar la placa como se muestra en la figura 3.19.

```
COM6

WiFiNINA firmware check.

Firmware version installed: 1.3.0
Latest firmware version available : 1.4.8

Check result: NOT PASSED
- The firmware version on the module does not match the
  version required by the library, you may experience
  issues or failures.
```

Figura 3. 17 Mensaje de placa desactualizada

Fuente: Autor

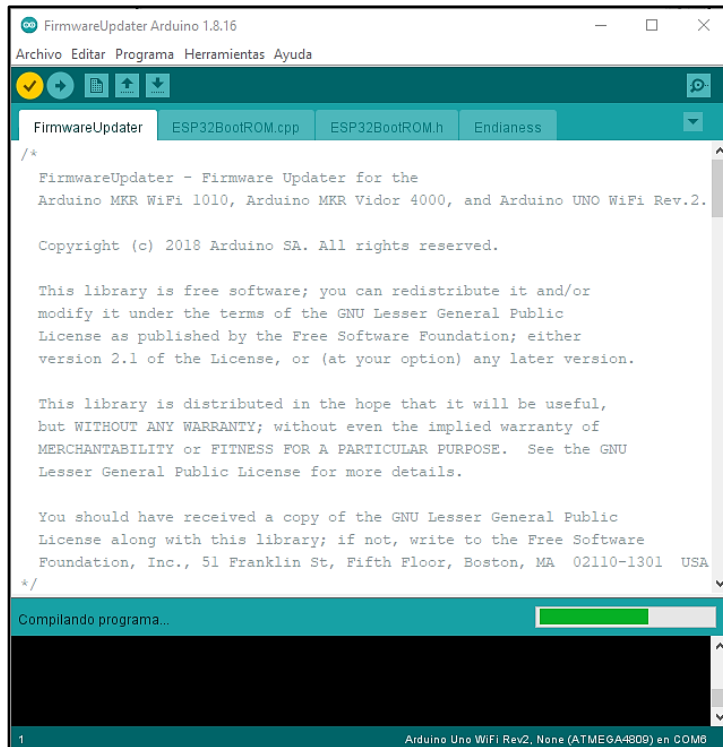


Figura 3. 20 Proyecto “FirmwareUpdater” y su compilación

Fuente: Autor

En las Herramientas de la aplicación Arduino IDE, se encuentra la opción “WiFinINA Firmware Updater”, con el cual la placa Arduino actualizará su firmware mediante la selección del puerto correspondiente y la segunda opción “Update Firmware”, como se muestra en la figura 3.20.

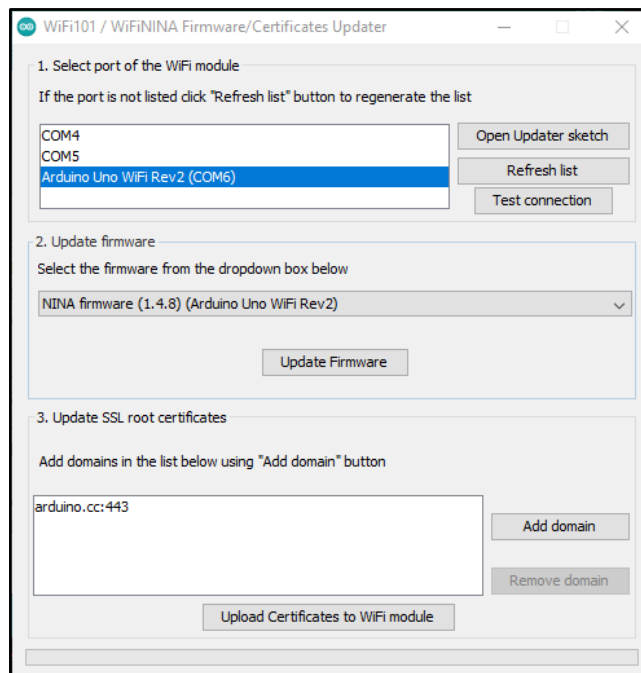


Figura 3. 19 Interfaz para actualizar firmware de la placa

Fuente: Autor

Si la placa se encuentra actualizada, se puede realizar de forma opcional un sketch personalizado de comunicación entre nodos o con la plataforma IoT o directamente en la programación de los nodos. En este trabajo se definió la programación del nodo coordinador en el anexo 4 en función de la recolección de datos de los nodos esclavos. En la figura 3.21 se muestra el diseño físico de conexión del nodo coordinador.

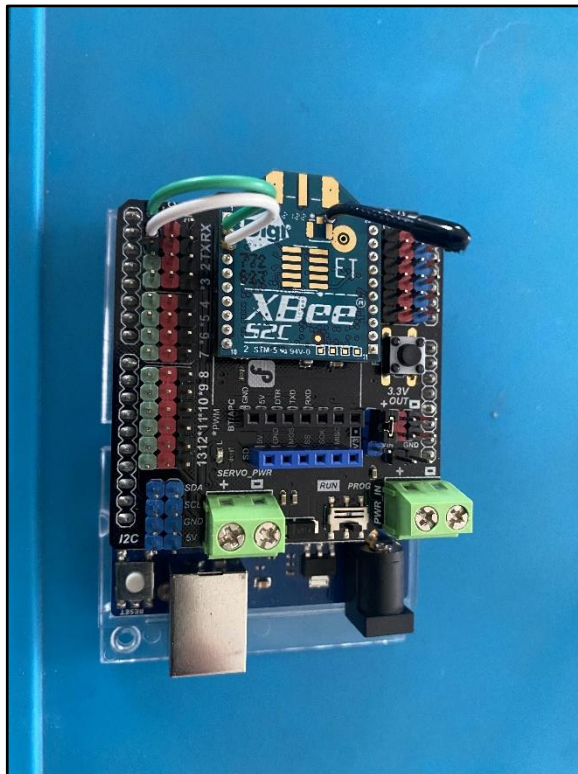


Figura 3. 21 Interfaz para actualizar firmware de la placa
Fuente: Autor

La característica más importante que resalta en este nodo es la carencia de dispositivos sensores que trabajen directamente, además del módulo RF. Esto se debe a que este nodo coordinador tuvo la tarea exclusiva de recibir datos mediante la transmisión y recepción de datos seriales. Los cuales almacenó e interpretó de tal forma fueron enviados a la plataforma de la nube. Adicionalmente, el microcontrolador fue programado para obtener un Gateway mediante un enlace con el router del ISP gracias a su módulo wifi integrado.

3.2.5 Diseño de la red de sensores inalámbricos

El diseño de la red conformado por los 4 nodos. Este modelo entrega facilidad en la gestión de información por un medio de transmisión no guiado.

De esta forma, se evita una arquitectura dependiente de un medio físico que requiere instalación y mantenimiento. El alcance de un nodo a otro puede variar por su configuración. Sin embargo, la verdadera limitante reside en los parámetros técnicos del módulo xBee mostrados en la tabla 3.6. Cabe destacar que existen varios modelos xBee, siendo la versión pro, el mejor modulo con un alcance significativo en comparación con la versión estándar, pero la característica de una mejor potencia conlleva a una mayor corriente de transmisión. La configuración que se utilizó es una red Zigbee, pero no se consideró que los datos viajen entre Routers para llegar al coordinador. El diseño se limitó al coordinador como único receptor de información.

Especificaciones	xBee S2C
Alcance interno o urbano	60 metros
Alcance línea de vista	1.200 metros
Potencia de transmisión	6.3mW Boost Mode = 8 dBm 3.1mW Modo normal = 5dBm
Tasa de datos RF	250.000 bps
Voltaje de operación	2.1V – 3.6V
Corriente de transmisión	45mA Boost mode 33mA Modo normal
Banda de frecuencia	ISM 2.4GHz – 2.5GHz
Rango de temperatura	-40°C a 85°C

Tabla 3. 6 Especificaciones técnicas de los módulos RF

Fuente: Autor

En la figura 3.22 se presenta un diagrama de conexión general del diseño de red. En el diagrama se puede observar que la conexión del nodo coordinador necesita una conexión a internet a través de un router de ISP. En su programación posee una conexión de la red wifi con los datos del SSID y el PASSWORD de la red. Adicional a esto integra la capacidad de recibir cadena de datos en diferentes tiempo gracias al Delay que incorpora en la programación los nodos esclavos. De esta manera leerá de manera secuencial los paquetes conformen llegan a través de la conexión serial inalámbrica y desglosa la cadena de datos a través de identificadores que ayudan a correlacionar los valores con las magnitudes correctas en el coordinador. La conexión es unilateral, no obstante, posee la capacidad de

interactuar entre todos los nodos si se activa la opción. Dicho de otro modo, la red permite escalar de tal forma que se pueden integrar actuadores que realicen tareas definidas, y el coordinador podrá emitir valores desde la plataforma IoT hacia la red interna.

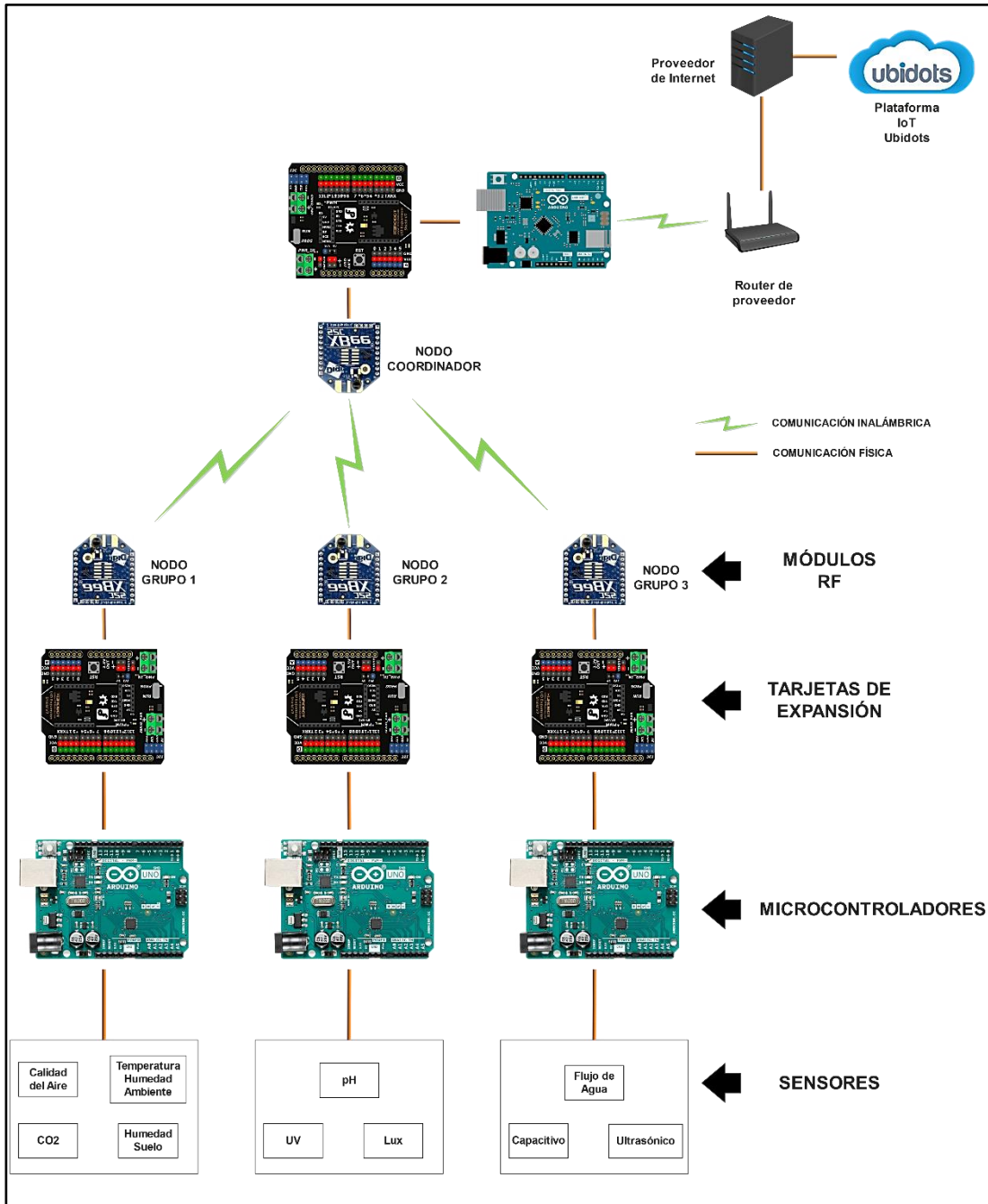


Figura 3. 22 Arquitectura de la red de sensores inalámbricos
Fuente: Autor

En la tabla 3.7 se incorpora los datos de conexión que se usaron para la comunicación entre sensores, módulos RF y microcontroladores. Estas entradas y salidas están relacionadas con las programaciones realizadas en

cada nodo. Cabe destacar que para la alimentación de los sensores, la tarjeta de expansión provee estos pines, por lo que no serán considerados en la tabla.

Sensores	Pines Sensores	Grupo Arduino	Pines Arduino
Sensor de Temperatura y Humedad	C	Grupo 1	A5
	D		A4
Sensor de Humedad de suelo	Amarillo		A0
Sensor CO2	Tx		D11
	Rx		D10
Sensor Calidad del Aire	SCL		SCL
	SDA		SDA
Sensor pH	A		Grupo 2
Sensor UV	A	A0	
Sensor Luz Ambiental	SCL	SCL	
	SDA	SDA	
	EN	5V	
Sensor Flujo de Agua	Verde	Grupo 3	D2
Sensor Ultrasónico	Trigger		D8
	Echo		D7
Sensor Capacitivo	Verde		D11 y D12

*Tabla 3. 7 Conexiones de pines para los sensores según la programación
Fuente: Autor*

3.3 Comunicación entre la red de sensores inalámbricos y la plataforma IoT Ubidots

Como último paso que culmina el desarrollo de la arquitectura de red inalámbrica se encuentra el servidor, donde se gestionan los datos que entregue la red. En este caso, el servidor es una plataforma web que integra herramientas orientadas a la gestión de datos del internet de las cosas. Para este caso se decidió la comunicación directamente entre el nodo coordinador y la plataforma mediante una puerta de enlace que enviará una estructura de mensaje en formato .JSON, requerido por la plataforma para la obtención y envío de valores.

En primer lugar, se debe contar con una suscripción para evitar la restricción de 5 variables de comunicación. Una vez suscritos, se crea un entorno o “dispositivo” en la plataforma, donde su función será el almacenamiento de las variables y datos que entregue el nodo coordinador. En la figura 3.23 se observa la creación del dispositivo con la etiqueta “Sensores_Agricultura”.

	NOMBRE	ÚLTIMA ACTIVIDAD	CREADO EN ↓	ORGANIZACIÓN	ACCIONES
	Sensores_Agricultura	6 hours ago	2022-08-31 19:53:07 -05:00	—	

Figura 3. 23 Creación del dispositivo en la plataforma Ubidots

Fuente: Autor

En la figura 3.24 se observa que las variables ya se encuentran creadas y almacenadas en el dispositivo, con un total de 13 valores utilizados en el trabajo. Esto se realizó por medio de un “Token” entregado a la plataforma como llave de comunicación entre la red y el dispositivo de la nube y es de uso privado para cada usuario. Fue utilizado en la programación para acceder a la comunicación respectiva.

Variable	Valor	Etiqueta	Última Actividad
cap_high	1.00	cap_high	6 hours ago
cap_low	1.00	cap_low	6 hours ago
cbok_carb	450.00	cbok_carb	6 hours ago
flow_agua	4.12	flow_agua	6 hours ago
gases_vic	15.00	gases_vic	6 hours ago
hum_rel	92.05	hum_rel	6 hours ago
hum_suel	35.00	hum_suel	6 hours ago
lx_amb	852.00	lx_amb	6 hours ago
ph_suel	8.00	ph_suel	6 hours ago
temp_amb	28.53	temp_amb	6 hours ago
uv_dcd	100.00	uv_dcd	6 hours ago
uv_amb	3.00	uv_amb	6 hours ago
vol_l	21.00	vol_l	

Figura 3. 24 Creación del dispositivo en la plataforma Ubidots

Fuente: Autor

De la misma manera se realizó la declaración de variables en el sketch del nodo coordinador para evitar conflictos o pérdida de comunicación al enviar valores hacia la plataforma. Las variables mostradas en la figura 3.24 pueden ser afirmadas por la declaración del sketch realizado en la figura 3.25.

```
Coordinador
#include <SPI.h>
#include <WiFiNINA.h>
#define DEVICE_LABEL "sensores_agricultura"
#define TOKEN "BBFF"

//Configuración de Arduino
const int HTTPPORT = 443;
char const *AGENT = "Arduino Wifi";
char const *HTTP_VERSION = " HTTP/1.1\r\n";
char const *VERSION = "1.0";
char const *PATH = "/api/v1.6/devices/";
//Configuración wifi
char const * SSID_NAME = "NetQuantum";
char const * SSID_PASS = "D0ntTurn0ff.Pls!";
int status = WL_IDLE_STATUS;

char const *SERVER = "industrial.api.ubidots.com";
WiFiSSLClient client;

//Configuración de variables
char const * VOC = "Gases_VOC";
char const * CO2 = "Diox_Carb";
char const * Temp = "Temp_Amb";
char const * HumR = "Hum_Rel";
char const * HumS = "Hum_Suel";
char const * pH = "pH_Suel";
char const * LX = "LX_Amb";
char const * UV = "UV_Amb";
char const * Flow = "Flow_Agua";
char const * VOL = "Vol_L";
char const * CapH = "Cap_High";
char const * CapL = "Cap_Low";
char const * Dist = "Ult_Dist";
```

Figura 3. 25 Sketch programado en Arduino Wifi Rev.2

Fuente: Autor

Estas variables son destinadas al uso de la representación de los valores en una interfaz gráfica que se debe crear. Esto facilita el conocimiento de la información a través de la interpretación estética y dinámica. Para este caso se diseñó una interfaz que permita observar mediante el uso de registros en gráficas, contadores, medidores de varios tipos y representaciones de estados en digital como se observa en la figura 3.26.

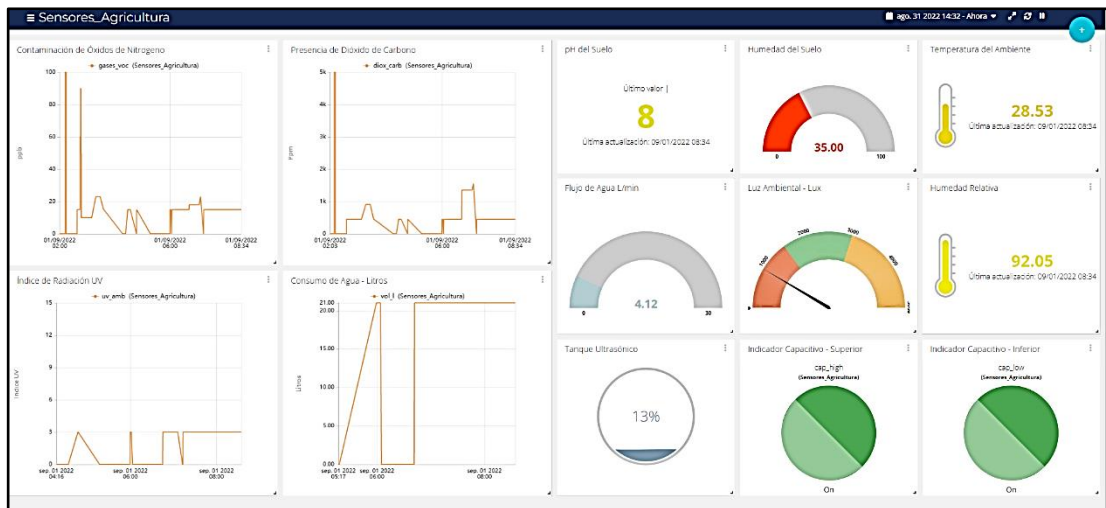


Figura 3. 26 Interfaz gráfica destinada al monitoreo de las variables

Fuente: Autor

Finalmente, se destaca el recurso de almacenamiento de la información que provee la plataforma Ubidots. Esta información tiene una retención de hasta 2 años con una disponibilidad de 24/7, incluso si la suscripción ha expirado. Esto facilita mucho para los usuarios que desean o aprovechan las estadísticas para elaborar planes de mejora o innovación en un campo de estudio general y no solo aplicado a la horticultura de un invernadero. En la figura 3.27 se observa la retención de datos de la variable “Humedad del suelo”.

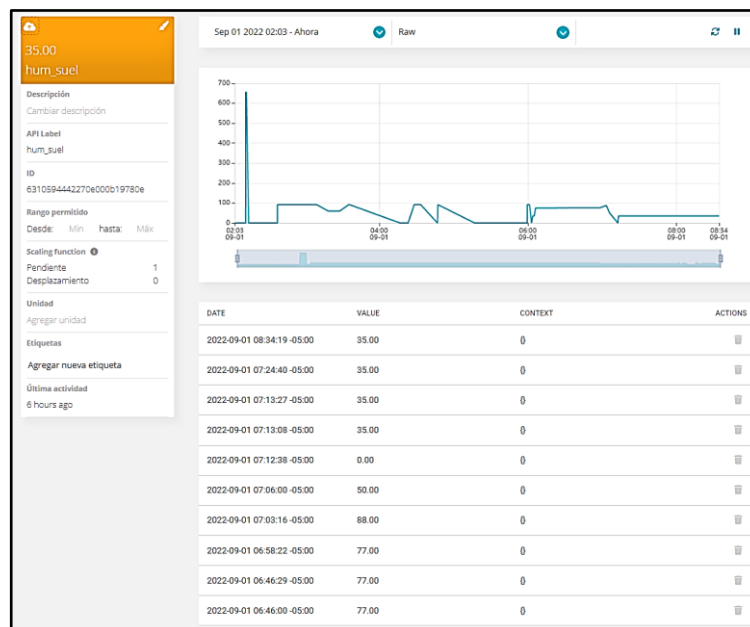


Figura 3. 27 Datos almacenados de la variable “hum_suel” enviados desde el nodo coordinador

Fuente: Autor

3.4 Costo detallado del proyecto

Como último punto a desarrollar en el trabajo de titulación es el costo de los dispositivos electrónicos intervinientes en la red. Se aclara que no se describe en el costo alguna fuente de energía específica, ya que el lector del trabajo de titulación que desee implementar el diseño podrá utilizar diversos tipos de fuentes de energía en DC, cumpliendo el requerimiento de 12V a 2A para un correcto funcionamiento de la red. Los precios pueden variar de acuerdo con el sitio donde se adquiera los dispositivos. Para este trabajo, se adquirieron los dispositivos electrónicos bajo el precio descrito en la tabla 3.8.

Dispositivos	Costo Unitario	Unidades	Costo Total
Controlador/Interruptor de fuente de alimentación DC Yinhing.	\$8.62	1	\$8.62
LM2596 DC. Convertidor Multicanal Taidacent.	\$25.58	1	\$25.58
Arduino Uno Wifi Rev2	\$69.65	1	\$69.65
Arduino Uno SMD Rev3	\$55.01	3	\$165.03
Gikfun xBee Shield 2pcs. EK1185x3	\$14.28	1	\$14.28
xBee - Series 2C TH	\$42.95	4	\$171.80
Gravity: IO Expansion Shield for Arduino V7.1. DFR0265	\$8.90	4	\$35.60
Gravity: Analog UV Sensor V2. SEN0162	\$8.90	1	\$8.90
Gravity: Liquid Level Sensor. SEN0204	\$11.90	2	\$23.80
Gravity: Water Flow Sensor (1/2") For Arduino. SEN0217	\$8.99	1	\$8.99
Gravity: UART Infrared CO2 Sensor (0-50000ppm). SEN0220	\$89.00	1	\$89.00
Gravity: Analog Waterproof Capacitive. SEN0308	\$18.90	2	\$37.80
Gravity: Analog pH Sensor / Meter Pro Kit V2. SEN0169-V2	\$64.90	1	\$64.90
Ambient Light Sensor(0-200klx). SEN0390	\$8.90	1	\$8.90
Gravity: DHT20 Temperature & Humidity Sensor. SEN0497	\$7.90	1	\$7.90
Fermion: ENS160 Air Quality Sensor (Breakout). SEN0515	\$14.90	1	\$14.90
Suscripción Ubidots plan IoT Entrepreneur. Mensual	\$49.00	1	\$49.00
Costo total del desarrollo			\$796.03

Tabla 3. 8 Costo total del diseño de red de sensores y su plataforma IoT

Fuente: Autor

CONCLUSIONES

- En el presente trabajo de titulación se planteó una red de sensores inalámbricos basada en un conjunto de sensores definidos en función de la atención de las necesidades de un invernadero y su cultivo de rosas. Sin embargo, es una base que puede ser utilizada en diferentes cultivos, puesto que la limitante reside en la condiciones de valores de los parámetros ambientales que utilizan diferentes cultivos.
- La red de sensores inalámbricos, al ser constituida por microcontroladores que reciben valores de los sensores, pueden ser expandidas mediante datos más precisos y atender otro tipo de variables presentes en un invernadero. Por ejemplo, se puede implementar un sensor de conductividad eléctrica para determinar la tendencia de nutrientes en el suelo del cultivo y así equilibrarlo de ser necesario.
- La plataforma Ubidots provee una mejora en la gestión de datos, ya que permite identificar y tomar decisiones oportunas de acuerdo con los datos almacenados, además de presentar la información en un entorno accesible y de fácil comprensión para el usuario.
- El uso de microcontrolador Arduino como nodo permite que las variables puedan ser traducidas en datos requeridos por el usuario y aseguren una carga estable para el microcontrolador principal encargado de la comunicación entre la red y la plataforma IoT. Se utilizó módulos de radiofrecuencia, dedicados a la comunicación interna de la red y representados como un medio de transmisión no guiado. Facilita los puntos de accesos donde equipos convencionales no pueden ser desplegados de forma normal, ya que puede ser implementados en espacios abiertos o espacios donde no requieran puntos de energización gracias a que el nodo está diseñado para trabajar con corriente directa de fuentes como batería o paneles solares.
- La red de sensores inalámbricos puede trabajar no solo con sensores, sino con actuadores que realicen tareas específicas gracias a su diseño de comunicación bidireccional, por lo cual puede ser implementado en conjunto con una adecuación en la programación base, su implementación en los nodos y su diseño en la interfaz de la plataforma.

RECOMENDACIONES

- Si los nodos utilizarán una fuente externa de alimentación, el voltaje debe de ser entre 7v a 12v, utilizando el pin “Vin” de los terminales del microcontrolador Arduino o de su tarjeta de expansión. De lo contrario podrían dañar el microcontrolador o algún dispositivo presente en el nodo.
- Si se desea implementar un nuevo sensor, se debe realizar la programación para obtener los datos precisos en el microcontrolador correspondiente y añadir la variable en el payload del microcontrolador coordinador. También se debe utilizar una etiqueta única antes de la variable con la finalidad de que el coordinador identifique el valor de la variable correspondiente y no cause problemas en la comunicación con la plataforma Ubidots.
- En el caso de implementar un nuevo nodo, se tendrá que configurar el nuevo módulo RF con los parámetros descritos en la tabla 3.3 del capítulo 3, correspondiente al trabajo de titulación.
- Para configurar o realizar mantenimientos en el nodo coordinador, se debe desconectar todos los nodos de la red antes de volver a operar el nodo coordinador. Esto se debe a que el nodo coordinador realiza un examen en su rango de alcance para detectar si existen redes con el ID y el canal anteriormente utilizado en su área, y en caso de existir desplegará su ID con otro canal. Los nodos esclavos enviarán solicitud al ID y canal anterior del nodo coordinador, por lo que se opta un apagado masivo de la red. Esto no es necesario cuando se configura o realiza mantenimiento en un nodo esclavo, ya que el no crea la red y solo se conecta.
- El sensor de pH requiere un manejo adecuado y delicado, ya que utiliza una burbuja de vidrio como electrodo, y no puede ser utilizado en objetos duros porque podrá rasguñar o romper el electrodo e invalidará el sensor. Para medir el suelo se requiere mezclar una muestra con agua destilada y eliminar la muestra sólida, filtrando la solución resultante.

Anexo 1: Programación del nodo 1

```
#include <DFRobot_DHT20.h>
#include <SoftwareSerial.h>
#include <DFRobot_ENS160.h>
#define I2C_COMMUNICATION

DFRobot_ENS160_I2C ENS160(&Wire, /*I2CAddr*/ 0x53);
DFRobot_DHT20 TempHum;
SoftwareSerial SenCo2(10, 11); // Transmisión UART RX, TX

unsigned char hexdata[9] = {0xFF, 0x01, 0x86, 0x00, 0x00, 0x00,
0x00, 0x00, 0x79}; //Para leer el comando de la densidad de gas
float Temp = 0;
float Hum = 0;
float Lux = 0;
int ValorAire = 625; //Constante calibrada con sensor en aire
int ValorAgua = 110; //Constante calibrada con sensor sumergido
hasta la profundidad recomendada
int HumSuelo = 0;

void setup()
{
  Serial.begin(9600);
  while (!Serial) {
  }
  SenCo2.begin(9600);
  TempHum.begin();
  ENS160.setPWRMode(ENS160_STANDARD_MODE);
  ENS160.setTempAndHum(/*temperature=*/30.0, /*humidity=*/60.0);
}

void loop()
{
  //sensor CO2
  SenCo2.write(hexdata, 9);
  delay(500);
  long CO2;
  for (int i = 0, j = 0; i < 9; i++)
  {
    if (SenCo2.available() > 0)
    {
      long hi, lo;
      int ch = SenCo2.read();

      if (i == 2) {
        hi = ch; //Alta Concentración
      }
      if (i == 3) {
        lo = ch; //Baja Concentración
      }
      if (i == 8) {
        CO2 = hi * 256 + lo; //Concentración Total de CO2
      }
    }
  }
  //Humedad de Suelo
  HumSuelo = ((ValorAire - analogRead(A0)) * (100.0 / (ValorAire -
ValorAgua)));
  //Temperatura Ambiental
```

```
Temp = TempHum.getTemperature();  
//Humedad Relavita  
Hum = TempHum.getHumidity() * 100;  
//Calidad del Aire  
uint16_t TVOC = ENS160.getTVOC();  
  
Serial.print("ppb"); Serial.print(TVOC); //Serial.print("\n");  
//delay(1000);  
Serial.print("Ppm"); Serial.print(CO2); //Serial.print("\n");  
//delay(1000);  
Serial.print("Cent"); Serial.print(Temp); //Serial.print("\n");  
//delay(1000);  
Serial.print("HumR"); Serial.print(Hum); //Serial.print("\n");  
//delay(1000);  
Serial.print("HumS"); Serial.print(HumSuelo); Serial.print("\n");  
delay(1000);  
}
```

Anexo 2: Programación del nodo 2

```
#include <DFRobot_B_LUX_V30B.h>
#include "DFRobot_PH.h"
#include <EEPROM.h>

#define PH_PIN A1

DFRobot_B_LUX_V30B SenLux(13);
DFRobot_PH ph;

float voltage, pHValue, temperature = 25;
int Lux = 0;

void setup() {
  Serial.begin(9600);
  SenLux.begin();
  ph.begin();
}

void loop() {
  //Luz ambiental
  Lux = SenLux.lightStrengthLux();
  //Luz UV
  int sensorValor;
  int ValorAn = analogRead(0);
  if (ValorAn < 20)
  {
    sensorValor = 0;
  }
  else
  {
    sensorValor = 0.05 * ValorAn - 1;
  }
  //Sensor pH
  static unsigned long timepoint = millis();
  if (millis() - timepoint > 1000U)
  {
    timepoint = millis();
    voltage = analogRead(PH_PIN) / 1024.0 * 5000;
    pHValue = ph.readPH(voltage, temperature); // Convierte el
    voltaje a pH compensando el registro con temperatura
  }

  Serial.print("pH"); Serial.print(pHValue); //Serial.print("\n");
  //delay(1000);
  Serial.print("LX"); Serial.print(Lux); //Serial.print("\n");
  //delay(1000);
  Serial.print("UV"); Serial.print(sensorValor); Serial.print("\n");
  delay(1000);
}
```

Anexo 3: Programación del nodo 3

```
#define ECHOPIN 7// Pin recibe echo
#define TRIGPIN 8// Pin envia trigger

volatile int PulsosSensor; //variable para la cantidad de pulsos
recibidos
int PinSensor = 2; //Sensor conectado en el pin 2
float factor_conversion = 7.11; //para convertir de frecuencia a
caudal
float volumen = 0;
long dt = 0; //variación de tiempo por cada bucle
long t0 = 0; //milisegundos del bucle anterior
int Capacit1 = 0;
int Capacit2 = 0;

//Función que se ejecuta en interrupción
void CuentaPulsos ()
{
  PulsosSensor++; //incrementamos la variable de pulsos
}
//Función para obtener frecuencia de los pulsos
int FreqTotal()
{
  int frecuencia;
  PulsosSensor = 0; //Ponemos a 0 el número de pulsos
  interrupts(); //Habilitamos las interrupciones
  delay(1000); //muestra de 1 segundo
  noInterrupts(); //Deshabilitamos las interrupciones
  frecuencia = PulsosSensor; //Hz(pulsos por segundo)
  return frecuencia;
}

void setup()
{
  Serial.begin(9600);
  pinMode(PinSensor, INPUT);
  attachInterrupt(0, CuentaPulsos, RISING); //(Interrupción
0(Pin2),función,Flanco de subida)
  t0 = millis();
  pinMode(11, INPUT);
  pinMode(12, INPUT);
  pinMode(ECHOPIN, INPUT);
  pinMode(TRIGPIN, OUTPUT);
  digitalWrite(ECHOPIN, HIGH);
}

void loop ()
{
  //Configuración Sensor de Flujo de Agua
  if (Serial.available())
  {
    if (Serial.read() == 'r')
    {
      volumen = 0; //restablecemos el volumen con r miniscula
    }
  }
  float frecuencia = FreqTotal(); //obtenemos la frecuencia de los
pulsos en Hz
```



```

    float caudal_L_m = frecuencia / factor_conversion; //calculamos el
caudal en L/m

    dt = millis() - t0; //calculamos la variación de tiempo
    t0 = millis();
    volumen = volumen + (caudal_L_m / 60) * (dt / 1000); //
volumen(L)=caudal(L/s)*tiempo(s)

    //Configuración Sensores Capacitivos
    Capacit1 = digitalRead(11);
    Capacit2 = digitalRead(12);

    //Configuración Sensor Ultrasónico
    digitalWrite(TRIGPIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGPIN, HIGH);
    delayMicroseconds(20);
    digitalWrite(TRIGPIN, LOW);
    int distancia = pulseIn(ECHOPIN, HIGH, 30000);
    distancia = distancia * 0.034 / 2;

    Serial.print ("L-min"); Serial.print(caudal_L_m, 3);
//Serial.print("\n");
    //delay(1000);
    Serial.print ("Vol"); Serial.print (volumen, 3);
//Serial.print("\n");
    //delay(1000);
    Serial.print("ALTO"); Serial.print(Capacit1, DEC);
//Serial.print("\n");
    //delay(1000);
    Serial.print("BAJO"); Serial.print(Capacit2, DEC);
//Serial.print("\n");
    //delay(1000);
    Serial.print("CM"); Serial.print(distancia); Serial.print("\n");
    delay(1000);
}

```

Anexo 4: Programación del nodo Coordinador

```
#include <SPI.h>
#include <WiFiNINA.h>
#define DEVICE_LABEL "sensores_agricultura"
#define TOKEN "TokenPrivado"

//Configuración de Arduino
const int HTTPPORT = 443;
char const *AGENT = "Arduino Wifi";
char const *HTTP_VERSION = " HTTP/1.1\r\n";
char const *VERSION = "1.0";
char const *PATH = "/api/v1.6/devices/";
//Configuración wifi
char const * SSID_NAME = "NetQuantum";
char const * SSID_PASS = "D0ntTurn0ff.Pls!";
int status = WL_IDLE_STATUS;
char const *SERVER = "industrial.api.ubidots.com";
WiFiSSLClient client;
//Configuración de variables
char const * VOC = "Gases_VOC";
char const * CO2 = "Diox_Carb";
char const * Temp = "Temp_Amb";
char const * HumR = "Hum_Rel";
char const * HumS = "Hum_Suel";
char const * pH = "pH_Suel";
char const * LX = "LX_Amb";
char const * UV = "UV_Amb";
char const * Flow = "Flow_Agua";
char const * VOL = "Vol_L";
char const * CapH = "Cap_High";
char const * CapL = "Cap_Low";
char const * Dist = "Ult_Dist";
String readString;
int index = 0;
int index2 = 0;
int index3 = 0;
int index4 = 0;
int index5 = 0;
int index6 = 0;
int index7 = 0;
int index8 = 0;
int index9 = 0;
int index10 = 0;
int index11 = 0;
int index12 = 0;
int index13 = 0;
//Grupo1 Inicial String
String TotalVOC;
String TotalCO2;
String TotalCent;
String TotalHumR;
String TotalHumS;
String TotalpH;
String TotalLX;
String TotalUV;
String TotalFA;
String TotalVOL;
String TotalCapH;
String TotalCapL;
String TotalUDist;
```

```

void printWiFiStatus() {
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);
    long rssi = WiFi.RSSI();

    Serial.print("signal strength (RSSI):");
    Serial.print(rssi);
    Serial.println(" dBm");
}

void getResponseServer() {
    Serial.println(F("\Respuesta del servidor de Ubidots:\n"));
    while (client.available()) {
        char c = client.read();
        Serial.print(c);
    }
}

void waitServer() {
    int timeout = 0;
    while (!client.available() && timeout < 5000) {
        timeout++;
        delay(50);
        if (timeout >= 5000) {
            Serial.println(F("Error, max timeout reached"));
            break;
        }
    }
}

void sendData(char* payload) {
    int contentLength = strlen(payload);
    if (client.connect(SERVER, HTTPPORT)) {
        Serial.println("connected to server");
        client.print(F("POST "));
        client.print(PATH);
        client.print(DEVICE_LABEL);
        client.print(F("/"));
        client.print(HTTP_VERSION);
        client.print(F("Host: "));
        client.print(SERVER);
        client.print(F("\r\n"));
        client.print(F("User-Agent: "));
        client.print(AGENT);
        client.print(F("\r\n"));
        client.print(F("X-Auth-Token: "));
        client.print(TOKEN);
        client.print(F("\r\n"));
        client.print(F("Connection: close\r\n"));
        client.print(F("Content-Type: application/json\r\n"));
        client.print(F("Content-Length: "));
        client.print(contentLength);
        client.print(F("\r\n\r\n"));
        client.print(payload);
        client.print(F("\r\n"));

        Serial.print(F("POST "));
    }
}

```

```

    Serial.print(PATH);
    Serial.print(DEVICE_LABEL);
    Serial.print(F("/"));
    Serial.print(HTTP_VERSION);
    Serial.print(F("Host: "));
    Serial.print(SERVER);
    Serial.print(F("\r\n"));
    Serial.print(F("User-Agent: "));
    Serial.print(AGENT);
    Serial.print(F("\r\n"));
    Serial.print(F("X-Auth-Token: "));
    Serial.print(TOKEN);
    Serial.print(F("\r\n"));
    Serial.print(F("Connection: close\r\n"));
    Serial.print(F("Content-Type: application/json\r\n"));
    Serial.print(F("Content-Length: "));
    Serial.print(contentLength);
    Serial.print(F("\r\n\r\n"));
    Serial.print(payload);
    Serial.print(F("\r\n"));
    waitServer();
    getResponseServer();
}
client.stop();
}

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

    while (!Serial) {
    }
    if (WiFi.status() == WL_NO_MODULE) {
        Serial.println("Fallo en comunicación wifi");
        while (true);
    }
    while (status != WL_CONNECTED) {
        Serial.print("Conectando con wifi: ");
        Serial.println(SSID_NAME);
        status = WiFi.begin(SSID_NAME, SSID_PASS);
        delay(5000);
    }
    Serial.println("Conexión Correcta");
    printWiFiStatus();
}

void loop() {
    Leer_Serial();
    if (readString.length() > 0) {
        Serial.println(readString);
        //Grupo1 syntax ejemplo ppb15Ppm450Cent38.03HumR92.05HumS35
        if (readString.indexOf("ppb") >= 0) {
            index = readString.indexOf("Ppm");
            TotalVOC = readString.substring(3, index);
            Serial.println(TotalVOC);
        }
        if (readString.indexOf("Ppm") > 0) {
            index2 = readString.indexOf("Cent");
            TotalCO2 = readString.substring(index + 3, index2);
            Serial.println(TotalCO2);
        }
        if (readString.indexOf("Cent") > 0) {

```

```

    index3 = readString.indexOf("HumR");
    TotalCent = readString.substring(index2 + 4, index3);
    Serial.println(TotalCent);
}
if (readString.indexOf("HumR") > 0) {
    index4 = readString.indexOf("HumS");
    TotalHumR = readString.substring(index3 + 4, index4);
    Serial.println(TotalHumR);
}
if (readString.indexOf("HumS") > 0) {
    index5 = readString.indexOf('\n');
    TotalHumS = readString.substring(index4 + 4, index5);
    Serial.println(TotalHumS);
}
//Grupo2 syntax ejemplo pH8LX852UV3
if (readString.indexOf("pH") >= 0) {
    index6 = readString.indexOf("LX");
    TotalpH = readString.substring(2, index6);
    Serial.println(TotalpH);
}
if (readString.indexOf("LX") > 0) {
    index7 = readString.indexOf("UV");
    TotalLX = readString.substring(index6 + 2, index7);
    Serial.println(TotalLX);
}
if (readString.indexOf("UV") > 0) {
    index8 = readString.indexOf('\n');
    TotalUV = readString.substring(index7 + 2, index8);
    Serial.println(TotalUV);
}
//Grupo3 syntax ejemplo L-min4.12Vol121ALTO1BAJO1CM29
if (readString.indexOf("L-min") >= 0) {
    index9 = readString.indexOf("Vol");
    TotalFA = readString.substring(5, index9);
    Serial.println(TotalFA);
}
if (readString.indexOf("Vol") > 0) {
    index10 = readString.indexOf("ALTO");
    TotalVOL = readString.substring(index9 + 3, index10);
    Serial.println(TotalVOL);
}
if (readString.indexOf("ALTO") > 0) {
    index11 = readString.indexOf("BAJO");
    TotalCapH = readString.substring(index10 + 4, index11);
    Serial.println(TotalCapH);
}
if (readString.indexOf("BAJO") > 0) {
    index12 = readString.indexOf("CM");
    TotalCapL = readString.substring(index11 + 4, index12);
    Serial.println(TotalCapL);
}
if (readString.indexOf("CM") > 0) {
    index13 = readString.indexOf('\n');
    TotalUDist = readString.substring(index12 + 2, index13);
    Serial.println(TotalUDist);
}
char payload[1000];
char str_val_1[30];
char str_val_2[30];
char str_val_3[30];
char str_val_4[30];

```

```

char str_val_5[30];
char str_val_6[30];
char str_val_7[30];
char str_val_8[30];
char str_val_9[30];
char str_val_10[30];
char str_val_11[30];
char str_val_12[30];
char str_val_13[30];
TotalVOC.toCharArray(str_val_1, 30);
TotalCO2.toCharArray(str_val_2, 30);
TotalCent.toCharArray(str_val_3, 30);
TotalHumR.toCharArray(str_val_4, 30);
TotalHumS.toCharArray(str_val_5, 30);
TotalpH.toCharArray(str_val_6, 30);
TotalLX.toCharArray(str_val_7, 30);
TotalUV.toCharArray(str_val_8, 30);
TotalFA.toCharArray(str_val_9, 30);
TotalVOL.toCharArray(str_val_10, 30);
TotalCapH.toCharArray(str_val_11, 30);
TotalCapL.toCharArray(str_val_12, 30);
TotalUDist.toCharArray(str_val_13, 30);
sprintf(payload, "%s", ""); //Limpia la carga
sprintf(payload, "{\");
sprintf(payload, "%s%s\":%s", payload, VOC, str_val_1);
sprintf(payload, "%s,\"%s\":%s", payload, CO2, str_val_2);
sprintf(payload, "%s,\"%s\":%s", payload, Temp, str_val_3);
sprintf(payload, "%s,\"%s\":%s", payload, HumR, str_val_4);
sprintf(payload, "%s,\"%s\":%s", payload, HumS, str_val_5);
sprintf(payload, "%s,\"%s\":%s", payload, pH, str_val_6);
sprintf(payload, "%s,\"%s\":%s", payload, LX, str_val_7);
sprintf(payload, "%s,\"%s\":%s", payload, UV, str_val_8);
sprintf(payload, "%s,\"%s\":%s", payload, Flow, str_val_9);
sprintf(payload, "%s,\"%s\":%s", payload, VOL, str_val_10);
sprintf(payload, "%s,\"%s\":%s", payload, CapH, str_val_11);
sprintf(payload, "%s,\"%s\":%s", payload, CapL, str_val_12);
sprintf(payload, "%s,\"%s\":%s", payload, Dist, str_val_13);
sprintf(payload, "%s}", payload);
sendData(payload);
readString = "";
delay(1500);
}
}

void Leer_Serial() {
while (Serial.available()) {
delay(25); //buffer
if (Serial.available() > 0) {
char c = Serial.read();
if (c == ',') {
break;
}
readString += c;
}
}
}
}

```

BIBLIOGRAFÍA

- Ahmad, H. (2021). *Concepts, Applications, Experimentation and Analysis of Wireless Sensor Networks* (Second Edition). Springer Cham.
<https://doi.org/10.1007/978-3-030-58015-5>
- Arduino. (s/f). [Electronic & IoT]. Arduino Official Store. Recuperado el 24 de agosto de 2022, de <https://store.arduino.cc/collections/arduino>
- Arturo Asistimbay, F. A., & Pin Soledispa, K. L. (2021). *Diseño y desarrollo de un prototipo inalámbrico para un sistema de casilleros Universitarios utilizando dispositivos IOT Y Ubidots* [Tesis de Grado, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Networking y Telecomunicaciones].
<http://repositorio.ug.edu.ec/handle/redug/59760>
- Astudillo, R. (2016). *Diseño e implementación de un prototipo de medidor de nivel de agua a través de un sensor ultrasónico para pasos deprimidos* [Tesis de Grado, Universidad Israel].
<http://repositorio.uisrael.edu.ec/handle/47000/1181>
- Bernstein, H. (2022). *Measuring electronics and sensors: Basics of measurement technology, sensors, analog, and digital signal processing*. Springer.
<https://doi.org/10.1007/978-3-658-35067-3>
- Castaño, J. (2016). *Diseño de una Maleta Didáctica de Pruebas para Sensores de Proximidad en el Área de Mecatrónica* [Tesis de Grado, Universidad Tecnológica de Pereira]. <https://repositorio.utp.edu.co/items/b60f3167-aacb-4446-886e-82eafac98f57>
- Chandra, S., Jayasundera, K., & Postolache, O. (2019). *Modern Sensing Technologies* (Vol. 29). Springer Nature Switzerland AG.
<https://doi.org/10.1007/978-3-319-99540-3>
- Corona, L., Abarca, G., & Mares, J. (2014). *Sensores y actuadores. Aplicaciones con Arduino*. Grupo Editorial Patria, S.A de C.V.
<http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=4569609>

- Corrêa, F. (2021). *Sensors and Actuators*. World Scientific Publishing Co. Pte. Ltd. <https://www.worldscientific.com/worldscibooks/10.1142/12426#t=suppl>
- Dumka, A., Chaurasiya, S., & Mandoria, H. (2019). *A complete guide to wireless sensor networks: From inception to current trends*. CRS Press / Taylor & Francis Group. <https://doi.org/10.1201/9780429286841>
- Fericean, S. (2018). *Inductive Sensors for Industrial Applications*. Artech House. <https://uk.artechhouse.com/Inductive-Sensors-for-Industrial-Applications-P1998.aspx>
- Guasch, J., & Calleja, M. (2019). *Monitorización de sensores con Arduino utilizando el protocolo MQTT* [Tesis de Grado, Universidad Politécnica de Cataluña]. <https://upcommons.upc.edu/handle/2117/134193>
- Gupta, D., Hugo, V., Khanna, A., & Lala, P. (2021). *Smart Sensors for Industrial Internet of Things*. Springer Cham. <https://doi.org/10.1007/978-3-030-52624-5>
- Hering, E., & Schönfelder, G. (Eds.). (2022). *Sensors in Science and Technology: Functionality and Application Areas*. Springer. <https://doi.org/10.1007/978-3-658-34920-2>
- Hidalgo, J. (2019). *Desarrollo de un prototipo de sensor inteligente de bajo costo para dispositivos de navegación de personas no videntes usando plataformas embebidas* [Tesis de Grado, Escuela Superior Politécnica de Chimborazo]. <http://dspace.esPOCH.edu.ec/handle/123456789/13665>
- Ida, N. (2020). *Sensors, actuators, and their interfaces: A multidisciplinary introduction* (Second Edition). The Institution of Engineering and Technology. <https://shop.theiet.org/sensors-actuators-and-their-interfaces-2nd-edition>
- Kurose, J., & Ross, K. (2021). *Computer networking: A top-down approach* (Eighth Edition). Pearson. <https://www.pearson.com/en-us/subject-catalog/p/computer-networking/P200000003334/9780135928615>
- Kuzmiakova, A. (2021). *Computer networks and communications*. Arcler Press. <http://public.ebib.com/choice/PublicFullRecord.aspx?p=6453469>

- Lang, W. (2021). *Sensors and Measurement Systems* (Second Edition). River Publishers. <https://ieeexplore.ieee.org/book/9428690>
- Lederkremer, M. (2019). *Redes Informáticas* (Primera Edición). Six Ediciones. https://books.google.com.ec/books?id=7frADwAAQBAJ&printsec=copyright&redir_esc=y#v=onepage&q&f=false
- Liberatori, M. (2018). *Redes de Datos y sus Protocolos* (Primera Edición). EUDEM. <http://www2.mdp.edu.ar/images/eudem/pdf/redes%20de%20datos.pdf>
- López, A. C. (2021). *Bioestimulación del crecimiento del botón floral en el cultivo de rosa (Rosa sp.), variedad Orange Crush* [Tesis de Grado, Universidad Técnica de Ambato]. <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/32361>
- Méndez, E. (2021). *Estudio y diseño de un parqueo inteligente utilizando Arduino a través del Internet de las cosas (IoT)*. [Tesis de Grado, Universidad Católica de Santiago de Guayaquil]. <http://repositorio.ucsg.edu.ec/handle/3317/17170>
- Muñiz, G. (2021). *Análisis de un sistema de microcontroladores para mejorar la enseñanza de la electrónica digital en la carrera de tecnologías de la información* [Tesis de Grado, Universidad Estatal Del Sur De Manabí]. <http://repositorio.unesum.edu.ec/handle/53000/3387>
- Muñoz, C., & Lesmes, E. (2016). *Diseño y Construcción de un Banco Didáctico de Pruebas para Sensores de Proximidad del Laboratorio de Tecnología en Electricidad* [Tesis de Grado, Universidad Distrital Francisco José de Caldas]. <http://repository.udistrital.edu.co/handle/11349/3126>
- Narváez, K., & Contreras, V. (2020). *Diseño y desarrollo de un prototipo de red de sensores IOT utilizando tecnología LoraWan para el monitoreo de parámetros ambientales en interiores y exteriores* [Tesis de Grado, Universidad Politécnica Salesiana Sede Cuenca]. <http://dspace.ups.edu.ec/handle/123456789/19439>
- Novák, M. (2020). *Introduction to Sensors for Electrical and Mechanical Engineers* (First Edition). CRC Press. <https://doi.org/10.1201/9781003081692>

- Peterson, L. L., & Davie, B. S. (2022). *Computer Networks: A Systems Approach* (Sixth Edition). Morgan Kaufmann Publishers. <https://www.elsevier.com/books/computer-networks/peterson/978-0-12-818200-0>
- Rupitsch, S. J. (2019). *Piezoelectric Sensors and Actuators: Fundamentals and Applications* (First Edition). Springer Berlin Heidelberg: Imprint: Springer. <https://doi.org/10.1007/978-3-662-57534-5>
- Singh, P. K., Bhargava, Bharat K., Paprzycki, M., Kaushal, Narottam Chand, Hong, Wei-Chiang, Bhargava, B., Chand, N., & Hong, W.-C. (2020). *Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario's*. Springer Nature Switzerland AG. <https://doi.org/10.1007/978-3-030-40305-8>
- Ubidots IoT platform features*. (s/f). [IoT Cloud]. Ubidots Official Website. Recuperado el 24 de agosto de 2022, de <https://ubidots.com/platform/>
- Velasco, A. (2019). *Sistema de monitoreo de condiciones climáticas en un invernadero de rosas* [Tesis de Grado, Universidad de las Américas]. <https://dspace.udla.edu.ec/handle/33000/10901+&cd=1&hl=en&ct=clnk&gl=e>
- Velasteguí, I., & Fernández, D. (2020). *Diseño e implementación de la arquitectura del internet industrial de las cosas para la recolección de datos en un proceso de manufactura*. [Tesis de Grado, Universidad del Azuay]. <http://dspace.uazuay.edu.ec/handle/datos/9814>



DECLARACIÓN Y AUTORIZACIÓN

Yo, **Chaucala Escalante, Joel Alexander** con C.C: # **0955966841** autor/a del trabajo de titulación: **Diseño de una red de sensores inalámbricos para la horticultura climáticamente inteligente para un invernadero de rosas mediante la plataforma Ubidots** previo a la obtención del título de **Ingeniero en Telecomunicaciones** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 09 de septiembre de 2022

f. Joel Chaucala E.

Nombre: **Chaucala Escalante, Joel Alexander**

C.C: **0955966841**



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN			
TEMA Y SUBTEMA:	Diseño de una red de sensores inalámbricos para la horticultura climáticamente inteligente para un invernadero de rosas mediante la plataforma Ubidots		
AUTOR(ES)	Chaucala Escalante Joel Alexander		
REVISOR(ES)/TUTOR(ES)	Vélez Tacuri Efraín Oliverio		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería en Telecomunicaciones		
TÍTULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	09 de septiembre del 2022	No. DE PÁGINAS:	66
ÁREAS TEMÁTICAS:	Comunicaciones Inalámbricas, Internet de las cosas, Automatización.		
PALABRAS CLAVES/ KEYWORDS:	Zigbee, Ubidots, IoT, Sensores, Radiofrecuencia, Redes, Invernadero, Rosas, Arduino.		
RESUMEN/ABSTRACT:	<p>El presente trabajo de titulación plantea un diseño de red de sensores inalámbricos que permite el monitoreo de los variables climáticas y recursos presentes en un invernadero de cultivo de rosas, con la finalidad de comprender la incidencia del entorno en el comportamiento de las rosas para identificar y proponer mejorar en el cuidado y crecimiento, asegurando así la calidad de la producción. El primer capítulo describe la situación del cultivo de rosas en regiones que no son óptimas para su desarrollo, con enfoque en la horticultura de un invernadero, y su necesidad de implementar sensores que genere información para la gestión de respuestas adecuadas frente a un requerimiento ambiental. Por ello, se plantea el problema y su solución mediante el objetivo general y objetivos específicos, así como el tipo de investigación implementado en el trabajo de titulación. En el segundo capítulo se encuentra la fundamentación teórica con la finalidad de comprender en detalle las partes del diseño de red mediante recursos académicos y científicos. En el tercer capítulo se determina las variables que se definieron para el desarrollo del trabajo de titulación, así como el diseño de los nodos, la programación de cada uno en función a los sensores seleccionado, la configuración de los módulos radiofrecuencia utilizados, la comunicación del nodo principal y la plataforma IoT remota, y el diseño de la interfaz en la plataforma IoT para la visualización de datos en tiempo real.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593-987524219	E-mail: alexander-ch@outlook.com	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE):	Nombre: Mgs. Vélez Tacuri, Efraín Oliverio		
	Teléfono: +593-994084215		
	E-mail: efrain.velez@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			