

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PUCP

**DISEÑO DE UN SISTEMA DE MONITOREO PARA UN
INVERNADERO EXPERIMENTAL BASADO EN UNA RED DE
SENSORES**

**Tesis para obtener el título profesional de Ingeniero de las
Telecomunicaciones**

AUTOR:

Jhordy Merlin Pozo Gonzales

ASESOR:

Mg. Luis Angelo Velarde Criado

Lima, febrero, 2021

Resumen

La agricultura en el Perú es una actividad económica muy sensible a los cambios en el clima, sin embargo, de ella depende la seguridad alimentaria de todos los peruanos. Debido a esto, se impulsa la aplicación de técnicas agrícolas que ofrezcan ventajas respecto al modelo tradicional. Por ejemplo, una de ellas es el uso de invernaderos, la cual permite, entre otras cosas, protección contra climas adversos, reducción del tiempo de cultivo y mejora de la calidad a través de una atmósfera interna controlable. Comúnmente, estos invernaderos son operados manualmente por el agricultor en base a su experiencia, por lo que no se obtiene condiciones ambientales adecuadas para una agricultura de precisión. Sumado a esto, el cambio climático y la diversidad de los pisos altitudinales a lo largo del país, hacen que sea necesario buscar alternativas de adaptación tecnológicas. En este sentido, el Internet de las Cosas (*IoT-Internet of Things*) y el desarrollo de software, ponen a nuestra disposición herramientas para la implementación de soluciones innovadoras aplicadas a la agricultura, las cuales ayudarán a incrementar el rendimiento y la calidad de los productos. Por lo tanto, la solución tecnológica propuesta, como objetivo de esta tesis, es diseñar un sistema de monitoreo que permita hacer el seguimiento de variables críticas dentro del invernadero basado en una red de sensores.

Dedicatoria

A mis padres y mi hermana, por su apoyo incondicional.

A Dios, por guiar mi camino.



Tabla de Contenidos

Índice de Tablas	vi
Índice de Figuras.....	vii
Capítulo 1. Identificación y Diagnóstico de la Problemática	1
1.1 Aspectos Generales de un Invernadero	1
1.2 Tipos de Invernaderos Según su Estructura	2
1.3 Importancia del Cultivo en Invernaderos	4
1.4 Referencias de Uso de Invernaderos en el Mundo	5
1.5 Situación Actual y Tendencias del Cultivo Invernadero en el Perú.....	6
1.6 Declaración del Marco Problemático.....	8
Capítulo 2. Marco Teórico	9
2.1 Introducción	9
2.2 Redes de Sensores	9
2.2.1 Definición	10
2.2.2 Topología	11
2.2.3 Estándares de Comunicación para Redes de Sensores Inalámbricas.....	12
2.2.4 Principales Fabricantes de Hardware.....	19
2.3 Monitoreo.....	20
2.3.1 Arquitectura de Software para Aplicaciones Web.....	21
2.3.2 Servicios Web	24
2.3.3 Base de Datos.....	26

2.3.4	Plataformas Cloud Computing.....	30
2.4	Modelo Teórico.....	35
Capítulo 3. Diseño de un Sistema de Monitoreo para un Invernadero Experimental Basado en una Red de Sensores		
3.1	Objetivos	37
3.1.1	Objetivo general.....	37
3.1.2	Objetivos específicos	37
3.2	Etapas del Proyecto	37
3.3	Análisis y Recopilación de Requerimientos.....	38
3.4	Diseño.....	40
3.4.1	Arquitectura de Software y Servicios Web.....	40
3.4.2	Lenguaje de Programación, Servidor Web y DBMS.....	41
3.4.3	Modelado de la Base de Datos.....	43
3.4.4	Mock Up y Diagramas de Flujo.....	46
3.5	Desarrollo.....	46
3.5.1	Interfaz Gráfica	47
Capítulo 4. Pruebas, Resultados y Análisis de Costos.....		
4.1	Validación de la Aplicación Web.....	51
4.2	Integración de la Aplicación Web con la Red de Sensores.....	53
4.2.1	Escenario Local.....	53
4.2.2	Hardware.....	54
4.2.3	Sistema de Comunicación entre la Red de Sensores y el Servidor Web	55

4.2.4	Integración	57
4.3	Resultados	58
4.3.1	Post procesamiento y análisis	62
4.4	Análisis de Costos	64
	Conclusiones	68
	Recomendaciones	70
	Bibliografía	71



Índice de Tablas

Tabla 1 Estándares de comunicación para redes de sensores inalámbricas.....	12
Tabla 2 Principales características de los modelos de Base de Datos	27
Tabla 3 Ventajas y desventajas de las arquitecturas de Base de Datos	28
Tabla 4 Listado de Requerimientos Funcionales y No Funcionales.....	39
Tabla 5 Entidades y atributos de la BD	43
Tabla 6 Relaciones entre las entidades de la BD	45
Tabla 7 Beans, Daos, Servlets y JSP implementados	46
Tabla 8 Pruebas de validación de la aplicación web.....	51
Tabla 9 Tipo de dato del JSON enviado por la red de sensores	56
Tabla 10 Tipo de dato del JSON enviado por el servidor web	56
Tabla 11 Inversión capital.....	65
Tabla 12 Consumo de energía mensual de la red de sensores	66
Tabla 13 Consumo de datos mensual de la red de sensores	66
Tabla 14 Costos operativos mensuales	67

Índice de Figuras

Figura 1 Orientación de Invernaderos.....	1
Figura 2 Invernadero tipo túnel.....	2
Figura 3 Invernadero tipo capilla.....	2
Figura 4 Invernadero tipo asimétrico.....	3
Figura 5 Invernadero tipo parral.....	3
Figura 6 Invernadero Domo PUCP.....	4
Figura 7 Principales áreas de invernaderos a nivel mundial.....	5
Figura 8 Vulnerabilidad agrícola por distrito.....	7
Figura 9 Infraestructura de una red inalámbrica de sensores.....	10
Figura 10 Topologías de red.....	11
Figura 11 Estándares basados en IEEE 802.15.4.....	13
Figura 12 Topologías de red en IEEE 802.15.4.....	13
Figura 13 Arquitectura ZigBee.....	14
Figura 14 Pila de protocolos 6LoWPAN.....	15
Figura 15 Arquitectura de red SIGFOX.....	17
Figura 16 Arquitectura LoRaWAN.....	19
Figura 17 Esquema general del entorno de servicios web de sensores.....	20
Figura 18 Arquitectura genérica en capas.....	21
Figura 19 Arquitectura cliente-servidor para una filmoteca.....	22
Figura 20 Arquitectura MVC.....	23
Figura 21 Capas de la pila de protocolo de Web Services.....	25
Figura 22 Arquitectura Azure IoT Suite.....	31
Figura 23 Ejemplo de Solución AWS IoT.....	32
Figura 24 Servicios IoT de Google.....	34
Figura 25 Solución IoT Ubidots.....	35

Figura 26 Modelo teórico.....	36
Figura 27 Etapas del Proyecto	38
Figura 28 Arquitectura de software y servicios web seleccionados	41
Figura 29 Arquitectura DBMS MySQL.	42
Figura 30 Panel principal de la aplicación web	47
Figura 31 Mediciones de los sensores	48
Figura 32 Esquema de integración.....	54
Figura 33 Circuito de la red de sensores	57
Figura 34 Nuevo invernadero creado.....	58
Figura 35 Nuevos sensores y modo de control configurado.....	58
Figura 36 Resultados Monitor Serie Arduino	59
Figura 37 Mediciones históricas de un sensor de temperatura	59
Figura 38 Mediciones históricas filtradas de los tres sensores de temperatura	60
Figura 39 Modo manual, estado “close”, LED OFF	60
Figura 40 Modo manual, estado “open”, LED ON.....	61
Figura 41 Modo Automático con umbral configurado	61
Figura 42 Modo automático LED OFF.....	62
Figura 43 Modo automático LED ON	62
Figura 44 Filtrado de mediciones en MATLAB.....	63
Figura 45 Reporte de temperatura de la estación Campo de Marte - SENAMHI	63
Figura 46 Distribución de sensores dentro del invernadero.....	65
Figura 47 Esquema de conexión inalámbrica para el invernadero	65

Capítulo 1. Identificación y Diagnóstico de la Problemática

1.1 Aspectos Generales de un Invernadero

El invernadero viene ganando presencia e importancia como una técnica agrícola avanzada respecto al modelo tradicional. Esto se debe a que el invernadero es una estructura cerrada capaz de proteger a los cultivos con la ayuda de cubiertas transparentes y que tiene por objetivo “reproducir o simular las condiciones climáticas más adecuadas para el crecimiento y desarrollo de las plantas cultivadas en su interior, con cierta independencia del medio exterior y cuyas dimensiones posibilitan el trabajo de las personas” (Guía de Invernaderos, s.f.).

Las condiciones climáticas dentro del invernadero dependen de cuatro factores principales: temperatura, humedad relativa, luz y dióxido de carbono. Estos afectan directamente al cultivo a lo largo de su ciclo de vida, por lo que es necesario mantenerlos dentro de sus límites según la especie vegetal, de lo contrario, las consecuencias son fatales (Control Climático en Invernaderos, s.f.).

Para la construcción de un invernadero es importante definir su orientación, ya sea Norte-Sur o Este-Oeste, la cual dependerá si la zona es templada o cálida respectivamente (ver Figura 1). En su gran mayoría son construidas en dirección Norte-Sur para aprovechar mejor la luz solar (Guía de Invernaderos, s.f.).

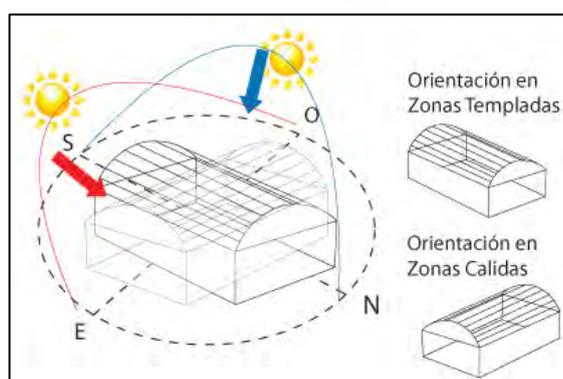


Figura 1 Orientación de Invernaderos
Tomado de Guía de Invernaderos (s.f.)

1.2 Tipos de Invernaderos Según su Estructura

Invernadero túnel o semicilíndrico

El invernadero tipo túnel es de los más sencillos y prácticos ya que en su mayoría son prefabricados. Como su propio nombre lo dice, su forma es curva o cilíndrica, compuesto de varios arcos galvanizados, lo que le permite soportar fuertes vientos. Entre las ventajas más importantes tenemos su buena capacidad de control del clima y su reparto de la luminosidad en el interior (Tipos de Invernaderos, s.f.).



Figura 2 Invernadero tipo túnel
Tomado de Tipos de Invernaderos (s.f.)

Invernadero capilla

El invernadero tipo capilla, a diferencia del tipo túnel, tiene paredes rectas en los laterales. También existe una variante denominada multicapilla, la cual consiste en dos estructuras adyacentes con cubiertas semicirculares metálicas. Entre sus ventajas principales tenemos la facilidad para la evacuación de agua de lluvia y su adaptabilidad para la instalación de ventilación cenital o perimetral (Tipos de Invernaderos, s.f.).

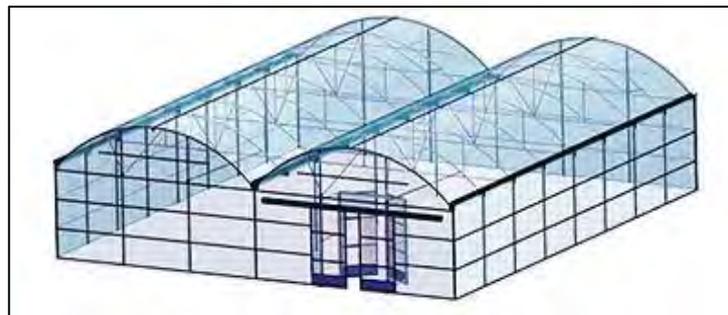


Figura 3 Invernadero tipo capilla
Tomado de Tipos de Invernaderos (s.f.)

Invernadero asimétrico

A diferencia de los invernaderos vistos anteriormente, el tipo asimétrico consta de dos cubiertas, una más inclinada que la otra. El ángulo de inclinación dependerá de la luz del sol al mediodía, pues la idea es aprovecharla al máximo. Su ventaja principal es la buena ventilación debido a su altura y el aprovechamiento de la luz (Tipos de Invernaderos, s.f.).



Figura 4 Invernadero tipo asimétrico
Tomado de Tipos de Invernaderos (s.f.)

Invernadero tipo parral

Este tipo de invernadero es el más simple, ya que solo cuenta con dos partes, una vertical y una horizontal con ligera pendiente. La altura típica es de 3 a 3.5 m. con ancho variable y largo de 20 m. aproximadamente. La ventilación se obtiene desde los laterales. La mayor ventaja es lo barato en su construcción (Guía de Invernaderos, s.f.).



Figura 5 Invernadero tipo parral
Tomado de Tipos de Invernaderos (s.f.)

Invernadero Domo

La forma particular de este tipo de invernadero toma como referencia la estructura de la cúpula geodésica, popularizado por el científico Buckminster Fuller en los mediados del siglo 20. Esta forma de estructura es muy fuerte, incluso, al recibir corrientes de viento lo tienden a fijar al suelo ya que no hay superficie de succión. Al tener una forma semiesférica, el Domo aprovecha al máximo la energía solar y a la vez refleja la luz al interior evitando así la pérdida de calor (Montes, s.f.). En la Figura 6 se muestra la fotografía de un Domo en las instalaciones de la PUCP.



Figura 6 Invernadero Domo PUCP
Elaboración propia

1.3 Importancia del Cultivo en Invernaderos

El fenómeno de cambio climático, por el que está atravesando nuestro planeta, hace que cada año sus consecuencias sean más agudas que el año anterior. Esto afecta a los cultivos que se encuentran al “aire libre”, pues están expuestos a cambios bruscos de temperatura (olas de calor, granizo, heladas, etc), los cuales ocasionan la aparición de plagas mortales que si no son contrarrestadas a tiempo pueden ocasionar la muerte de las plantas.

Como vimos anteriormente, existen cuatro factores (temperatura, humedad relativa, luz y dióxido de carbono) que afectan directamente al cultivo durante su crecimiento (Control Climático en Invernaderos, s.f.). Justamente estos factores son alterados por el fenómeno del

cambio climático. En ese sentido, surge la necesidad de generar mecanismos de adaptación a estos cambios ambientales de manera que no se afecte al desarrollo de las especies vegetales.

Una alternativa de adaptación al cambio climático es el uso de invernaderos para la protección de los cultivos. De esta manera se crea una barrera entre el ambiente exterior, que es muy variante y hostil, frente a un ambiente interior más estable y adaptable a las condiciones para el correcto crecimiento de los cultivos. Gracias a esta atmósfera interior controlable, el invernadero ayuda a producir cultivos fuera de temporada y también a incrementar el rendimiento a través de ciclos vegetativos más cortos, sin descuidar la calidad de los productos.

1.4 Referencias de Uso de Invernaderos en el Mundo

A lo largo de los años han evolucionado las técnicas de protección de cultivos. El antecedente histórico más claro es el uso de camas móviles que se protegían en cabañas durante el invierno, las cuales eran mayormente usadas en el norte de Italia y el sur de Alemania en el siglo XV. Ahora, en los últimos años, los diversos tipos de invernaderos están presentes, mayormente, en Asia y Europa (López Hernández & Pérez-Parra, 2006).



Figura 7 Principales áreas de invernaderos a nivel mundial
Tomado de López Hernández & Pérez-Parra, 2006

Entre los países con referencias tenemos a Portugal, con invernaderos tipo capilla basados en madera barata como el eucalipto; España, con invernaderos tipo parral que en su mayoría se concentran en la provincia de Almería; Francia, con invernaderos tipo túnel que ofrecen buena resistencia al viento; y, en China, invernaderos con paredes de ladrillos en sentido Este-Oeste

para absorber el calor durante el día y cederlo durante la noche (López Hernández & Pérez-Parra, 2006).

1.5 Situación Actual y Tendencias del Cultivo Invernadero en el Perú

De acuerdo a lo que vimos en el apartado anterior, la presencia de invernaderos es más frecuente en países de Europa y Asia. Sin embargo, en los países latinoamericanos, como el Perú, el uso de los invernaderos aún no ha alcanzado niveles importantes de penetración.

La agricultura en el Perú es un sector de gran relevancia económica, el cual genera el 25.8% de empleos para la Población Económicamente Activa Ocupada, siendo así su mayor componente (Ministerio de Agricultura y Riego, 2018). Además, cabe resaltar que, el 5.4% del PBI está cubierto por la actividad agrícola, de la cual depende en gran medida la seguridad alimentaria de todos los peruanos (Instituto Nacional de Estadística e Informática, 2019).

Sin embargo, la agricultura es una actividad sensible a los cambios en el clima. El impacto que puede ocasionar el cambio climático en los cultivos depende de la vulnerabilidad agrícola del lugar geográfico. El IPCC (Intergovernmental Panel on Climate Change) en su informe Cambio Climático 2007: Impactos, Adaptación y Vulnerabilidad, define el concepto de vulnerabilidad como “grado de susceptibilidad o incapacidad de un sistema para afrontar los efectos negativos del cambio climático, incluidos la variabilidad y los fenómenos extremos. La vulnerabilidad está en función del carácter, la dimensión y el índice de variación climática a que está expuesto un sistema, su sensibilidad y su capacidad de adaptación” (p. 29).

En el Perú, el PLANGRACC-A (Plan de Gestión de Riesgos y Adaptación al Cambio Climático en el sector Agrario, período 2012-2021) elaboró un mapa de vulnerabilidad agrícola por distritos a nivel nacional, en el cual se tuvo en cuenta 12 cultivos principales: papa, arroz, maíz amarillo duro, yuca, café, cacao, trigo, plátano, maíz amiláceo, cebada grano, haba grano y frijol grano; y tres principales especies de pastos y forrajes: alfalfa, avena forrajera y brachiaria. Este mapa podemos observarlo en la Figura 8.

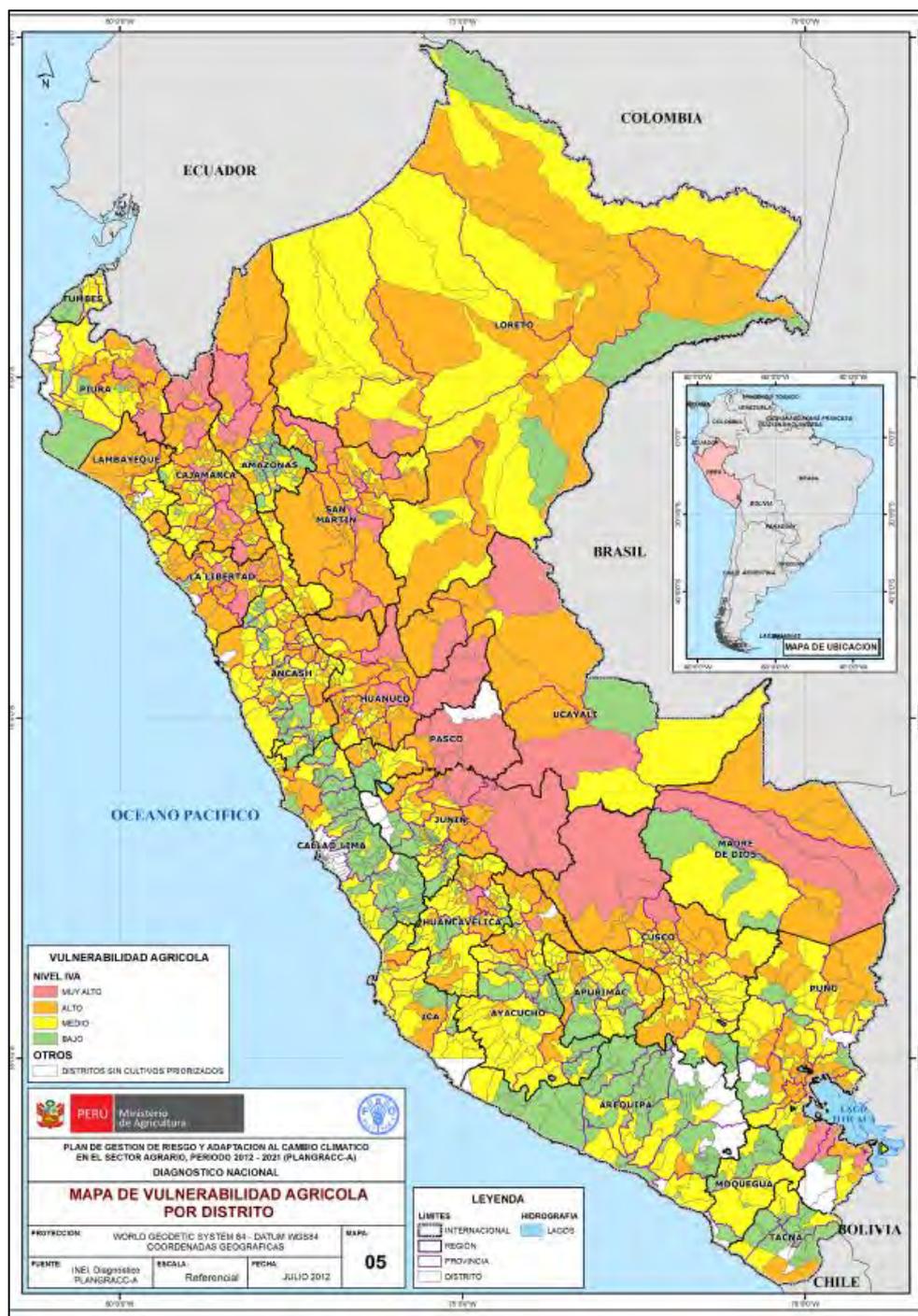


Figura 8 Vulnerabilidad agrícola por distrito
Tomado de PLANGRACC-A, 2012

En el mapa anterior podemos resaltar que los sectores con mayor vulnerabilidad agrícola se encuentran en los límites de la región sierra y selva. Uno de los motivos, que hace que este índice de vulnerabilidad no disminuya, es el déficit de uso de la tecnología en la agricultura, por ejemplo, la instalación de invernaderos con sistemas de automatización y monitoreo. Cabe

aclarar que, la mayoría de los invernaderos al interior del país, son operados manualmente por los agricultores, ya que no poseen un sistema de control de parámetros climáticos ni tampoco cuentan con una plataforma de monitoreo local o remoto, que extraiga los datos y ayude a visibilizar la información climática del invernadero en cualquier parte del mundo a través de Internet.

1.6 Declaración del Marco Problemático

De acuerdo a lo expuesto, el cambio climático altera las variables como la temperatura, humedad relativa, luz y dióxido de carbono, las cuales son vitales para el crecimiento de los cultivos. Esta variación climática genera enfermedades y/o plagas mortales para las plantas. En el Perú, a este fenómeno se suma la diversidad climática de las regiones del país debido a los diferentes pisos altitudinales. De esta manera, se genera la necesidad de buscar alternativas de adaptación al cambio climático, aplicadas a la agricultura, mediante el uso de soluciones tecnológicas.

En ese sentido, en la presente tesis, se propone el diseño de un sistema de monitoreo para un invernadero experimental basado en una red de sensores, la cual ayude a detectar condiciones ambientales que podrían ser optimizadas para favorecer el desarrollo de los cultivos; y así, mejorar la calidad y elevar la productividad.

Capítulo 2. Marco Teórico

2.1 Introducción

Actualmente, los sistemas de monitoreo basados en redes de sensores son la pieza fundamental para la optimización de procesos y automatización. Por un lado, el explosivo crecimiento del desarrollo de software, ha permitido que los sistemas de monitoreo sean construidos sobre la base de aplicaciones web. Por otro lado, la evolución constante de la tecnología de sensores, ha ampliado los tipos de datos que se pueden recolectar, almacenar y procesar, con el objetivo de utilizarlos en diferentes escenarios e industrias. Esto, sumado a la capacidad que nos ofrece Internet, ha potenciado el avance mundial del IoT (*Internet of Things*).

2.2 Redes de Sensores

Las redes de sensores permiten obtener diferentes tipos de datos de acuerdo a la aplicación en la que se utilizan, entre las principales tenemos:

- Medioambiente: medición de variables climáticas
- Seguridad: circuito de cámaras, sensores de movimiento, etc.
- Automotriz: conducción autónoma.
- Aviación y vehículos no tripulados (drones).
- Hogar: domótica.
- Medicina: diagnóstico, tratamiento y control de pacientes.
- Industria y comercio: monitoreo de procesos y máquinas, administración de inventarios, calidad de producto, etc.

2.2.1 Definición

Una red de sensores es un conjunto de dispositivos autónomos y especializados, los cuales se comunican de forma inalámbrica o alámbrica, cuyo objetivo principal es obtener datos físicos como temperatura, humedad, movimiento, presión, etc. La data recolectada es transmitida hasta un punto externo para su procesamiento, almacenamiento y ejecución de acciones programadas. Básicamente, una red de sensores está conformada de los siguientes componentes (Palma Gómez, 2009):

- **Nodo sensor:** responsable de obtener la data que se quiere medir. Posee un sistema de comunicación y, comúnmente, una batería.
- **Gateway:** dispositivo encargado de consolidar los datos recibidos desde los nodos sensores, los cuales serán enviados a una computadora externa para procesarlas, almacenarlas y, finalmente, presentarlas al usuario. Posee conexión de energía eléctrica y, mayormente, salida a Internet.

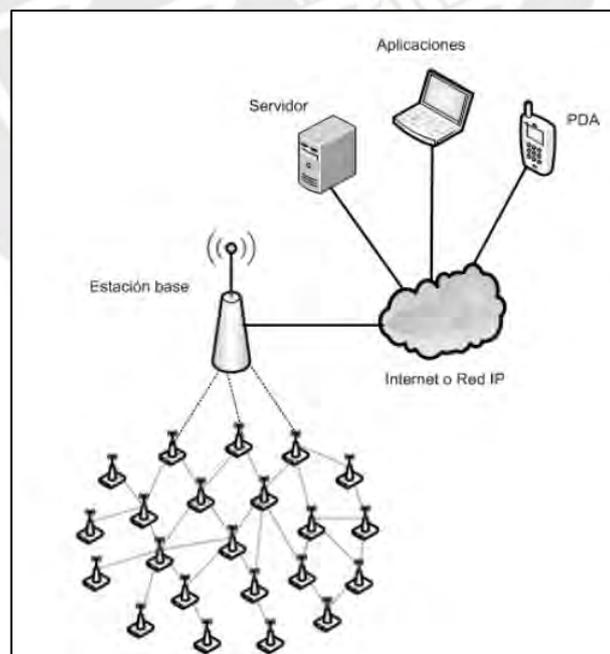


Figura 9 Infraestructura de una red inalámbrica de sensores
Tomado de Garbarino, 2011

2.2.2 Topología

Para la interconexión de los nodos de una red de sensores se tienen las siguientes topologías mayormente utilizadas (Palma Gómez, 2009):

- Estrella: es la más básica. Todos los nodos se conectan directamente al gateway.
- Anillo: referenciado de las redes de fibra óptica. No es muy práctica debido a la distribución espacial.
- Bus: todos los nodos se conectan a un medio común para luego llegar al gateway.
- Árbol: la data viaja desde los nodos extremos, a través de otros nodos, hasta el gateway.
- Mallado con conexión total: la más costosa pero confiable ya que todos los nodos se conectan entre sí formando una malla. Por ejemplo, una red con “n” nodos requiere de $n*(n - 1) / 2$ enlaces directos.
- Mallado con conexión parcial: a diferencia de la topología anterior, no todos los nodos se interconectan entre sí, sino solo algunos. Es menos costosa pero también menos redundante.

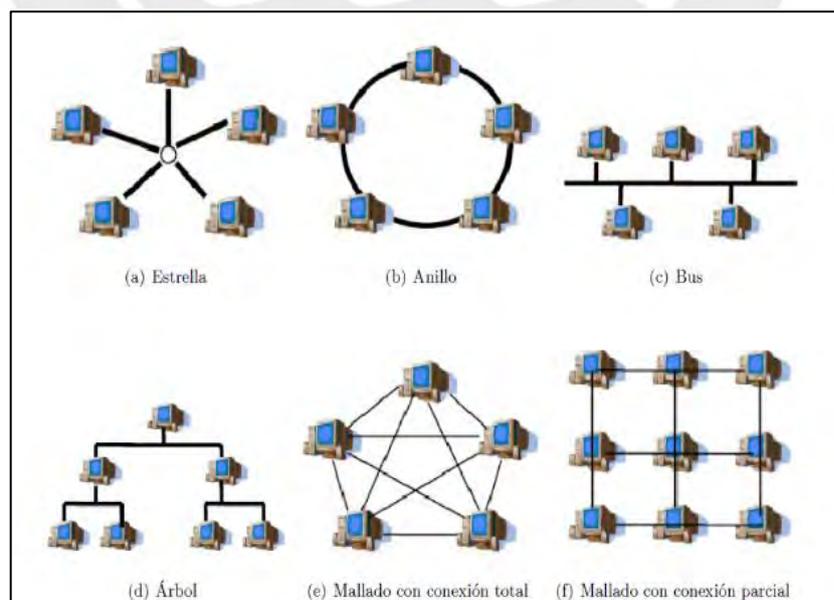


Figura 10 Topologías de red
Tomado de Palma Gómez, 2009

2.2.3 Estándares de Comunicación para Redes de Sensores Inalámbricas

Los estándares de comunicación permiten la integración efectiva entre dispositivos de diferentes fabricantes. En el caso de las redes de sensores inalámbricas, estos estándares están orientados al bajo consumo de energía, lo cual potencia el despliegue masivo de nodos sensores con un período de vida extenso (Suhonen, Kohvakka, Kaseva, Hämäläinen, & Hännikäinen, 2012). En la Tabla 1 podemos observar los estándares de comunicación más destacados en el ámbito de las redes de sensores inalámbricas.

Tabla 1 *Estándares de comunicación para redes de sensores inalámbricas*

Standard	Frequen- cy (MHz)	Data rate (kbps)	Protocol layers					Security	
			PHY	MAC	NWK	TRP	APS	ACL	Encryption
IEEE 802.15.4	868	20	●	●	○	○	○	●	●
	915	40	●	●	○	○	○	●	●
	2400	250	●	●	○	○	○	●	●
ZigBee ¹	-	-	○	○	●	●	●	●	●
6LoWPAN ¹	-	-	○	○	●	○	○	●	●
WirelessHART ²	2400	250	●	●	●	●	●	●	●
ISA100.11a ²	2400	250	●	●	●	●	●	●	●
Z-Wave	865	40	●	●	●	○	●	○	○
	915	40	●	●	●	○	●	○	○
Bluetooth									
Low Energy	2400	1000	●	●	●	●	●	●	●
ANT/ANT+	2400	1000	●	●	●	○	●	●	○
ONE NET	868/	38.4	○	●	●	○	○	●	●
	915	230	○	●	●	○	○	●	●
DASH7	433	27.8	●	●	○	○	○	○	○
IEEE 1902.1 RuBee	0.131	1.2	●	●	○	○	○	○	○

¹ Uses IEEE 802.15.4 as PHY and MAC layers

² Uses IEEE 802.15.4 as PHY layer

Tomado de Low-Power Wireless Sensor Networks, 2012

2.2.3.1 IEEE 802.15.4

El estándar IEEE 802.15.4 está diseñado para aplicaciones con baja exigencia en la tasa de transmisión de datos y que tengan recursos limitados de energía y procesamiento. En consecuencia, es adecuado para redes LR-WPAN “*Low-Rate Wireless Personal Area Network*” como las que se usan en domótica, monitoreo climático, vehículos autónomos, agricultura, etc. (Yang, 2014).

El estándar detalla la capa física y la capa de control de acceso al medio o más conocida como MAC, por sus siglas en inglés. Ambas capas son reutilizadas por otros estándares (ver Figura 11), como ZigBee o 6LoWPAN, debido a su gran popularidad (Suhonen, Kohvakka, Kaseva, Hämäläinen, & Hännikäinen, 2012).

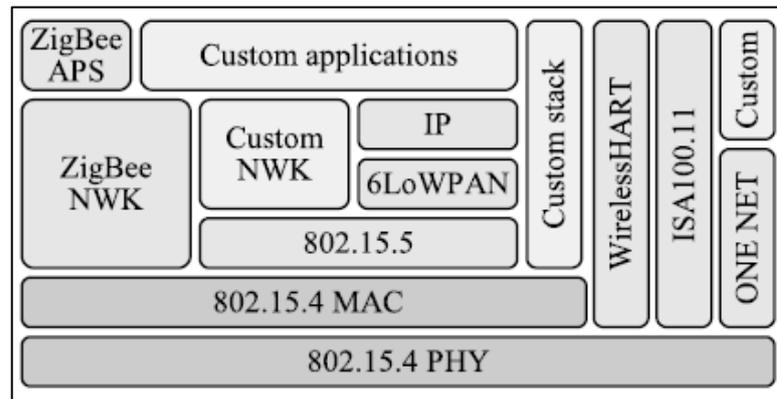


Figura 11 Estándares basados en IEEE 802.15.4
Tomado de Low-Power Wireless Sensor Networks, 2012

Dentro del estándar se define dos tipos de dispositivos: *full-function device* (FFD) y *reduced-function device* (RFD). El FFD se caracteriza por ser un equipo con capacidad de iniciar y administrar la red, a esto se le denomina *PAN coordinator*. Mientras que el RFD es más simple y ligero, ya que ejecuta tareas básicas como obtener las mediciones del sensor para luego enviarlo al FFD (Suhonen, Kohvakka, Kaseva, Hämäläinen, & Hännikäinen, 2012). La topología donde se observa la interacción entre FFD y RFD se muestra en la Figura 12.

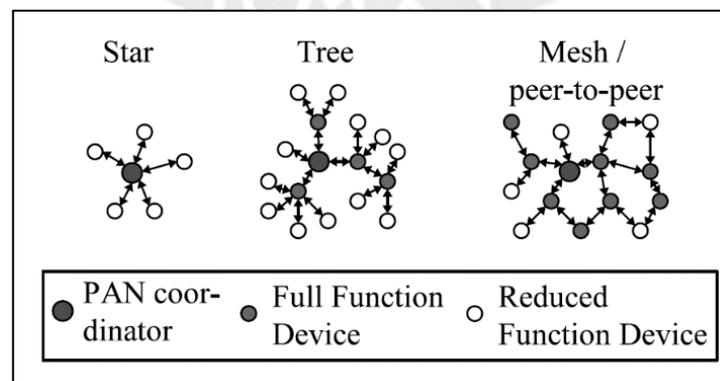


Figura 12 Topologías de red en IEEE 802.15.4
Tomado de Low-Power Wireless Sensor Networks, 2012

2.2.3.2 ZigBee

ZigBee es un estándar enfocado en ofrecer bajo consumo de energía, baja tasa de transmisión de datos, bajo costo y facilidad para su implementación. Fue creado en el 2004 por *ZigBee Alliance* y está construido sobre la base del estándar IEEE 802.15.4, ya que reutiliza la capa física y de control de acceso al medio (Yang, 2014).

La arquitectura de ZigBee se divide en tres secciones: la primera, conformada por la capa física y MAC (IEEE 802.15.4); la segunda, por la capa de red (NWK), la subcapa de soporte de aplicación (APS), el administrador de servicios de seguridad y *ZigBee object* (ZDO); y la tercera que es la capa de aplicación (Yang, 2014). El detalle de esta distribución se observa en la Figura 13.

Por otro lado, las topologías soportadas por ZigBee son las mismas que el estándar IEEE 802.15.4, tales como estrella, árbol y mesh. De igual forma, se tiene un coordinador PAN o coordinador ZigBee y varios dispositivos finales (Yang, 2014).

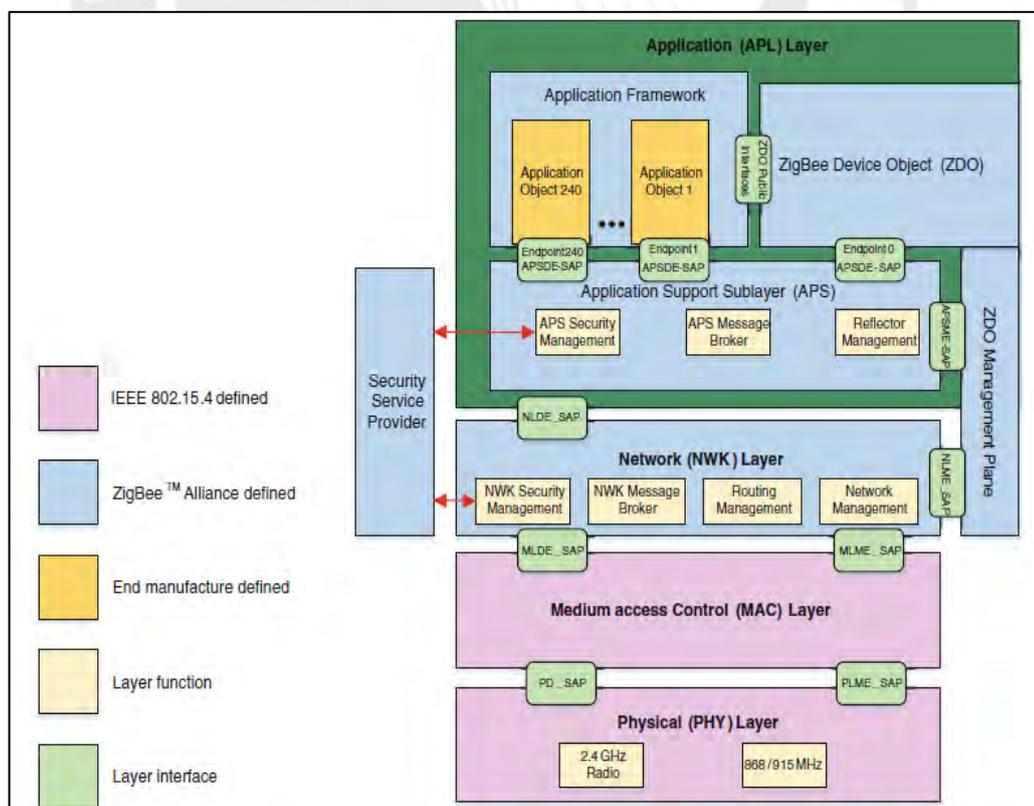


Figura 13 Arquitectura ZigBee
Tomado de Wireless Sensor Networks, 2014

2.2.3.3 6LoWPAN

Las siglas significan *IPv6 over Low Power Wireless Personal Area Networks*; es decir, el estándar define el uso de IPv6 sobre las redes basadas en IEEE 802.15.4. Fue desarrollado por la *Internet Engineering Task Force* (IETF) en el 2007. De esta manera, un nodo dentro de la red de sensores puede salir a Internet o ser alcanzado desde la misma, facilitando el uso de las diferentes aplicaciones basadas en IP. Los principales beneficios de este estándar son los siguientes (Yang, 2014):

- Interoperabilidad: permite el uso de la infraestructura de red existente basado en IP.
- Internet: los dispositivos inalámbricos pueden acceder a Internet fácilmente.
- Full IP: permite el uso de todas las tecnologías basadas en IP. Esto incluye los protocolos como HTTP, SNMP, etc.

Dentro de las diferentes capas del estándar 6LoWPAN, resalta una denominada *adaptation layer* o *LoWPAN layer*, la cual reside entre la capa MAC y la capa de red IPv6. Las funciones principales de esta capa intermedia son: comprimir la cabecera IPv6, fragmentar el payload IPv6 y comprimir la cabecera UDP (Yang, 2014). En la Figura 14 se observa la pila de protocolos completa de 6LoWPAN.

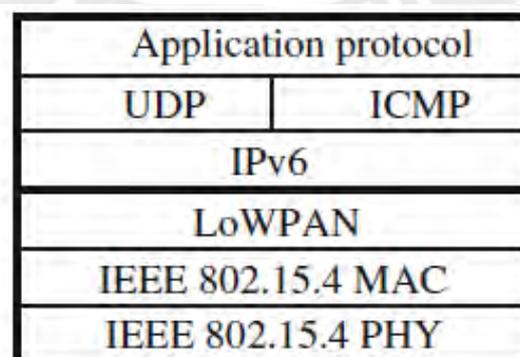


Figura 14 Pila de protocolos 6LoWPAN
Tomado de Wireless Sensor Networks, 2014

2.2.3.4 Bluetooth Low Energy

El estándar Bluetooth Low Energy, o más conocido como BLE, es una variante del ya conocido Bluetooth, orientado a dispositivos inalámbricos de bajo consumo de energía. Sin embargo, no es compatible con Bluetooth, pues utiliza un protocolo distinto en la capa de enlace. Con el objetivo de minimizar el tiempo de actividad de los dispositivos, BLE implementa el modo de ahorro de energía automático cuando el nodo no transmite. Adicionalmente, reduce la complejidad al disminuir la cantidad de estados de conectividad y el formato de los mensajes (Suhonen, Kohvakka, Kaseva, Hämäläinen, & Hännikäinen, 2012).

2.2.3.5 SIGFOX

Sigfox es una empresa que se dedica a masificar la adopción de IoT a través de una red global dedicada, la cual se basa en conceptos como, bajo consumo de energía, alcance amplio y datos pequeños. Propone un estándar, denominado “0G”, que se basa en los siguientes principios (SIGFOX, 2020):

- Proveer un canal mínimo para transferir mensajes pequeños.
- Crear un canal de respaldo “back-up” en caso el enlace principal falle o en casos donde la red no esté disponible por desastres naturales, actos vandálicos, etc.
- Incrementar y garantizar la seguridad de la red y de las transacciones.
- Simplificar el acceso a diferentes redes para incrementar la adopción.
- Reducir el consumo de energía asociado a las redes de telecomunicaciones.

El rango de frecuencia de operación va desde 862 MHz hasta 928 MHz, el cual depende del país. En Perú se utiliza la banda no licenciada 902 MHz-928 MHz con un ancho de banda de 200 kHz. Utiliza un sistema de comunicación Ultra Narrow-Band (UNB) basado en la modulación Differential Binary Phase-Shift Keying (D-BPSK) con

alta eficiencia espectral. El bitrate varía desde 100 bps hasta 600 bps. A esto se suma la definición de un protocolo ligero con cabeceras optimizadas sin uso de señalización constante, por lo que consume menos energía y, en consecuencia, prolonga la vida de la batería de los sensores. Para el sentido uplink permite un payload máximo de 12 bytes (140 mensajes/día) y para el downlink permite un payload máximo de 8 bytes (4 mensajes/día) (SIGFOX, 2020).

Por otro lado, debido a que la energía de este tipo de señal se concentra en una banda estrecha o “narrow band”, hace que sea robusta frente a las interferencias y también permite que sean fácilmente de-moduladas por las estaciones base. La alta sensibilidad de recepción de las estaciones base permite manejar valores que van desde -142dBm hasta -134dBm dependiendo del bitrate (SIGFOX, 2020).

El aspecto de seguridad es una pieza importante dentro de la solución Sigfox. Los dispositivos tienen un ID único y una llave privada que sirve para autenticarlos. Las estaciones base tienen una conexión punto a punto con la plataforma Sigfox Cloud a través de una red privada virtual (VPN) encriptada. Finalmente, para la conexión entre Sigfox Cloud y la plataforma cliente se usa el protocolo seguro HTTPS (SIGFOX, 2020). En la Figura 15 se muestra la arquitectura de una red SIGFOX.

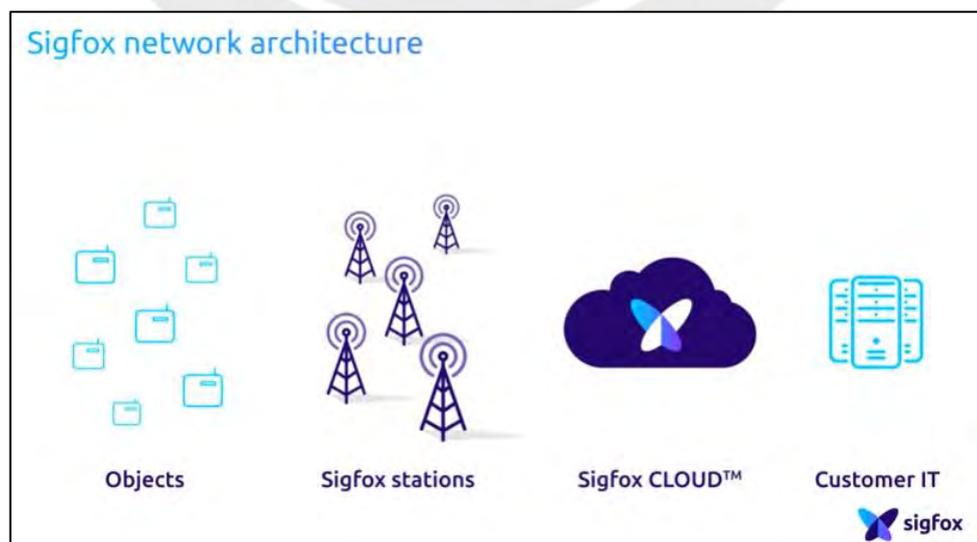


Figura 15 Arquitectura de red SIGFOX
Tomado de SIGFOX, 2020

2.2.3.6 LORA

LoRa, del inglés *long range* o largo alcance, es una tecnología inalámbrica de capa física, la cual implementa la modulación *Chirp Spread Spectrum* de bajo consumo, tanto de energía como ancho de banda, pero con capacidad de crear comunicaciones a grandes distancias. Sobre la base de LoRa se ha construido LoRaWAN como un protocolo de tipo *Low Power Wide Area Network* (LPWAN) que permite conectar a dispositivos, de energía limitada, con aplicaciones alojadas en Internet. Actualmente, la especificación y mantenimiento de este protocolo está a cargo de LoRa Alliance, una asociación abierta y sin fines de lucro (LoRa Alliance, 2021).

La arquitectura de LoRaWAN está conformada principalmente por cuatro entidades, con cifrado AES de extremo a extremo, tal como se muestra en la Figura 16. Los dispositivos o nodos necesitan un módulo LoRa que opere en la banda de frecuencia no licenciada, 902 MHz-928 MHz, para el caso de Perú. Los nodos se conectan a múltiples gateways en forma de estrella y transmiten sus datos con bitrate limitado entre 0.3kbps-50kbps. El gateway posee una alta sensibilidad en sus antenas para recibir los datos y reenviarlo al servidor central a través de una conexión de banda ancha tipo Ethernet, WiFi, móvil, etc. La inteligencia y complejidad está a cargo del servidor central que está alojado típicamente en un entorno Cloud. El servidor central se encarga de administrar la red, realizar las verificaciones de seguridad y filtrar los paquetes redundantes para reenviarlos a las aplicaciones correspondientes; así como también, retornar la comunicación hacia los nodos. Finalmente, las aplicaciones son las encargadas del procesamiento y análisis de los datos; incluso, se tiene la posibilidad de integrarse a otras plataformas Cloud tipo AWS o Azure (The Things Network, 2021).

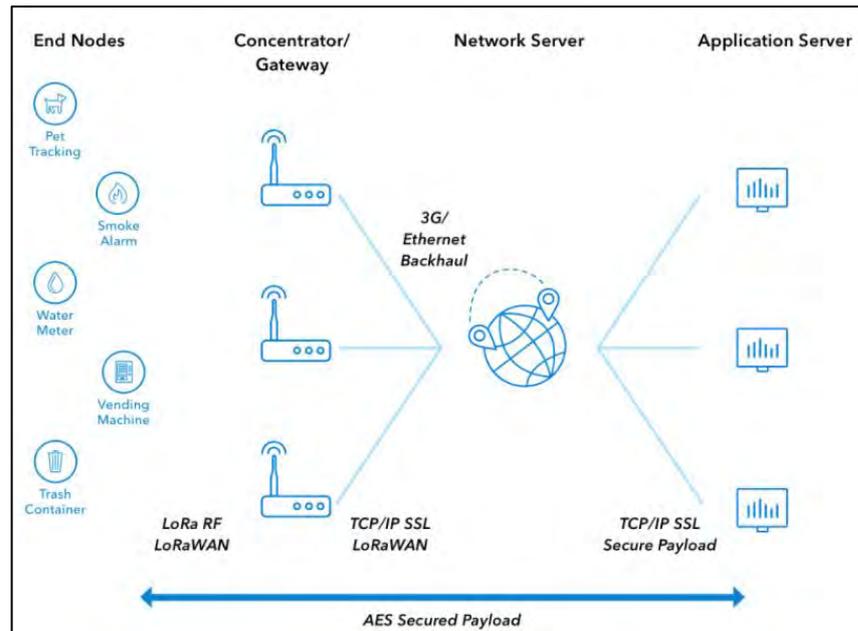


Figura 16 Arquitectura LoRaWAN
Tomado de The Things Network, 2021

2.2.4 Principales Fabricantes de Hardware

- **Arduino:** fabricante mundialmente conocido en el desarrollo de hardware y software para aplicaciones de bajo costo donde se necesite simplicidad. Arduino es considerado una plataforma electrónica Open Source de fácil uso para los principiantes y flexible para los usuarios avanzados (Arduino, 2020).
- **Raspberry:** fabricante de hardware y software para distintos tipos de aplicaciones. Su producto principal es el Raspberry Pi, el cual es una micro-computadora muy flexible que se puede usar, entre otras cosas, para controlar redes basadas en sensores (Raspberrypi, 2020).
- **Texas Instruments:** mundialmente conocido por la fabricación de gran cantidad de dispositivos electrónicos, tales como sensores, microcontroladores y hasta kits completos de IoT (Texas Instruments, 2020).
- **Libelium:** fabricante de hardware y software para redes de sensores inalámbricas. Su producto principal es Waspote, un dispositivo integrado donde se pueden conectar

sensores de acuerdo a la aplicación y transmitir la información a una red de datos externa (Libelium, 2020).

- ATIM: fabricante de sensores para redes IoT que destacan por su característica de bajo consumo de energía o más conocidas como “Low Power Wide Area Network” (LPWAN) (ATIM, 2020).

2.3 Monitoreo

El crecimiento continuo de Internet y el desarrollo de software han abierto el camino para la implementación de sistemas de monitoreo sobre la base de aplicaciones y servicios web. De esta manera, los datos recolectados por la red de sensores pueden estar disponibles en cualquier parte del mundo a través de Internet. La información elaborada a partir de estos datos puede servir, entre otras cosas, para detectar condiciones, tomar acciones inmediatas y también para modelar y/o predecir comportamientos.

El enfoque de Choi y Rhee (2012) denomina a la plataforma web como *Semantic Sensor Web Platform*, la cual interactúa con los gateways de las redes de sensores y con los usuarios finales a través de servicios web tipo REST que explicaremos más adelante. En la Figura 17 se observa el esquema planteado por los autores.

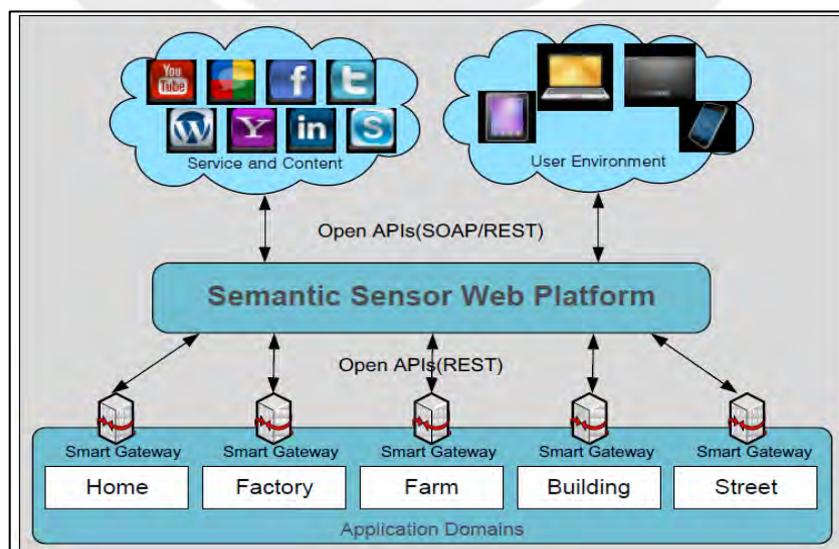


Figura 17 Esquema general del entorno de servicios web de sensores
Tomado de Choi y Rhee, 2012

2.3.1 Arquitectura de Software para Aplicaciones Web

De acuerdo a los conceptos de la Ingeniería de Software, el software posee un ciclo de vida, el cual comprende las siguientes etapas secuenciales: recopilación de requerimientos, diseño, desarrollo, validación, implementación y mantenimiento. Durante la etapa del diseño, entre otras cosas, se define la arquitectura o patrón arquitectónico de software a utilizar. En ese sentido, la arquitectura de software es considerada como un medio para documentar las decisiones de diseño de alto nivel. Además, nos ayuda a entender de manera más clara los componentes de software que cubrirán las necesidades recogidas en la primera etapa (Mora, 2011). A continuación, se detalla los patrones arquitectónicos de software recomendados para las aplicaciones web.

2.3.1.1 Arquitectura en Capas

Sommerville (2011), en su libro Ingeniería de Software, describe a este tipo de arquitectura como la que “organiza el sistema en capas con funcionalidad relacionada con cada capa. Una capa da servicios a la capa de encima, de modo que las capas de nivel inferior representan servicios núcleo que es probable se utilicen a lo largo de todo el sistema” (p.158). En la Figura 18 se muestra el ejemplo de una arquitectura en capas con cuatro componentes, donde la capa base incluye el software de soporte al sistema, tipo base de datos y sistema operativo.

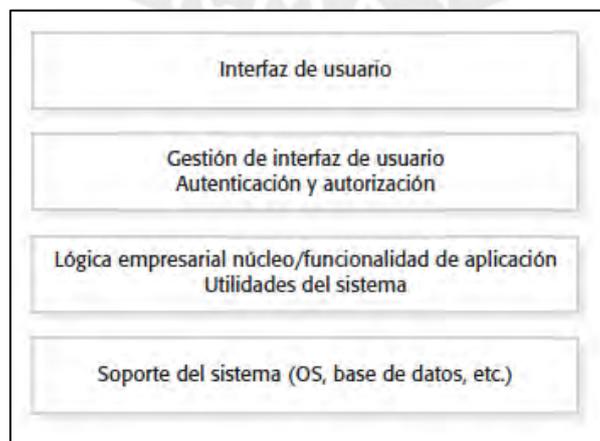


Figura 18 Arquitectura genérica en capas
Tomado de Sommerville, 2011

2.3.1.2 Arquitectura Cliente – Servidor

Este tipo de arquitectura maneja dos componentes. Primero, el servidor, el cual administra el (los) servicios garantizando su disponibilidad. Segundo, el cliente, el cual es un usuario de dichos servicios y que para utilizarlos necesita tener conectividad con el (los) servidor(es). Una aplicación web que implemente este tipo de arquitectura debe tener en cuenta sus principales componentes (Sommerville, 2011):

- Un conjunto de servidores que ofrecen servicios.
- Un conjunto de clientes que solicitan los servicios que ofrecen los servidores.
- Una red que permite a los clientes acceder a dichos servicios.

En la Figura 19, podemos observar un ejemplo de la arquitectura cliente-servidor donde interactúan los componentes descritos anteriormente.

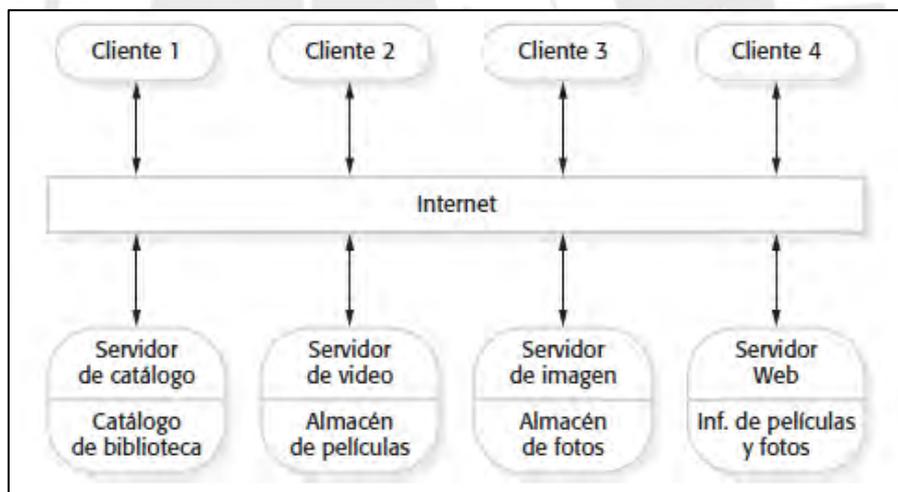


Figura 19 Arquitectura cliente-servidor para una filmoteca
Tomado de Sommerville, 2011

2.3.1.3 Arquitectura Modelo Vista Controlador (MVC)

Esta arquitectura es la más popular al ser extensamente utilizada en las aplicaciones web. El principio que rige este patrón se resume en la frase “divide y vencerás”; es decir, se puede separar la funcionalidad principal del negocio (más conocida como *core business*) de la presentación. Esto debido a que la interfaz de usuario varía mucho más

rápido que la lógica núcleo. Por ello, la arquitectura MVC propone dividir la aplicación en tres capas o niveles, donde un cambio en una de ellas no genere impacto elevado en la otra (Sommerville, 2011):

- **Modelo:** Representa la información de la aplicación; es decir, el Modelo es el objeto que representa los datos del programa, los administra y controla todas sus transformaciones. No tiene conocimiento ni referencias hacia las otras capas.
- **Vista:** Esta capa es la encargada de presentar la información al usuario. Ingresa a los datos a través del Modelo y especifica cómo debe mostrarse. La vista es la responsable de mantener la consistencia en la presentación cuando cambia el Modelo. Entre algunos ejemplos de Vistas tenemos a la interfaz gráfica de las aplicaciones móviles/desktops o el código HTML visto desde el navegador.
- **Controlador:** Soporta la lógica de la aplicación al traducir las interacciones con la Vista en acciones que deben ser ejecutadas por el Modelo. En una aplicación web, las interacciones aparecen como peticiones HTTP tipo GET o POST. Las acciones realizadas por el Controlador pueden incluir cambios de estado en el Modelo.

A continuación, se elaboró la Figura 20, donde se observa el flujo de la información entre los componentes de una arquitectura MVC:

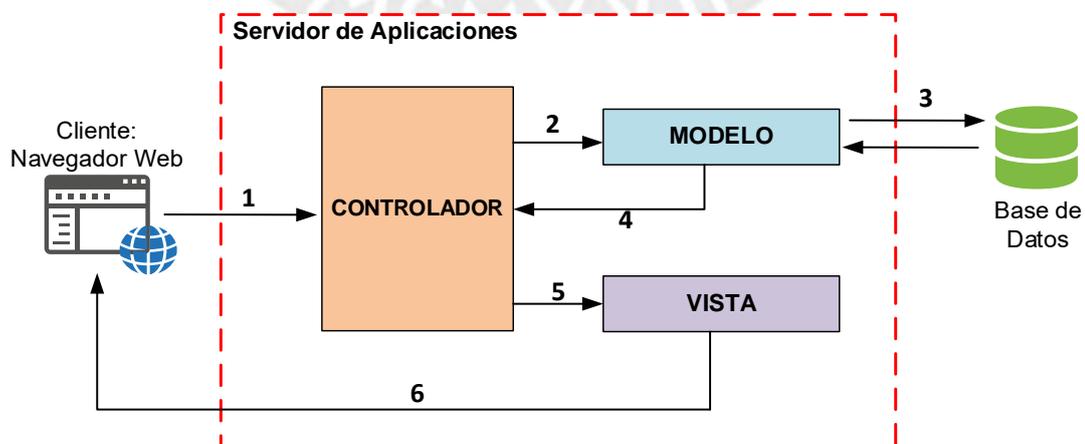


Figura 20 Arquitectura MVC
Elaboración propia basado en Sommerville, 2011

El flujo de la información es de la siguiente manera:

1. El usuario, desde su navegador web, interactúa al enviar una petición al Controlador.
2. El Controlador procesa la interacción y lo traduce en una acción para el Modelo.
3. El Modelo ejecuta la acción solicitada por el Controlador. En este caso es una operación de lectura/escritura en la base de datos.
4. El Modelo devuelve el resultado de la ejecución al Controlador.
5. El Controlador procesa este resultado y selecciona la Vista adecuada para ser enviada al cliente.
6. La vista seleccionada es enviada al cliente.

2.3.2 Servicios Web

El crecimiento acelerado de Internet va de la mano con el desarrollo de nuevos paradigmas de software. Entre los cuales destaca el concepto de “servicios web”, el cual aprovecha los estándares y protocolos disponibles para facilitar la interoperabilidad de las aplicaciones en la red, en especial, sobre Internet.

Según *World Wide Web Consortium (W3C)*, la definición correspondiente es "sistema de software diseñado para apoyar la interacción de máquina a máquina sobre una red interoperable. Tiene una interfaz descrita en un formato procesable por máquina (específicamente WSDL)". En otras palabras, los servicios web son un conjunto de tecnologías estándares de software, que pueden estar desarrollados en diferentes tipos de lenguajes, pero con el objetivo de asegurar el intercambio de datos entre aplicaciones (Morales Machuca, 2014).

2.3.2.1 Stack de Protocolos de los Servicios Web

De acuerdo a lo mencionado anteriormente, los servicios web utilizan estándares y protocolos para lograr la interoperabilidad entre las aplicaciones. El conjunto de estos

estándares y protocolos son agrupados en una pila denominada en inglés *Web Services Protocol Stack* (ver Figura 21).

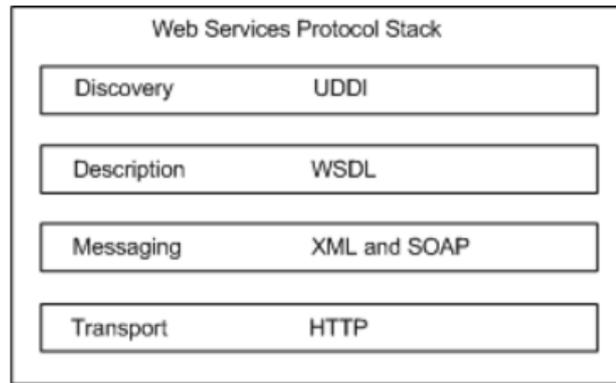


Figura 21 Capas de la pila de protocolo de Web Services
Tomado de Oracle, 2020

Como se observa en la figura anterior, existen cuatro capas. A continuación, una breve explicación de cada una (Morales Machuca, 2014):

- **Transporte:** Como su propio nombre lo describe, la capa de transporte se encarga del transporte de los mensajes entre aplicaciones. Incluye varios protocolos tales como HTTP (*HyperText Transfer Protocol*), FTP (*File Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), JMS (*Java Message Service*).
- **Mensajería:** Es el conjunto encargado de la codificación de los mensajes en un estándar que pueda ser interpretado en cualquiera de los nodos, por ejemplo, XML, JSON. Entre los componentes más utilizados en esta capa se encuentran: REST (*Representational State Transfer*), RPC (*Remote Procedure Calls*), XML (*eXtended Markup Language*) y SOAP (*Simple Object Access Protocol*).
- **Descripción:** Esta capa permite que el servicio web cuente con una interfaz pública descrita por un formato denominado WSDL (*Web Services Descripción Languages*), el cual es un documento tipo XML que describe lo que hace, su ubicación y la forma

de invocar al servicio web. Adicionalmente, expone información importante como la descripción del formato de los mensajes y a cuáles puede responder.

- **Descubrimiento:** Esta capa permite a los propietarios de los servicios web publicarlos en el registro denominado UDDI (*Universal Description Discovery and Integration*). Este registro es accesible como otros servicios web, por lo que los clientes pueden buscarlo para extraer los detalles del servicio y su funcionalidad.

2.3.3 Base de Datos

Los datos recolectados por la red de sensores, luego de ser recibidos por la aplicación web, tienen que ser almacenados en otro elemento denominado Base de Datos (BD), a donde tendrán acceso algunos usuarios definidos. Al respecto, Luis Valencia (2014), de la Universidad de Sevilla, define una BD como un “conjunto de datos que modelan hechos y objetos de una parcela de la realidad y sirven de soporte a una aplicación informática. Dichos datos deben estar almacenados físicamente en forma de ficheros informáticos y deben estar relacionados entre sí mediante una determinada estructura lógica” (p.20). Para complementar, es importante diferenciar los siguientes conceptos y tener claro las propiedades de una BD (Valencia Cabrera, 2014):

- **Datos:** valores que representan hechos o realidades de un dominio específico.
- **Información:** representa el significado de los datos y sirve para que los usuarios puedan sacar conclusiones, tomar decisiones y mejorar procesos.
- **Propiedades:** son tres principalmente; primero, representa un aspecto del mundo real y sus cambios asociados; segundo, es una colección de datos congruentes cuyo significado lo da su creador y/o los usuarios; tercero, está diseñada, implementada y poblada con datos para un fin específico.

2.3.3.1 Modelos de Base de Datos

El modelo de BD permite definir la forma lógica en la que estarán organizados los datos. Con el uso de herramientas, tipo tablas o árboles, se construye el modelo de la realidad (Camps Paré, y otros, 2005).

A lo largo de los años, junto con la evolución del desarrollo de software, han aparecido diferentes enfoques de modelos de BD. Actualmente, resaltan cuatro modelos: jerárquico, red, relacional y no relacional. El uso del último modelo, no relacional, ha aumentado debido al crecimiento de Internet y los volúmenes de información que circula en la red. Se elaboró la Tabla 2 donde se detalla las características principales de cada modelo (Camps Paré, y otros, 2005) (Lith & Mattsson, 2010).

Tabla 2 Principales características de los modelos de Base de Datos

Modelos de BD			
Jerárquico	Red	Relacional	No Relacional
Su estructura consta de registros relacionados en forma de árboles.	Al igual que el modelo jerárquico, también existe registros e interrelaciones, pero ahora un registro ya no está limitado a ser “hijo” de un solo registro tipo.	Basado en el paradigma Entidad-Relación (E/R). Cualquier sistema u objeto puede ser representado mediante entidades (tablas con atributos) y su relación entre ellas. Para la gestión se utiliza mayormente el lenguaje SQL (<i>Structured Query Language</i>).	Mayormente usado para aplicaciones con tratamiento de grandes volúmenes de datos. Carece de esquemas fijos como tablas. Su escalamiento es horizontal.

Elaboración propia

2.3.3.2 Arquitectura de Base de Datos

Se plantea cuatro tipos de arquitecturas para una Base de Datos (Olarte, 2014).

- i. Centralizada
- ii. Cliente-Servidor

- iii. Paralela
- iv. Distribuida

Se elaboró la Tabla 3 donde se muestra las ventajas y desventajas de cada arquitectura.

Tabla 3 *Ventajas y desventajas de las arquitecturas de Base de Datos*

Arquitecturas de Base de Datos				
	Centralizada	Cliente-Servidor	Paralela	Distribuida
Ventajas	<ul style="list-style-type: none"> - Administración centralizada. - Reducción de puntos de falla al estar toda la data concentrada en un solo servidor. - Solución de problemas focalizado. 	<ul style="list-style-type: none"> - Reparte las tareas entre los clientes y el servidor. - Múltiples clientes acceden a un mismo recurso en el servidor. - Diversidad de interfaces de comunicación (APIs) entre cliente/servidor 	<ul style="list-style-type: none"> - Tolerante ante fallos al contar con múltiples recursos de memoria y disco. - Independencia de operaciones de E/S. - Escalable: aumentar más recursos en paralelo. 	<ul style="list-style-type: none"> - Datos compartidos: los usuarios de cada nodo pueden acceder a los datos de otros nodos. - Autonomía: la administración de cada nodo es local. - Disponibilidad: los nodos que son réplicas permiten que el sistema no se detenga totalmente frente a un incidente.
Desventajas	<ul style="list-style-type: none"> - No permite la redundancia geográfica. - No recomendable para aplicaciones con bajo RTO (Recovery Time Objective). 	<ul style="list-style-type: none"> - Vulnerabilidad de sobrecarga de procesamiento causado por múltiples clientes conectados. 	<ul style="list-style-type: none"> - Variación de velocidades de acceso hacia los recursos paralelos. - Incremento de costos por cada nodo paralelo. 	<ul style="list-style-type: none"> - Complejidad en los mecanismos de distribución de los datos. - Inversión inicial elevada e incremento de OPEX por cada nodo.

Elaboración Propia

2.3.3.3 Sistema de Gestión de Base de Datos

El término equivalente en inglés es *Data Base Management System* (DBMS). Luis Valencia (2014), de la Universidad de Sevilla, define el DBMS como “una aplicación

informática que permite a los usuarios definir, construir, mantener y consultar una base de datos, proporcionando un acceso controlado a la misma”. En ese sentido, el DBMS facilita los siguientes cuatro procesos entre el usuario y la Base de Datos:

- i. **Definir:** especificar los tipos de datos, las estructuras y las restricciones a través del lenguaje especializado llamado *Data Definition Language* (DDL).
- ii. **Construir:** almacenar los datos en la Base de Datos.
- iii. **Mantener:** rutinas de actualización, eliminación o inserción de datos a través del lenguaje especializado llamado *Data Management Language* (DML).
- iv. **Consultar:** recuperar información según criterios variados tipo selección, filtrado, agrupación, ordenación, etc, con el objetivo de mostrarlo al usuario. También hace uso del lenguaje especializado DML.

A continuación, se lista los DBMS más comunes y de mayor uso en la actualidad:

- MS Access (Microsoft)
- DB2 (IBM)
- Oracle (Oracle)
- SQL Server (Microsoft)
- MySQL (Oracle)
- Postgres SQL
- SQLite
- Redis (NoSQL)
- Cassandra (NoSQL)
- MongoDB (NoSQL)
- Altibase
- SAP Sybase ASE
- Teradata
- FileMaker
- AmazonRDS
- Neo4j
- OrientDB
- phpMyAdmin
- SQL Developer
- Seqel PRO
- Hadoop HDFS
- Cloudera
- MariaDB

2.3.4 Plataformas Cloud Computing

El concepto de *Cloud Computing* refiere a un conjunto de servicios de computación que son ofrecidos a través de una red privada, pública (Internet) o una mezcla de ambas. En el mercado informático actual podemos encontrar diversas opciones de plataformas Cloud que ofrecen servicios IoT, los cuales están orientados a la recepción, almacenamiento y procesamiento de los datos generados por las redes de sensores. A continuación, se presenta los servicios IoT de las plataformas Cloud con mayor presencia a nivel mundial.

Microsoft Azure

Azure es el producto de Microsoft que nos ofrece un conjunto de servicios en su propia nube o Cloud. Uno de estos servicios, denominado *Azure IoT Suite*, está orientado al Internet de las Cosas (IoT). Lo que promete este servicio es asumir la complejidad de una solución integral a través de la implementación de diversos servicios que conforman la suite; y, de esta manera, empezar rápidamente un proyecto IoT.

Azure IoT Suite permite conectar dispositivos con la plataforma Cloud de Microsoft. Estos dispositivos pueden ser desde nodos gateway de las redes de sensores hasta sistemas embebidos con conexión a Internet. Cabe aclarar que las aplicaciones programadas en los dispositivos deben usar las librerías disponibles en *Azure IoT device SDK*. El protocolo de comunicación más usado es HTTPS, pero también soporta otras como AMQP y MQTT. La cantidad de mensajes por día (desde y hacia los dispositivos) y el tamaño de cada uno (máximo uplink/dowlink 256 KB/64 KB) determinará el costo de nuestra solución (Microsoft Azure, 2019).

La arquitectura de la solución de *Azure IoT Suite* consta de varios servicios con funcionalidades específicas. Entre las cuales destacan: IoT Hub (Cloud Gateway), servicio gestionado que concentra la comunicación bidireccional entre los dispositivos y

la aplicación principal; IoT Central, plataforma gestionada que maneja la lógica principal de la aplicación IoT y tiene como principal componente *Analytics & Machine Learning*, encargado del análisis de la data almacenada en la plataforma para automatizar acciones y/o modelar comportamientos de acuerdo a la aplicación (Microsoft Azure, 2018). Podemos ver la arquitectura en la Figura 22.

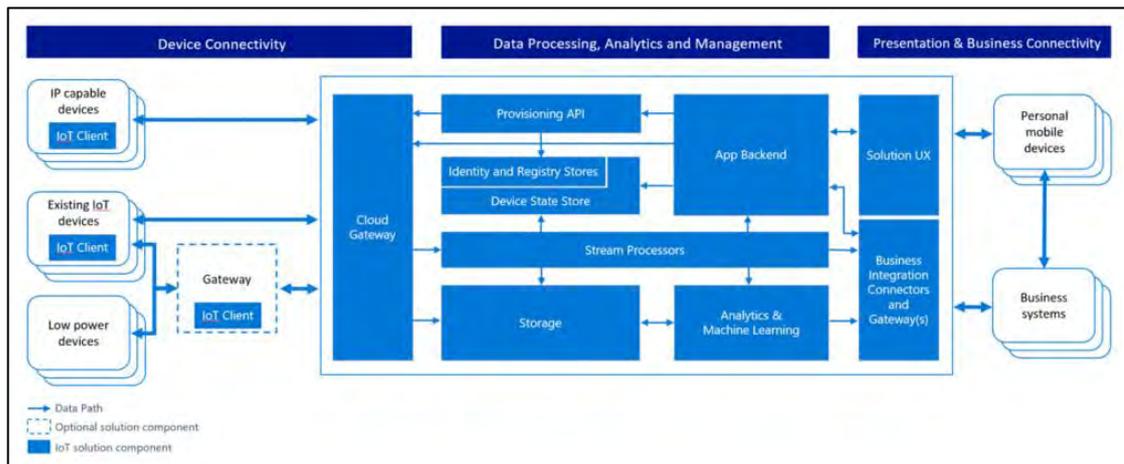


Figura 22 Arquitectura Azure IoT Suite
Tomado de Microsoft Azure, 2020

Amazon Web Services

Amazon Web Services o más conocido por sus siglas como AWS es una plataforma Cloud que ofrece una gran variedad de servicios y, además, cuenta con su propia infraestructura distribuida a nivel mundial. Dentro de su cartera IoT ofrece servicios de control y análisis, así como también, software para dispositivos IoT.

Los dispositivos pueden ser nodos gateway o sensores que tengan capacidad de conectarse a Internet. AWS pone a disposición FreeRTOS, el cual es un sistema operativo de código abierto que puede ejecutarse en los dispositivos IoT para facilitar la conexión con los servicios en la nube de AWS. Otra opción es que los dispositivos implementen las librerías disponibles en *AWS IoT Device SDK*. Al igual que Azure, el protocolo de comunicación más usado es HTTPS, pero también soporta MQTT. El tamaño máximo de los mensajes desde y hacia los dispositivos es de 128 KB, los cuales se contabilizarán

en grupos de 5 KB para la facturación. A esto se le suma el costo por el tiempo de conexión de los dispositivos a la nube de AWS (Amazon Web Services, 2020).

Entre los servicios de control IoT, que ofrece AWS, destacan tres: *AWS IoT Core*, servicio gestionado que permite conectar los dispositivos con la nube y entre ellos mismos; *AWS IoT Device Management*, facilita las tareas de gestión remota de los dispositivos IoT; y *AWS IoT Defender*, servicio gestionado que permite auditar continuamente las configuraciones y monitorear el comportamiento de los dispositivos para detectar vulnerabilidades de seguridad. Por otro lado, AWS también ofrece servicios de análisis, donde destaca *AWS IoT Analytics*, el cual puede ejecutar análisis de grandes volúmenes de datos para obtener información con la que se pueda tomar decisiones más precisas y definir casos de aprendizaje automático (Amazon Web Services, 2020). En la Figura 23 se observa un ejemplo de una solución AWS IoT con sensores y servicios en la nube de AWS.

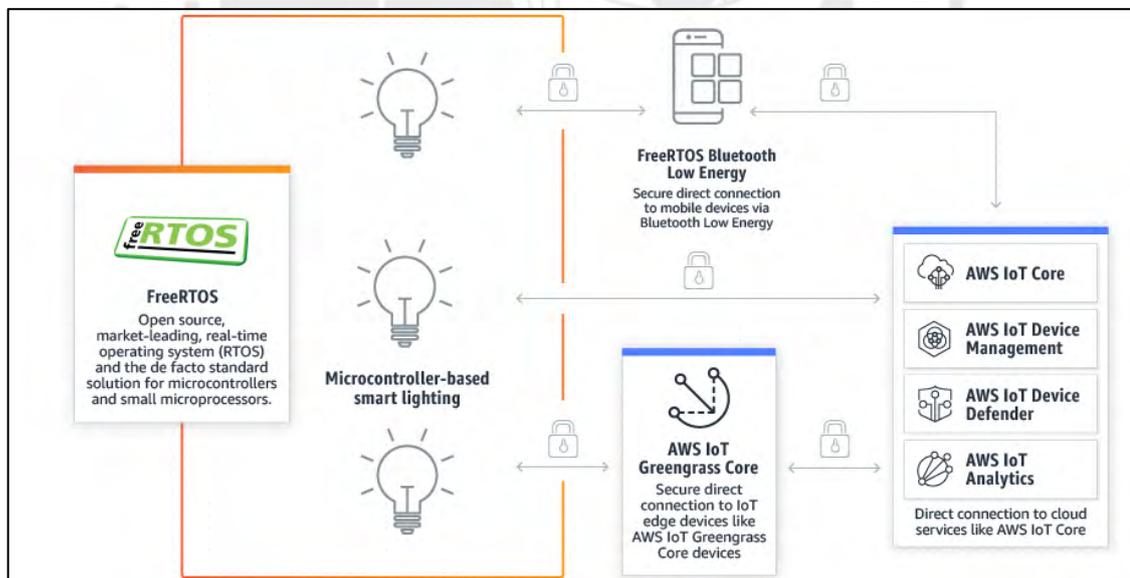


Figura 23 Ejemplo de Solución AWS IoT
Tomado de Amazon Web Services, 2020

Google Cloud

Google también forma parte de la industria de Cloud, donde nos ofrece una gama de servicios alojados en sus propios centros de datos repartidos alrededor del mundo.

Google Cloud IoT es el producto que posee un conjunto de herramientas gestionadas y escalables para conectar dispositivos; además de procesar, almacenar y analizar los datos generados.

Los dispositivos que deseen conectarse a Google Cloud pueden implementar las librerías disponibles en *Google Cloud IoT Device SDK* para comunicaciones a través del protocolo MQTT; o también, pueden usar las REST API para comunicaciones a través del protocolo HTTPS. El tamaño límite de los mensajes intercambiados es de 256 KB. A diferencia de otras soluciones, Google solo factura por el volumen de datos intercambiados entre los dispositivos y el Cloud, sin importar la cantidad de mensajes ni el tiempo de conexión (Google Cloud, 2020).

Los servicios Cloud IoT más destacados que Google pone a nuestra disposición son: *Cloud IoT Core*, encargado de conectar, administrar y recibir datos desde los dispositivos de forma segura; *Cloud Pub/Sub*, servicio de almacenamiento temporal y entrega de mensajes a otros servicios o dispositivos; *BigQuery*, servicio de almacenamiento seguro de grandes volúmenes de datos y acceso inmediato a través de consultas SQL; *Cloud Machine Learning Engine*, servicio de ejecución de analítica avanzada y aprendizaje automático para apoyar en la toma de decisiones y predecir comportamientos según la aplicación (Google Cloud, 2020). En la Figura 24 se observa un esquema genérico de la interacción de los servicios IoT de Google.

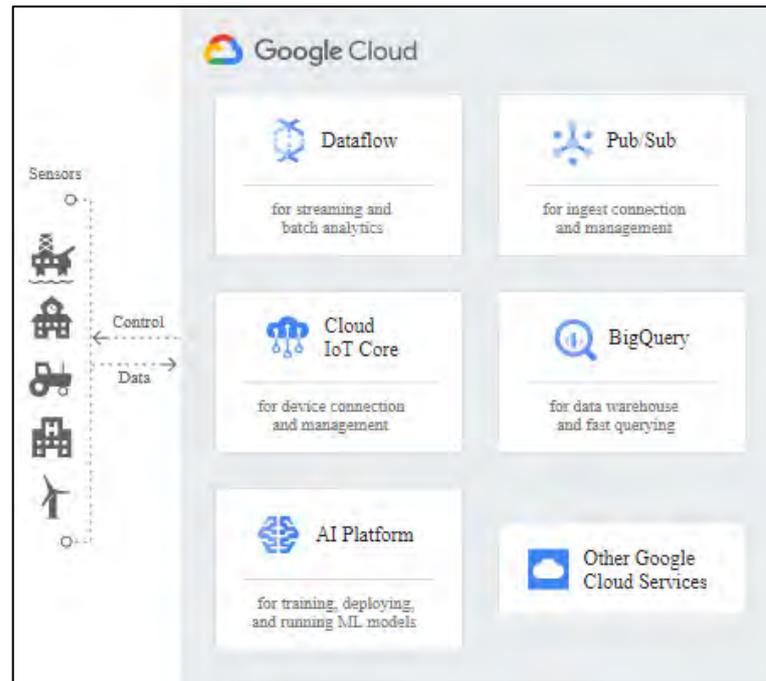


Figura 24 Servicios IoT de Google
Tomado de Google Cloud, 2020

Ubidots

Ubidots es una empresa joven que inició sus operaciones el 2012 y, a lo largo de los años, ha innovado en los servicios IoT que pone a disposición de los estudiantes, investigadores y empresas. Ofrece dos plataformas: Ubidots STEM y Ubidots. La primera, para uso no-comercial, está orientada a estudiantes e investigadores con fines educativos para alcanzar una mayor adopción de IoT. La segunda, orientada a aplicaciones comerciales que necesitan estabilidad y herramientas avanzadas.

Para lograr que los dispositivos se conecten al Cloud de Ubidots deben implementar las librerías SDK disponibles e integrarlas a través de HTTP, MQTT, TCP o UDP. Otra opción es el desarrollo de un API con métodos GET o POST. Para fines de facturación, un dispositivo puede contener hasta 20 variables o sensores y, la cantidad de *dots*, o datos con valores, que se pueden enviar/recibir dependerá del plan contratado. Existe un límite de cuatro transacciones por segundo con un máximo de 10,000 bytes (Ubidots, 2020).

Entre los servicios más destacados que ofrece Ubidots tenemos: *Analytics Engine*, para procesar los datos a través de fórmulas y estadísticas con el objetivo de aportar a la toma de decisiones; *Time Series Backend*, encargado de recibir, almacenar y entregar los datos cuando aplique; *Events Engine*, para implementar alertas vía SMS, email y apps como Telegram y Slack; *Branding*; para personalizar la interfaz gráfica (Ubidots, 2020).

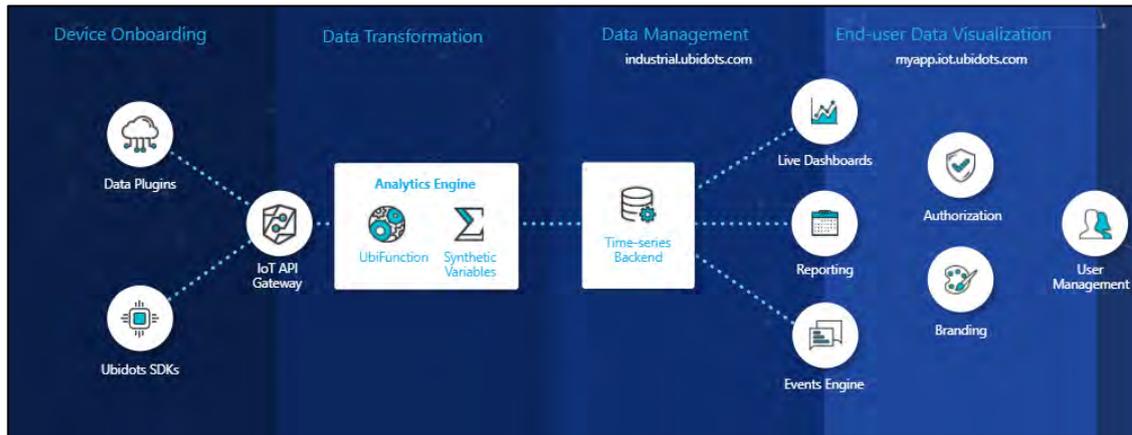


Figura 25 Solución IoT Ubidots
Tomado de Ubidots, 2020

2.4 Modelo Teórico

De acuerdo a lo revisado en el presente capítulo, el modelo teórico propuesto para el presente estudio está estructurado principalmente por dos componentes. Se elaboró la Figura 26 donde se observa la interacción de ambos componentes.

i. La Red de Adquisición de Datos

Conformada por la red de sensores, los cuales medirán las variables climáticas de interés dentro del invernadero y luego enviarán los datos al servidor web principal. Por otro lado, también podrá recibir las respuestas del servidor con las órdenes que deben ejecutar los actuadores, por ejemplo, abrir o cerrar una ventana.

ii. La Red de Distribución de Datos

Conformada por el servidor web y la Base de Datos. Validará la identidad de los sensores y recibirá los datos medidos para ser almacenados. Finalmente, los datos serán procesados para

transformarlos en información, la cual será consumida por los clientes de la aplicación web (smartphones, tablets, laptops, etc.) a través de Internet.

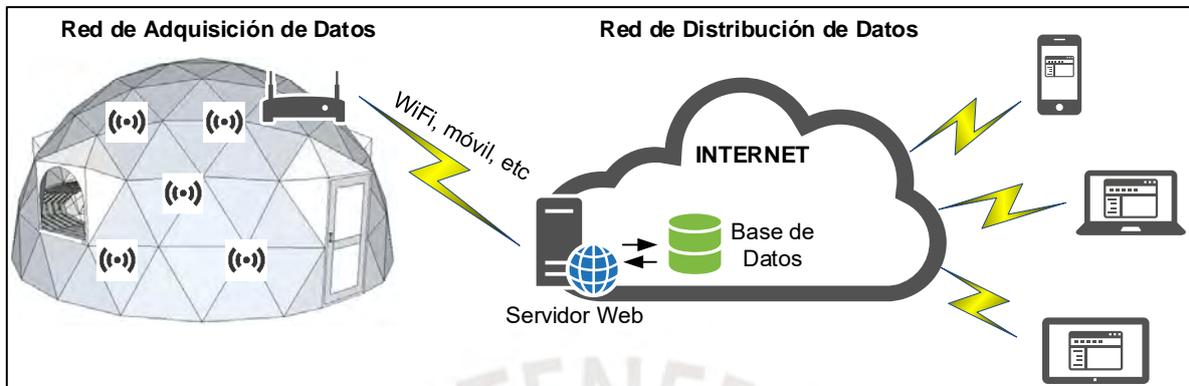
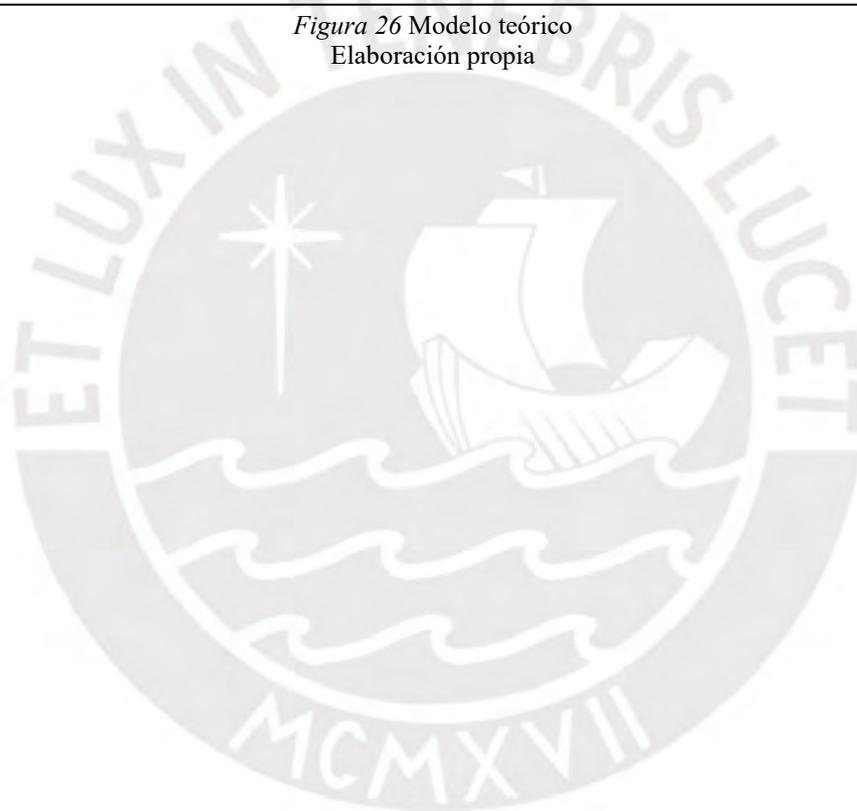


Figura 26 Modelo teórico
Elaboración propia



Capítulo 3. Diseño de un Sistema de Monitoreo para un Invernadero Experimental Basado en una Red de Sensores

3.1 Objetivos

3.1.1 Objetivo general

Diseñar un sistema de monitoreo de variables críticas para un invernadero experimental basado en una red de sensores.

3.1.2 Objetivos específicos

- i. Seleccionar la arquitectura de software adecuada para el sistema de monitoreo.
- ii. Seleccionar el lenguaje de programación, servidor y servicios web para el sistema de monitoreo.
- iii. Seleccionar, diseñar e implementar el esquema de Base de Datos.
- iv. Diseñar e implementar las funciones y la lógica principal del sistema de monitoreo.
- v. Diseñar e implementar una interfaz web gráfica para el monitoreo de las variables ambientales dentro del invernadero.
- vi. Diseñar el sistema de comunicación entre la red de sensores y el servidor.
- vii. Elaborar el presupuesto del proyecto.

3.2 Etapas del Proyecto

Las etapas del presente proyecto fueron estructuradas de acuerdo al ciclo de vida del software. Se elaboró la Figura 27 tomando como referencia el modelo en cascada para el desarrollo del software (Sommerville, 2011).

En primer lugar, se realizó un análisis de requerimientos para nuestro sistema de monitoreo clasificándolos como funcionales y no funcionales. En seguida, en la etapa de diseño, se escogió la arquitectura de software que se adecue mejor a los requerimientos anteriores. De la

misma manera se definieron los servicios web, servidor web, el sistema de administración de BD (DBMS), los flujos de información y se modeló el esquema de BD. Luego, en la etapa de desarrollo, se plasmó todo lo anterior en el lenguaje de programación elegido, donde se consiguió como resultado las funciones y la interfaz gráfica de nuestro sistema de monitoreo. Posteriormente, en la etapa de validación, se realizaron las pruebas de las funcionalidades desarrolladas en nuestro sistema donde se tuvo en cuenta los posibles escenarios de error en la interacción, ya sea con el usuario, o con la red de sensores. Finalmente, se implementó el sistema de monitoreo en un escenario local, el cual contempló la red de adquisición de datos (red de sensores) que se integró con la red de distribución de datos (servidor web y Base de Datos) para verificar la correcta recepción de las mediciones, su almacenamiento, procesamiento y visualización por parte de los usuarios. El desarrollo más detallado de estas etapas se encuentra en los siguientes apartados.

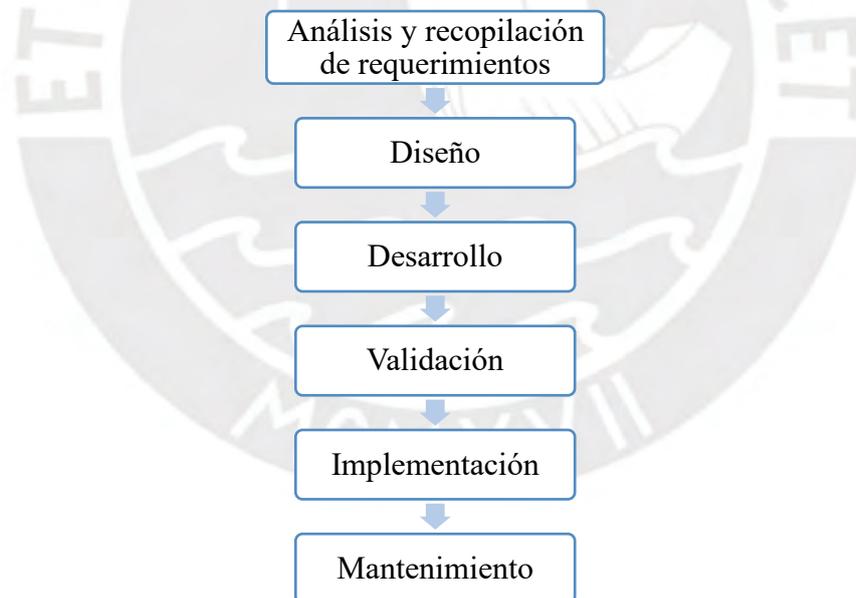


Figura 27 Etapas del Proyecto
Elaboración propia basado en Sommerville, 2011

3.3 Análisis y Recopilación de Requerimientos

Esta primera etapa del proyecto es considerada de vital importancia pues de ella dependerá el éxito de las siguientes fases. En este sentido, se hizo una lista de requerimientos funcionales

y no funcionales. Los requerimientos funcionales especifican lo que el sistema debe hacer, los servicios que debe proveer y su comportamiento ante situaciones específicas. Mientras que los requerimientos no funcionales son especificaciones del sistema como un todo; por ejemplo, usabilidad, rendimiento, disponibilidad, seguridad, etc. (Sommerville, 2011). Se elaboró la Tabla 4 donde se presenta los requerimientos funcionales y no funcionales del proyecto.

Tabla 4 *Listado de Requerimientos Funcionales y No Funcionales*

Requerimientos Funcionales	Requerimientos No Funcionales
<ul style="list-style-type: none"> • Autenticar a los usuarios para el acceso a la plataforma con una única credencial que tenga todos los privilegios. • El invernadero debe ser identificable y georreferenciado. Albergará uno o más sensores dentro de su estructura. • El sensor debe ser identificable, georreferenciado y único dentro del invernadero. También será de un tipo determinado, por ejemplo, sensor de temperatura, humedad, radiación, etc. • Almacenar el registro histórico de todas las mediciones realizadas por los sensores en una Base de Datos exclusiva para la plataforma. • Crear, editar y borrar cada entidad: invernadero, sensor y tipo sensor. • Calcular el promedio, valor máximo/mínimo y filtrar las mediciones según los operadores mayor igual o menor igual. • Mostrar gráficos de las mediciones históricas de los sensores que permitan el monitoreo del invernadero. • Controlar remotamente el actuador del invernadero bajo dos modos: automático y manual. El modo automático dependerá de un valor umbral de temperatura, mientras que el modo manual solo tendrá dos estados (<i>open/close</i>). 	<ul style="list-style-type: none"> • Interfaz clara y de fácil uso para que la interacción con el usuario sea intuitiva y evite errores. • Escalabilidad: para poder monitorear un conjunto creciente de invernaderos con sus sensores correspondientes. • Alto rendimiento: para procesar de manera rápida y simultánea la data enviada por la red de sensores de cada invernadero y a la vez atender a los usuarios de la aplicación web. • Seguridad: identificar a cada sensor e invernadero previo al procesamiento de los datos recibidos. El acceso a la Base de Datos debe ser garantizada por el proceso de autenticación tanto para lectura como escritura de datos. Debe garantizar la protección contra ataques cibernéticos. • Uso de protocolos estándar para facilitar el intercambio de datos entre el sistema, la red de sensores y los usuarios finales. • La disponibilidad del servicio de la aplicación web y la Base de datos debe estar garantizada al 99.9%. • El intervalo de tiempo mínimo para el envío de datos desde la red de sensores hacia el servidor debe ser 5 minutos.

3.4 Diseño

3.4.1 Arquitectura de Software y Servicios Web

Los siguientes puntos fueron considerados para la elección de la arquitectura de software y los servicios web de la aplicación:

- La aplicación web debe concentrar toda la administración de la data recolectada por la red de sensores.
- La aplicación web será la encargada de realizar el procesamiento de la data recolectada para convertirla en información de interés para el usuario.
- La información elaborada debe ser visible a través de una interfaz gráfica para que pueda ser accedida por los usuarios autorizados.

Al tomar en cuenta los enunciados anteriores y lo investigado en el marco teórico, se tomó como referencia la arquitectura Modelo Vista Controlador (MVC) basada en el esquema Cliente-Servidor. La elección de los servicios web se basó en los dos primeros componentes de su stack de protocolos. De esta manera, en la capa de transporte se utilizó el protocolo HTTP y en la capa de mensajería, el componente REST con formato JSON. Adicionalmente, para cubrir la necesidad del monitoreo en tiempo real se implementó AJAX, el cual es una técnica de desarrollo web que permite la ejecución de código en el cliente, mientras se mantiene la comunicación asíncrona con el servidor en segundo plano para mejorar la interactividad y velocidad de la aplicación web. Se elaboró la Figura 28 donde podemos observar lo explicado en esta sección.

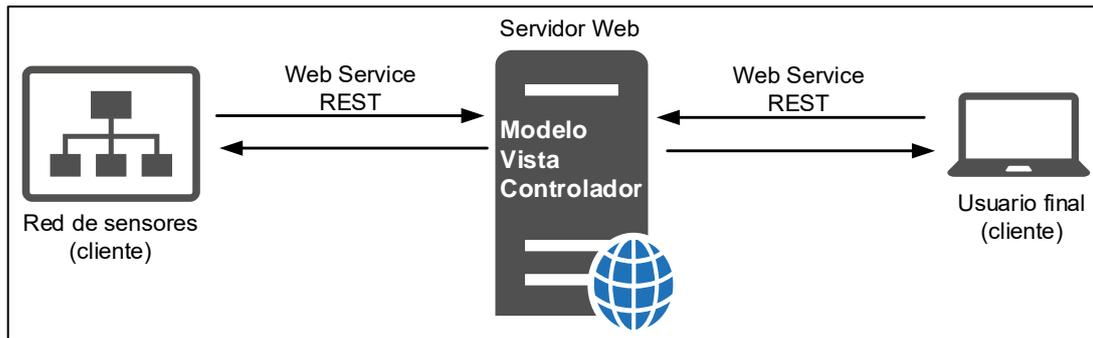


Figura 28 Arquitectura de software y servicios web seleccionados
Elaboración propia

3.4.2 Lenguaje de Programación, Servidor Web y DBMS

Para la elección del lenguaje de programación adecuado para la aplicación; así como también para el servidor web y el DBMS (*DataBase Management System*) se tuvo en cuenta lo siguiente:

- El lenguaje de programación debe adecuarse a la arquitectura elegida (Modelo Vista Controlador) y brindar la capacidad de conexión a la BD.
- Programación orientada a objetos.
- El servidor web debe ser modular y configurable.
- El servidor web debe ser compatible con el lenguaje de programación elegido.
- El DBMS debe soportar el uso de una base de datos relacional.
- La gestión de los datos y el control de acceso tiene que ser sencillo, pues la aplicación no requiere complejidad en el almacenamiento de la data recibida desde la red de sensores.

En ese sentido, se escogió el lenguaje de programación Java por su facilidad en la integración con la arquitectura MVC a través del uso de Beans, Daos y Servlets. Adicionalmente, Java tiene disponible el API JDBC (*Java Database Conectivity*), el cual permite la conexión hacia la BD y ejecutar operaciones sobre la misma. El servidor web elegido es Apache Tomcat, ya que, además de soportar Java desde su concepción, es

gratuito y de código abierto. Respecto al DBMS, se eligió MySQL por las siguientes razones (Oracle Corporation, 2020):

- *Open source* (código abierto).
- Permite el manejo de base de datos relacionales por lo que es considerado un RDBMS (*Relational DataBase Management System*).
- Hace uso del lenguaje SQL (*Structured Query Language*) que permite una gran variedad de operaciones sobre la BD para explotar la potencia de los modelos relacionales.
- Multi-usuario: acceso simultáneo de varios usuarios y/o aplicaciones.
- Multi-hilo: manejo de conexiones independientes para cada usuario y hace uso de los sistemas multiprocesador.
- Seguridad: mediante privilegios se restringe las operaciones y el acceso a ciertas tablas.
- Portabilidad: SQL es un lenguaje estandarizado, de modo que las consultas hechas usando SQL son fácilmente portables a otros sistemas y plataformas.
- Amplia documentación y soporte de la comunidad MySQL.

MySQL trabaja bajo una arquitectura Cliente-Servidor, por lo tanto, dicha arquitectura fue adoptada para la BD de la aplicación. Se elaboró la Figura 29 donde se observa los componentes del DBMS y la interacción con los clientes.

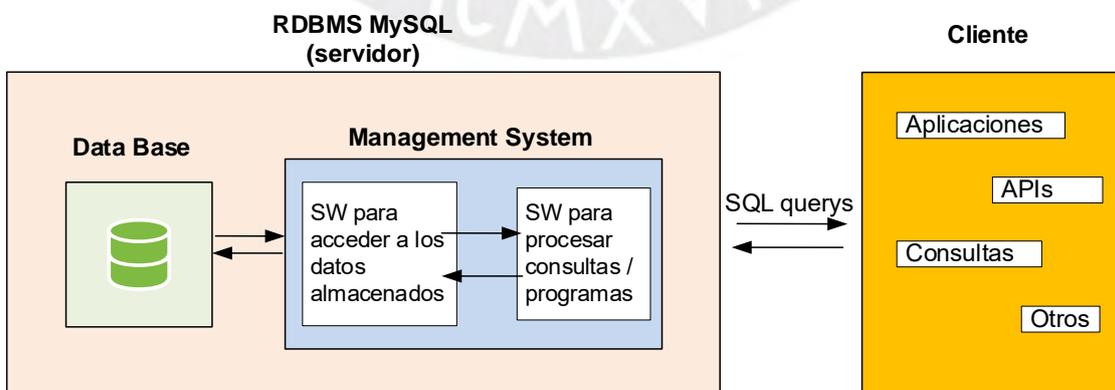


Figura 29 Arquitectura DBMS MySQL.

Elaboración propia basada en Software para Telecomunicaciones II, 2013

3.4.3 Modelado de la Base de Datos

De acuerdo a los enunciados expuestos anteriormente, el modelo de BD elegido es de tipo relacional. Este esquema nos permite establecer relaciones entre entidades (Entidad-Relación), las cuales tienen sus propios atributos. Como parte del diseño de la BD de la aplicación se elaboró la Tabla 5 con las entidades identificadas en los requerimientos funcionales: invernadero, sensor, tipo sensor, mediciones, control y usuarios; cada una con sus respectivos atributos.

Tabla 5 Entidades y atributos de la BD

Entidad	Atributos
Invernadero	<ul style="list-style-type: none"> • ID global invernadero • Nombre • Ubicación • Descripción • Latitud • Longitud • Activo • Fecha de creación • Fecha de última modificación
Sensor	<ul style="list-style-type: none"> • ID global sensor • ID local sensor* • Latitud • Longitud • Activo • Fecha de creación • Fecha de última modificación
Tipo Sensor	<ul style="list-style-type: none"> • ID global del tipo sensor • Nombre • Unidad • Activo • Fecha de creación • Fecha de última modificación
Mediciones	<ul style="list-style-type: none"> • ID global de la medición • Valor • Fecha de la medición • Activo • Fecha de creación • Fecha de última modificación

Control	<ul style="list-style-type: none"> • ID global de control • Modo • Parámetro • Activo • Fecha de creación • Fecha de última modificación
Usuarios	<ul style="list-style-type: none"> • ID global de usuario • Nombre • Email • Password • Activo • Fecha de creación • Fecha de última modificación

Elaboración propia

*Se consideró un ID adicional para la entidad “Sensor”, pues servirá para identificar la data recibida de la red de sensores para un sensor en particular. Este ID tiene un alcance local; es decir, es único dentro de un invernadero.

Las relaciones entre las entidades pueden ser de tres tipos: uno a uno (1:1), uno a muchos (1:N) o viceversa (N:1) y muchos a muchos (N:N). Cada uno de estos tres tipos puede ser una relación fuerte, si una de las dos entidades no tiene significado en sí dentro del modelo si no está relacionada con la otra; o débil, si ambas entidades tienen significado de por sí en el modelo (Software para Telecomunicaciones II PUCP, 2013). En consecuencia, las relaciones establecidas entre las entidades fueron determinadas de acuerdo a los siguientes enunciados de diseño, los cuales están basados en los requerimientos funcionales que fueron expuestos anteriormente:

- i. Un invernadero debe albergar uno o más sensores dentro de su estructura.
- ii. Un sensor debe ser de un tipo determinado sin importar que existan más sensores del mismo tipo.
- iii. Un sensor debe tener registrado el histórico de todas sus mediciones.

- iv. El invernadero debe tener la capacidad de variar su modo de control (automático o manual) de acuerdo a la necesidad.

Se elaboró la Tabla 6 donde se observa las traducciones de estos enunciados en relaciones entre las entidades.

Tabla 6 *Relaciones entre las entidades de la BD*

Enunciado	Entidad A	Entidad B	Relación
i.	Invernadero	Sensor	Débil Uno (A) a muchos (B) [1:N]
ii.	Sensor	Tipo Sensor	Débil Muchos (A) a uno (B) [N:1]
ii.	Sensor	Mediciones	Débil Uno (A) a muchos (B) [1:N]
iv.	Invernadero	Control	Débil Uno (A) a uno (B) [1:1]

Elaboración propia

Una vez definido, tanto las entidades como las relaciones, se plasmaron en la herramienta MySQL Workbench. Es aquí donde las entidades se convierten en tablas, los atributos de las entidades en atributos de las tablas y las relaciones entre las entidades serán las relaciones entre las tablas. Esta herramienta nos brinda la posibilidad de conectarnos a la BD MySQL como si fuera un cliente más de dicho servidor. Además, Workbench posee una interfaz gráfica que nos facilita tres cosas: diseño de la BD (modelado), desarrollo (editor para la programación en lenguaje SQL) y administración (estado de la BD, accesos de usuarios, etc.). En el Anexo "A" se observa el modelo relacional de la BD en un diagrama de MySQL Workbench.

3.4.4 Mock Up y Diagramas de Flujo

Se elaboró un *mock-up* para tener plasmada una primera idea de la interfaz gráfica de la aplicación web. En el Anexo “B” se observa el *mock-up* diseñado para las vistas principales de la aplicación web con la ayuda de la herramienta Balsamiq. Por otro lado, en esta sección se ha diseñado los diagramas de flujo para cada interacción posible, ya sea entre un usuario y la aplicación web, o la red de sensores y la aplicación web. El detalle de cada diagrama de flujo se encuentra en el Anexo “C”.

3.5 Desarrollo

Java para aplicaciones web (Java Web) contiene los siguientes elementos relacionados al patrón MVC: Bean y Dao representan al Modelo; las páginas de contenido como HTML y JSP (Java Server Pages) representan la Vista; y los Servlets, al Controlador.

En base a las premisas de diseño de la sección anterior se desarrollaron los Beans, Daos, Servlets y JSP que se observan en la Tabla 7. El detalle del desarrollo; es decir, el código Java, se encuentra adjunto en el Anexo “F”. En resumen, estos componentes junto con la Base de Datos, forman la estructura de la red de distribución de datos que fue definida en el modelo teórico.

Tabla 7 Beans, Daos, Servlets y JSP implementados

	Nombre	Descripción
Bean	- Invernadero - Sensor - TipoSensor - Medición - Control - User	Beans usados para reflejar las entidades y atributos de la BD.
	- InfoStream - Sensado - Reply	Beans usados para la interacción entre la red de sensores y la aplicación web.
Dao	- BaseDao	Dao usado para establecer la conexión con la BD.

	<ul style="list-style-type: none"> - InvernaderoDao - SensorDao - TipoSensorDao - ControlDao - UserDao 	Daos usados para realizar la lectura/escritura de la data almacenada en la BD.
Servlet	<ul style="list-style-type: none"> - LoginServlet - InvernaderoServlet - SensorServlet 	Servlets usados para manejar la lógica y funcionalidades de la aplicación.
	<ul style="list-style-type: none"> - InfoServlet 	Servlet usado para manejar la lógica de interacción entre la red de sensores y la aplicación web.
JSP	<ul style="list-style-type: none"> - Index - Admin - NuevoInvernadero - NuevoSensor - ConfigurarTipoSensor - DetailInvernadero - EditarSensor 	Páginas usadas como interfaz gráfica para la interacción entre el usuario y la aplicación web.

Elaboración propia

3.5.1 Interfaz Gráfica

En esta sección se describen las vistas principales de nuestra aplicación web. En la Figura 30 se muestra el panel principal donde se observa todos los invernaderos registrados. En la Figura 31 se detalla las mediciones de los sensores a través de la opción de filtrado. Estas y otras vistas de la interfaz gráfica se encuentran en el Anexo “D”.

Figura 30 Panel principal de la aplicación web

Elaboración propia

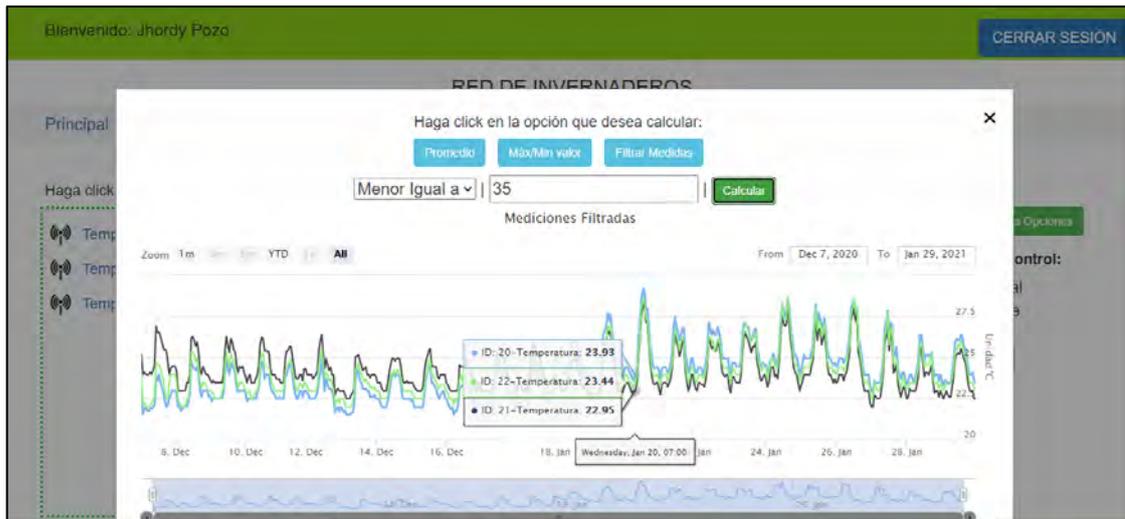


Figura 31 Mediciones de los sensores
Elaboración propia

3.5.1.1 Inicio de sesión

En esta vista el usuario debe ingresar sus credenciales, correo electrónico y contraseña, para autenticarse contra la Base de Datos y así poder ingresar al panel principal (Figura D1).

3.5.1.2 Panel principal

Esta vista contiene la lista de los invernaderos registrados en la aplicación con sus detalles respectivos y su ubicación geográfica en el mapa. Cuenta con las opciones para editar y borrar un invernadero determinado. Además, muestra la opción para añadir un nuevo invernadero y, también otra, para configurar los tipos de sensores (Figura D2).

3.5.1.3 Añadir nuevo invernadero

Esta vista es un formulario donde el usuario tendrá que llenar los campos solicitados para la creación de un invernadero. También es necesario configurar el modo de control y el parámetro respectivo; así como también, los sensores correspondientes (Figura D3).

3.5.1.4 Configurar Tipo de Sensor

Esta vista muestra el listado de los tipos de sensores configurados. Cuenta con las opciones para editar y borrar un tipo de sensor determinado. En la parte inferior se visualiza un formulario para crear un nuevo tipo de sensor (Figura D4).

3.5.1.5 Detalle de un invernadero

Esta vista contiene lo siguiente (Figura D5):

- Información del invernadero (ID y nombre).
- Lista de sensores registrados y su ID local correspondiente.
- Estado actual del control y la opción de editar el modo y parámetro.
- Opción de agregar nuevos sensores.
- Más opciones: promedio, valor máximo/mínimo y filtrar mediciones.

3.5.1.6 Añadir nuevo sensor

Esta vista muestra la lista de sensores existentes con la opción de borrar el que se elija. En la parte inferior se visualiza un formulario donde el usuario puede agregar la cantidad de sensores que desee al invernadero correspondiente (Figura D6).

3.5.1.7 Detalle de un sensor

Esta vista muestra el gráfico que representa los valores históricos medidos por el sensor. También contiene la información del sensor como el ID local y tipo de sensor. En la parte inferior se tienen las opciones de editar y borrar el sensor (Figura D7).

3.5.1.8 Editar un sensor

Esta vista muestra el formulario precargado con la información actual del sensor (ID local y tipo de sensor), los cuales se pueden editar (Figura D8).

3.5.1.9 Más opciones

Esta vista muestra tres opciones de cálculo basado en las mediciones de los sensores, tales como promedio, valor máximo/mínimo y filtrar las mediciones a partir de un valor determinado (Figura D9, D10, D11).



Capítulo 4. Pruebas, Resultados y Análisis de Costos

4.1 Validación de la Aplicación Web

Se realizó el *smoke test* de la aplicación web que consiste en pruebas de todas las funcionalidades diseñadas en el capítulo anterior. Este proceso nos permite encontrar y corregir posibles puntos de falla durante la interacción del usuario con la aplicación web. Se elaboró la Tabla 8 donde se observa las pruebas realizadas a la aplicación junto con el resultado obtenido.

Tabla 8 *Pruebas de validación de la aplicación web*

Entidad	Funcionalidad	Descripción de la prueba	Resultado esperado	Valoración
Usuario	Inicio de sesión	Ingresar un correo electrónico en formato distinto a: example@example.com .	La aplicación solicita ingresar un correo electrónico válido.	OK
		Ingresar un correo electrónico y contraseña no registrados en la aplicación.	La aplicación solicita ingresar credenciales válidas.	OK
		Ingresar solamente el correo sin contraseña o viceversa.	La aplicación solicita llenar ambos campos.	OK
Tipo Sensor	Registrar nuevo tipo de sensor	Enviar el formulario con los campos vacíos.	La aplicación solicita completar los campos vacíos.	OK
	Editar atributos de un tipo de sensor	Enviar el formulario con los campos vacíos.	La aplicación solicita completar los campos vacíos.	OK
		Modificar el ID del tipo de sensor a un formato incorrecto o que no sea activo, antes de enviar el formulario.	La aplicación valida el ID y no ejecuta la actualización.	OK
	Borrar un tipo de sensor	Eliminar un tipo de sensor mientras existen sensores de dicho tipo.	La aplicación solicita borrar primero todos los sensores asociados a dicho tipo de sensor.	OK

		Modificar el ID del tipo de sensor a un formato incorrecto o que no sea activo, antes de eliminarlo.	La aplicación valida el ID y no ejecuta el borrado.	OK	
Invernadero	Registrar nuevo invernadero	Enviar el formulario con los campos vacíos.	La aplicación solicita completar los campos vacíos.	OK	
		Ingresar un modo distinto a manual/automático o el parámetro de control automático en formato diferente al numérico o el parámetro manual diferente a open/close.	La aplicación solicita ingresar el modo y parámetro de control correcto.	OK	
		Ingresar los sensores con ID locales en formato diferente al numérico o repetidos dentro del invernadero.	La aplicación solicita ingresar los ID locales en formato numérico y que sean únicos dentro del invernadero.	OK	
	Editar atributos de un invernadero	Enviar el formulario con los campos vacíos.	La aplicación solicita completar los campos vacíos.	OK	
		Modificar el ID del invernadero a un formato incorrecto o que no sea activo, antes de enviar el formulario.	La aplicación valida el ID y no ejecuta la actualización.	OK	
	Visualizar los detalles de un invernadero	Modificar el ID del invernadero a un formato incorrecto o que no sea activo, antes de ingresar a ver sus detalles.	La aplicación valida el ID y no muestra la página con detalles del invernadero.	OK	
	Visualizar las operaciones basadas en las mediciones de los sensores	Modificar el ID del invernadero a un formato incorrecto o que no sea activo, antes de seleccionar la operación de interés.	La aplicación valida el ID y no muestra la vista con el resultado de la operación seleccionada.	OK	
	Borrar invernadero	Modificar el ID del invernadero a un formato incorrecto o que no sea activo, antes de eliminarlo.	La aplicación valida el ID y no ejecuta el borrado.	OK	
	Sensor	Registrar nuevo sensor	Ingresar los sensores con ID locales en formato diferente al numérico o	La aplicación solicita ingresar los ID locales en	OK

		repetidos dentro del invernadero.	formato numérico y que sean únicos dentro del invernadero.	
	Editar atributos de un sensor	Ingresar un ID local en formato incorrecto o que ya está asignado a otro sensor dentro del mismo invernadero.	La aplicación solicita ingresar el ID local en formato numérico y que sea único dentro del invernadero.	OK
		Modificar el ID del sensor a un formato incorrecto o que no sea activo, antes de enviar el formulario	La aplicación valida el ID y no ejecuta la actualización.	OK
	Visualizar las mediciones de un sensor	Modificar el ID del sensor a un formato incorrecto o que no sea activo, antes de ingresar a ver sus mediciones.	La aplicación valida el ID y no muestra la vista de las mediciones.	OK
	Borrar sensor	Modificar el ID del sensor a un formato incorrecto o que no sea activo, antes de eliminarlo.	La aplicación valida el ID y no ejecuta el borrado.	OK
Control	Configurar el tipo de control	Ingresar un modo distinto a manual/automático o el parámetro de control automático en formato diferente al numérico o el parámetro manual diferente a open/close, antes de enviar el formulario.	La aplicación solicita ingresar el modo y parámetro de control correcto.	OK

Elaboración propia

4.2 Integración de la Aplicación Web con la Red de Sensores

4.2.1 Escenario Local

Para la integración entre la aplicación web y la red de sensores se implementó un escenario de prueba que consiste en la conexión de ambos componentes en una misma red local (LAN). Este escenario práctico permite probar el envío, recepción,

almacenamiento y procesamiento de los datos, así como también la visualización de la información en un cliente web. Se elaboró la Figura 32 donde se observa el esquema de integración propuesto.

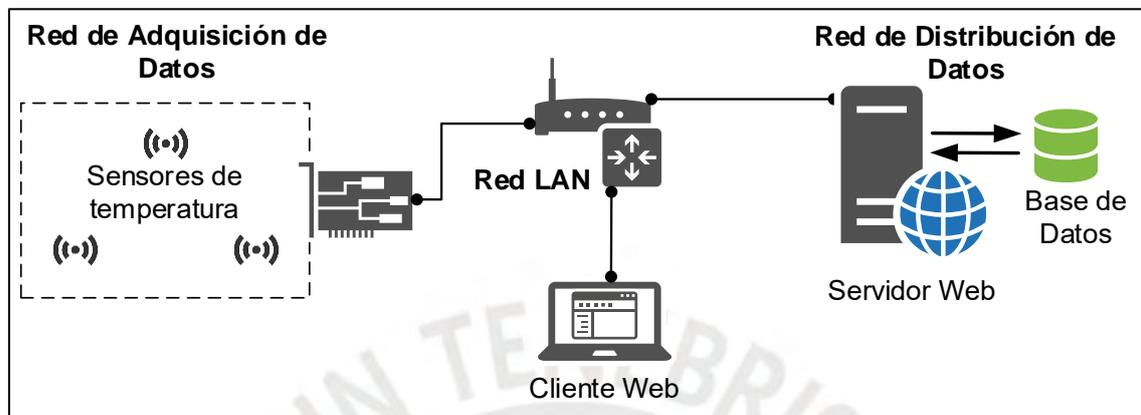


Figura 32 Esquema de integración
Elaboración propia

4.2.2 Hardware

A continuación, se detalla la lista del hardware utilizado para las pruebas.

- (01) Arduino UNO REV3: tarjeta basada en el microcontrolador ATmega328P. Principalmente cuenta con seis (06) entradas analógicas, catorce (14) pines digitales input/output y posee una entrada de alimentación donde se puede conectar un adaptador AC/DC de 9V (Arduino, 2020).
- (01) Arduino Shield Ethernet: cubierta para el Arduino UNO que permite la conexión a una red IP a través de su puerto Ethernet con un conector RJ45.
- (03) Sensores de temperatura LM35: sensor de precisión con un voltaje de salida linealmente proporcional a la temperatura en grados centígrados ($+10\text{mV}/^\circ\text{C}$). Su rango de medición va desde los 0°C hasta los 100°C . Posee una precisión de $\pm 1.5^\circ\text{C}$ que depende de las condiciones de temperatura, voltaje y corriente (Texas Instruments, 2017).
- (01) Diodo LED
- (01) Protoboard

- (01) Cable Modem Router Askey
- (01) Computadora portátil Core i5, 12GB RAM.
- Conectores y cables

4.2.3 Sistema de Comunicación entre la Red de Sensores y el Servidor Web

De acuerdo a lo expuesto en el Capítulo 3, para el envío y recepción de los datos entre la red de sensores y el servidor web se usó el protocolo HTTP con el método POST y los datos en formato JSON. En ese sentido, la estructura del método POST, enviado por la red de sensores, debe ser la siguiente:

```
POST /InvernaderoWatcher/InfoServlet HTTP/1.1
Host: (IP del servidor)
Content-Type: application/json
Content-Length: (tamaño de los datos)
{Datos en formato JSON}
```

El formato JSON que debe enviar la red de sensores, hacia el servidor web es el siguiente:

```
{  "timemedicion": YYYYYYYYYYYYYYYY,
   "iddomo": X,
   "sensado": [  {  "id": Y,
                  "valor": W
                },
                ... (más elementos del arreglo)
              ]
}
```

Se elaboró la Tabla 9 donde se especifica el tipo de dato de cada parámetro que forma parte de la estructura del JSON.

Tabla 9 *Tipo de dato del JSON enviado por la red de sensores*

Parámetro	Descripción	Tipo de dato
timemedicion	Fecha de la medición	Long
iddomo	ID global del invernadero	Integer
sensado	Arreglo con la información recolectada por los sensores	Array
id	ID local del sensor	Integer
valor	Valor medido por el sensor	Float

Elaboración propia

Por otro lado, cada vez que el servidor web reciba las mediciones desde la red de sensores, responderá con otro JSON, el cual contiene el resultado de la lógica de control procesada. En base a este valor, el actuador del invernadero debe ejecutar la orden, ya sea abrir o cerrar. El formato JSON enviado desde el servidor web hacia la red de sensores es el siguiente:

```
{  "ejecuta": "yyyyy"
}
```

Se elaboró la Tabla 10 donde se especifica el tipo de dato del parámetro anterior.

Tabla 10 *Tipo de dato del JSON enviado por el servidor web*

Parámetro	Descripción	Tipo de dato
ejecuta	Puede ser "open" o "close"	String

Elaboración propia

4.2.4 Integración

Para la red de sensores, se construyó el circuito de la Figura 33 con el hardware especificado anteriormente, entre los que destaca los tres sensores de temperatura, el LED y el Arduino con Shield Ethernet. Se definió un intervalo de 15 minutos para el envío de las mediciones, lo cual respeta lo exigido en los requerimientos iniciales. El detalle del código programado para el Arduino se encuentra en el Anexo “E”.

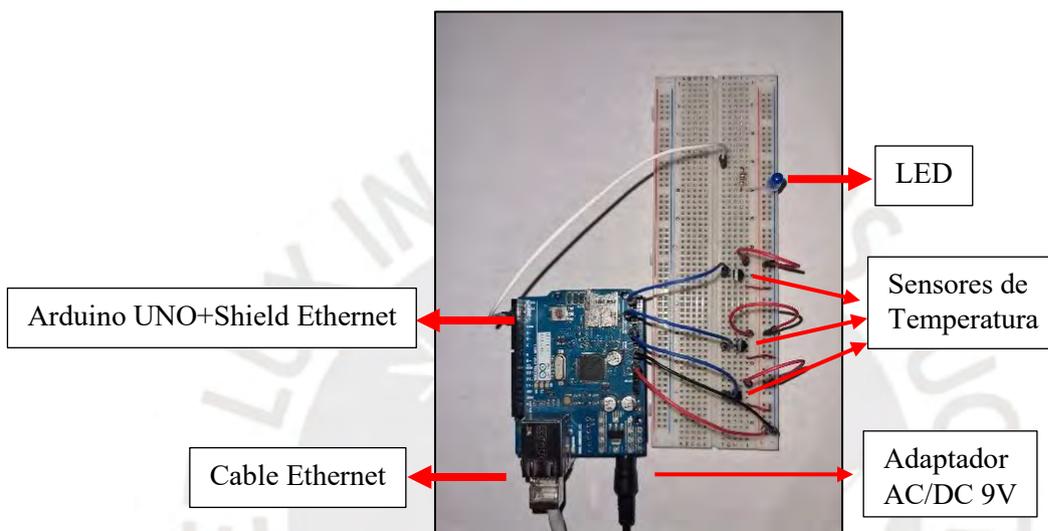


Figura 33 Circuito de la red de sensores
Elaboración propia

Antes de iniciar con el envío y recepción de las mediciones, se utilizó las funcionalidades desarrolladas en el capítulo anterior para crear el tipo de sensor (temperatura), el invernadero (Domo1), sus sensores (3), la ubicación geográfica (Lima, San Miguel) y el modo de control (manual/close) dentro de la aplicación web. En la Figura 34 y 35 se observa el detalle de lo creado.

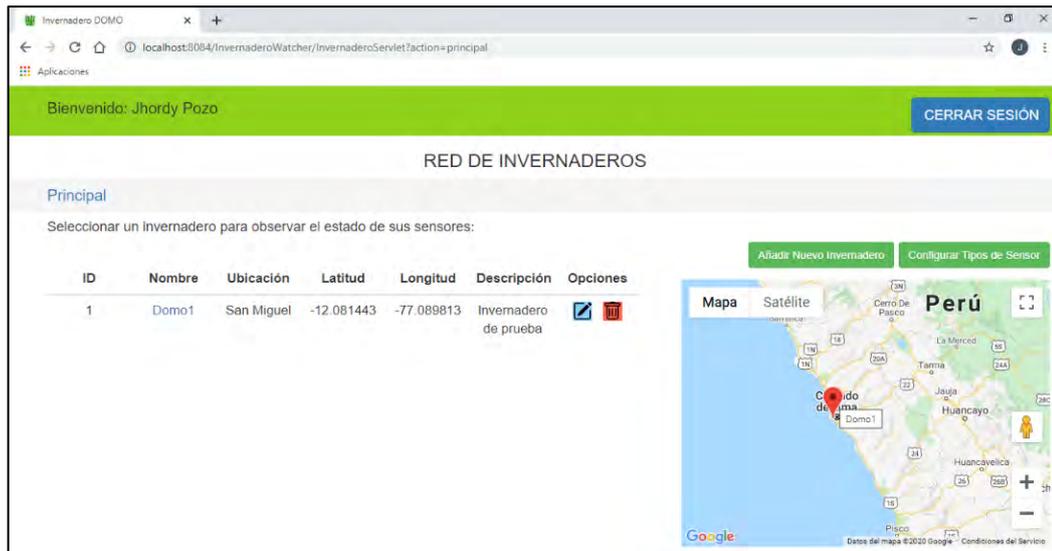


Figura 34 Nuevo invernadero creado
Elaboración propia

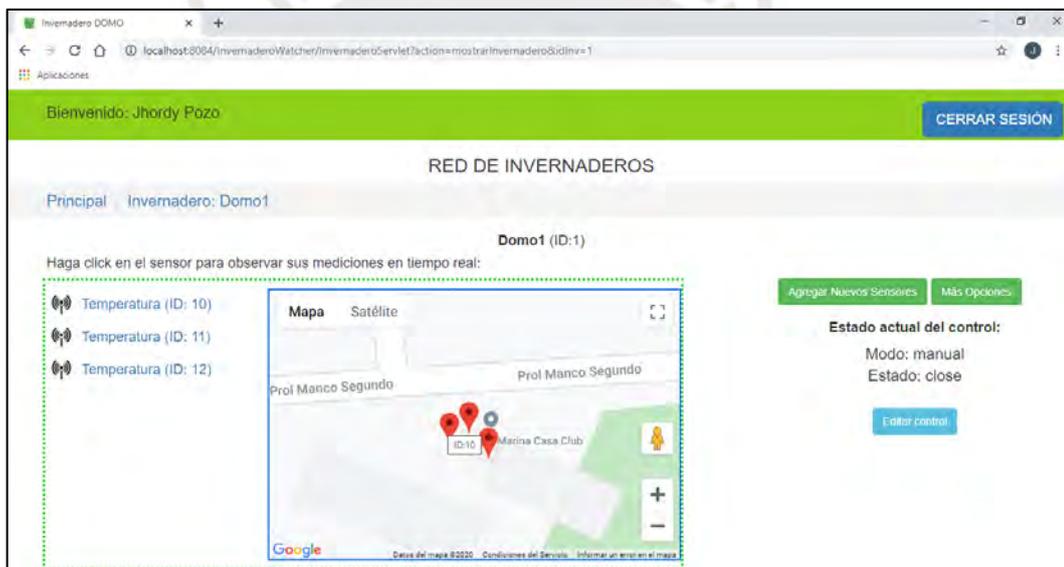


Figura 35 Nuevos sensores y modo de control configurado
Elaboración propia

4.3 Resultados

Los datos enviados desde la red de sensores y la respuesta del servidor web respetan el formato establecido para el sistema de comunicación, lo cual se puede visualizar con la ayuda de la herramienta *Monitor Serie* del Arduino. En la Figura 36 se muestra la evidencia de lo mencionado anteriormente.

```

COM4
Serial ready
Ethernet ready
{
  "timemedicion": 1607377675,
  "iddomo": 1,
  "sensado": [
    {
      "id": 10,
      "valor": 23.95
    },
    {
      "id": 11,
      "valor": 25.90
    },
    {
      "id": 12,
      "valor": 24.44
    }
  ]
}
}Connect to 192.168.1.31
Connected
POST /InvernaderoWatcher/InfoServlet
Content-Length: 122
192.168.1.33
{"ejecuta":"close"}
Disconnect
  
```

JSON enviado desde la red de sensores

JSON recibido del servidor web

Figura 36 Resultados Monitor Serie Arduino
Elaboración propia

Una vez que los datos son recibidos por el servidor pueden ser visualizados en un cliente que también forme parte de la red LAN e inicie sesión en la aplicación web. En la Figura 37 se muestra la vista de las mediciones almacenadas y procesadas de un sensor específico y en la Figura 38 se observa las mediciones de los tres sensores a través de la opción filtrar medidas.

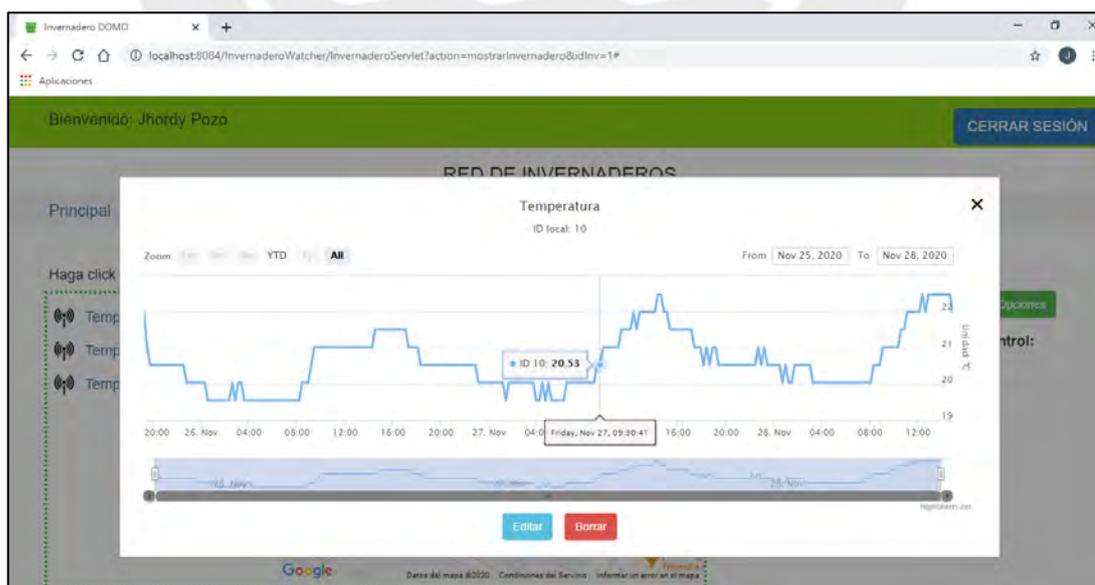


Figura 37 Mediciones históricas de un sensor de temperatura
Elaboración propia

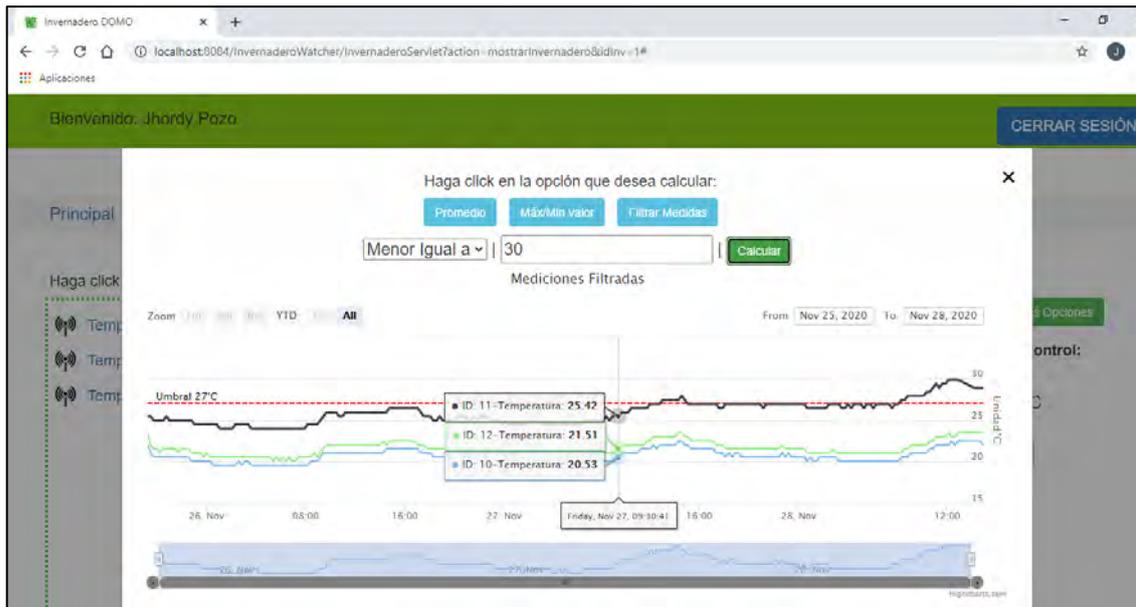


Figura 38 Mediciones históricas filtradas de los tres sensores de temperatura
Elaboración propia

Para verificar la lógica de control se utilizó un diodo LED en la red de sensores, el cual representa el actuador. El LED cambiará a estado ON, cuando la orden desde el servidor web sea abrir (open); u OFF, cuando sea cerrar (close). Debemos recordar que se definió dos modos de control, manual y automático. En el caso del modo manual y el estado “close”, se verifica que el LED permanece en OFF, según la Figura 39. En el caso del modo manual y el estado “open”, se verifica que el LED cambia a ON, según la Figura 40.

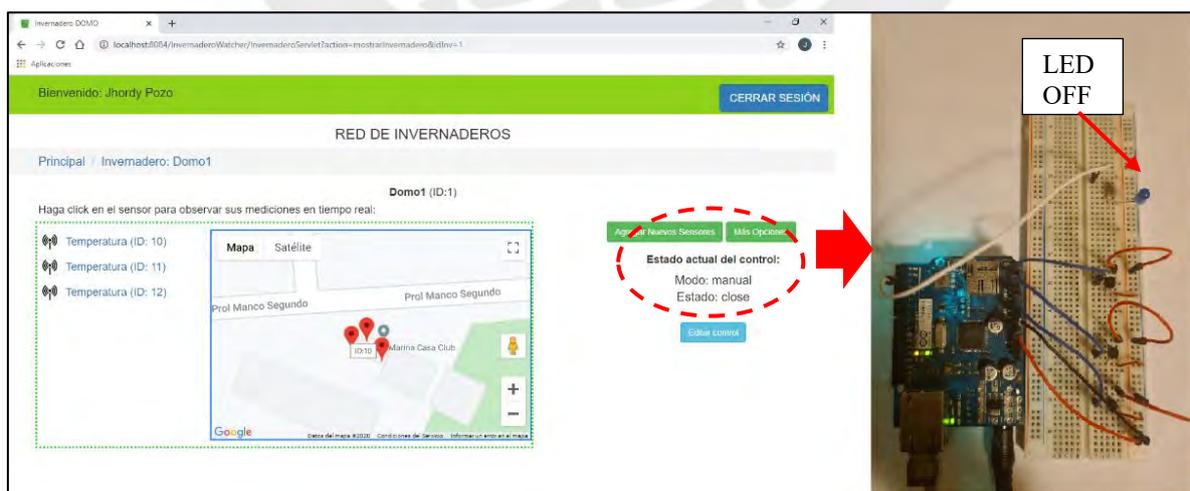


Figura 39 Modo manual, estado “close”, LED OFF
Elaboración propia

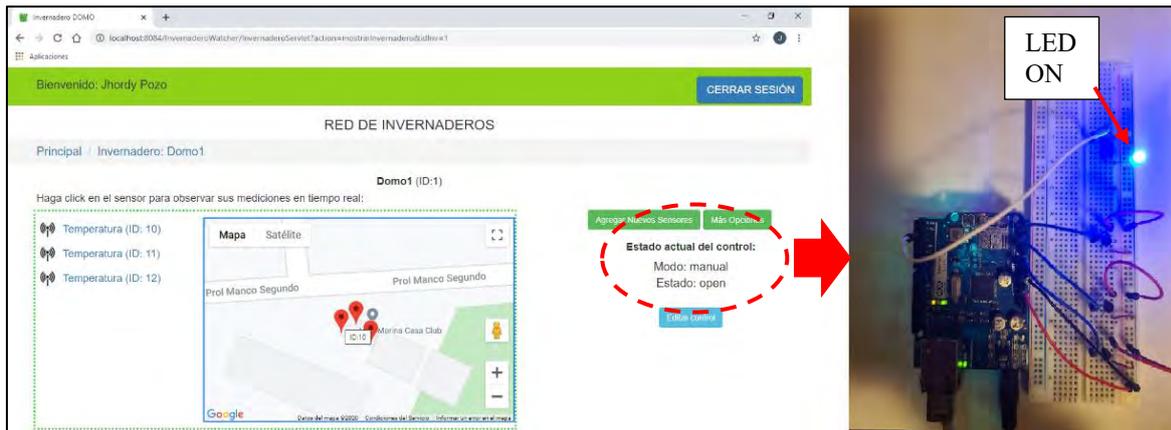


Figura 40 Modo manual, estado “open”, LED ON
Elaboración propia

Cuando el modo de control es automático dependerá del valor umbral configurado (ver Figura 41) para determinar el estado del LED. El servidor realiza el cálculo del promedio de las cuatro últimas mediciones de los sensores de temperatura. Mientras el promedio sea menor que el umbral, el LED permanecerá en OFF, según la Figura 42. Si el promedio es mayor o igual que el umbral, el LED cambiará a ON, según la Figura 43.

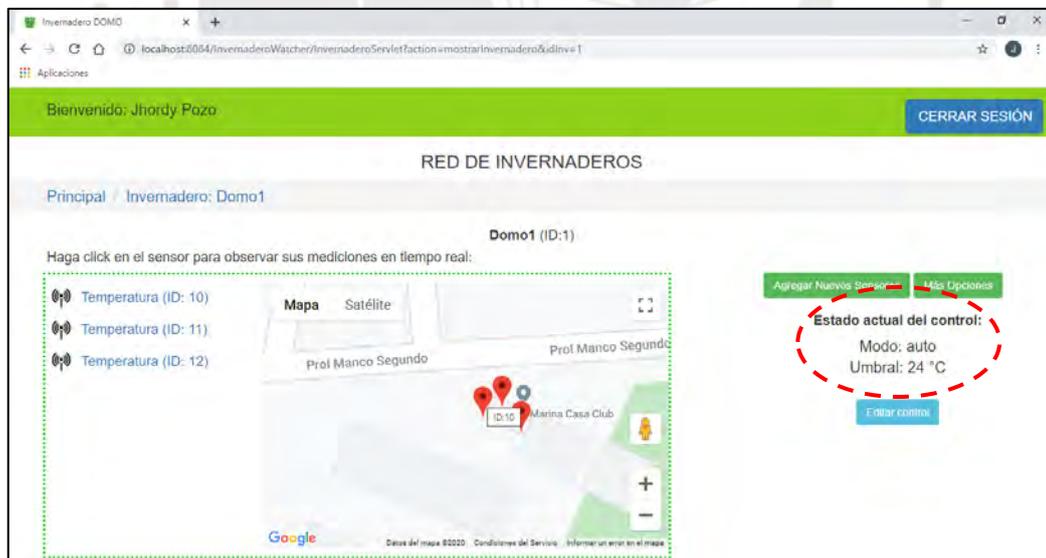


Figura 41 Modo Automático con umbral configurado
Elaboración propia



Figura 42 Modo automático LED OFF
Elaboración propia

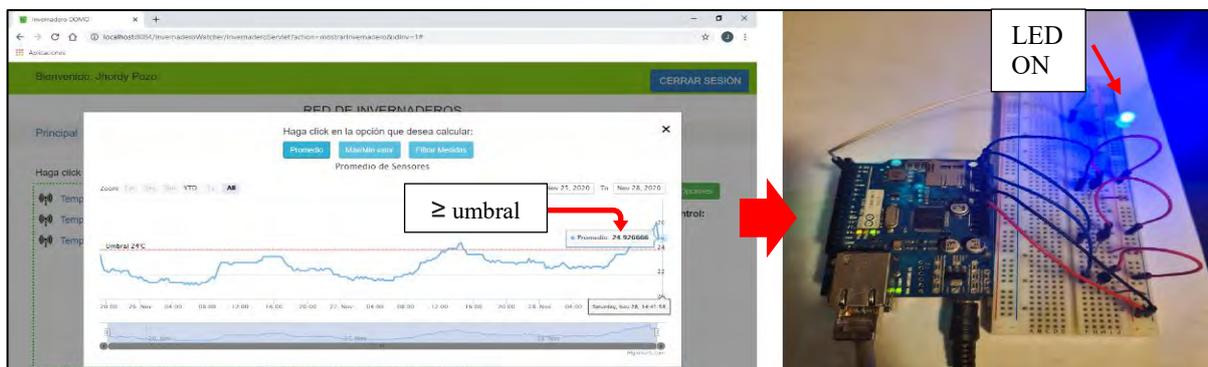


Figura 43 Modo automático LED ON
Elaboración propia

4.3.1 Post procesamiento y análisis

Con las mediciones ya almacenadas en la Base de Datos se procedió a exportar los valores en formato separado por coma (CSV) para procesarlas en la herramienta MATLAB. Se seleccionaron las mediciones de 7 días consecutivos, del 18 al 24 de enero de 2021. Con el objetivo de visualizar la tendencia de la curva, sin considerar las variaciones abruptas debido a la precisión del sensor ($\pm 1.5^{\circ}\text{C}$), se aplicó un filtro de media móvil. En la Figura 44 se observa el resultado del procesamiento.

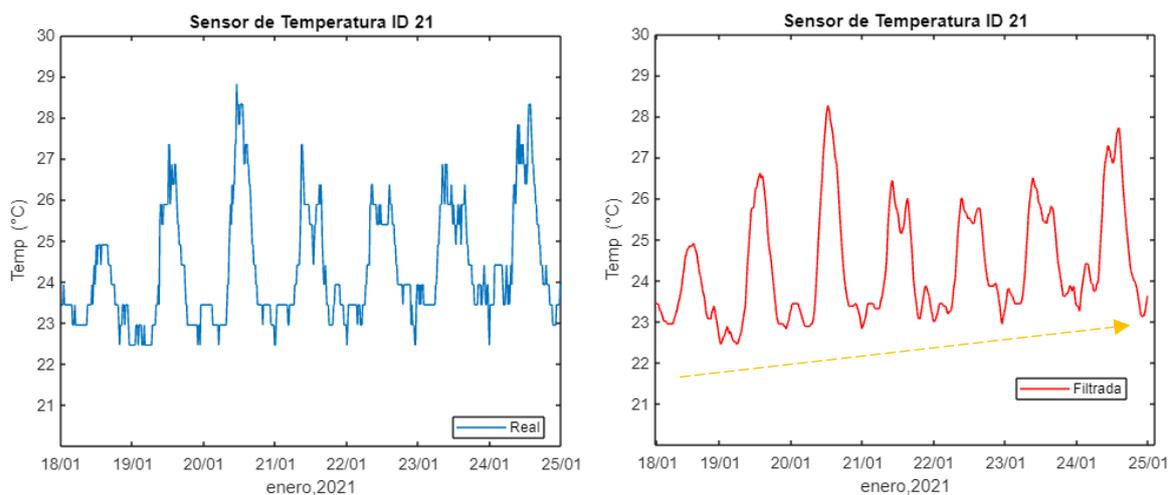


Figura 44 Filtrado de mediciones en MATLAB
Elaboración propia

En el intervalo de tiempo analizado observamos una tendencia de incremento de temperatura debido a la temporada de verano en Lima. La referencia externa más cercana, que corrobora esta tendencia, es el reporte de temperatura de la estación meteorológica automática del Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI, 2021), ubicada en el Campo de Marte, distrito de Jesús María. En la Figura 45 se muestra el reporte en mención.

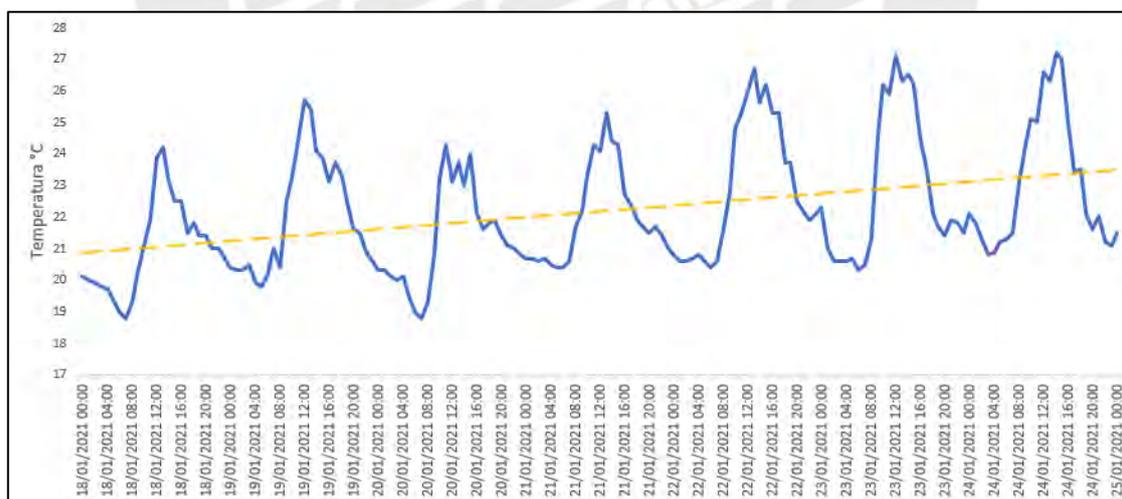


Figura 45 Reporte de temperatura de la estación Campo de Marte - SENAMHI
Elaboración propia

Una vez verificada la tendencia, con el apoyo de la referencia externa, se concluye que las mediciones de nuestro sistema son consistentes y válidas para relacionarlas con otros factores que ayuden a acelerar la productividad, por ejemplo, la cantidad de abono, agua, calor, luz, etc.

4.4 Análisis de Costos

Inversión Capital

Es la inversión inicial para empezar con el proyecto. Se elaboró la Tabla 11 donde se detalla los precios estimados para cada componente de acuerdo a dos grupos:

- i. Red de adquisición de datos:** se consideró un invernadero de tipo capilla de dimensiones 9m (largo) x 4m (ancho) x 3m (alto) ubicado en la parte posterior al pabellón V de la PUCP. Se propone instalar una red de sensores de cuatro tipos: temperatura, humedad, luz y CO₂; distribuidos tal como se muestra en la Figura 46 (Myong-Jin , y otros, 2012). Los sensores serán controlados por el Arduino UNO, el cual se conectará vía WiFi a la red inalámbrica que será implementada especialmente para el invernadero. Para la red inalámbrica se propone instalar un Access Point en modo Bridge de tal forma que se conecte al WiFi PUCP y, a la vez, se pueda crear una nueva red WiFi con un SSID específico que cuente con cifrado WPA2 para el invernadero. De esta forma podremos tener control sobre los dispositivos que se conecten a nuestra nueva red WiFi, ya que aplicaremos el filtrado por MAC address como mecanismo de seguridad. En la Figura 47 se observa el esquema de conexión inalámbrica propuesto. La configuración se detalla en el Anexo “G”. Otra alternativa que también se está considerando, en caso no se use el WiFi PUCP, es la implementación de un Hotspot conectado a la red de datos móviles, donde también se implementará el cifrado WPA2 y filtrado por MAC address. La configuración de esta alternativa se detalla en el Anexo “H”. Por otro lado, también se incluyó el costo del kit actuador de ventana y un sistema de protección eléctrica, según recomendación NFPA 77.
- ii. Red de distribución de datos:** se tomó en cuenta el costo total del software de la aplicación web, el cual contempla todas las etapas del ciclo de vida del software desarrolladas anteriormente, cuyo valor es fijo y no incrementará al tener más invernaderos o sensores registrados en la aplicación.

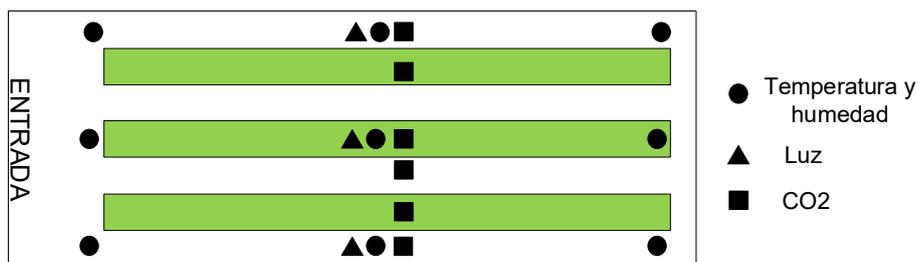


Figura 46 Distribución de sensores dentro del invernadero
Elaboración propia

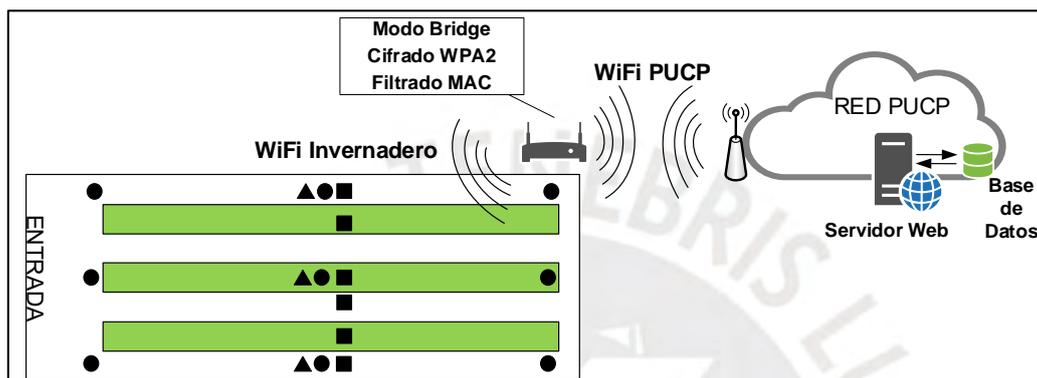


Figura 47 Esquema de conexión inalámbrica para el invernadero
Elaboración propia

Tabla 11 *Inversión capital*

	Descripción	Cantidad	Precio Unitario	Subtotal
Red de Adquisición de Datos	Arduino UNO WiFi Rev2	2	S/.200.00	S/.400.00
	Sensor de temperatura y humedad DHT22	9	S/.25.00	S/.225.00
	Sensor de luz LDR5516	3	S/.10.00	S/.30.00
	Sensor de CO2 MQ135	6	S/.15.00	S/.90.00
	Cableado y otros materiales.	1	S/. 150.00	S/. 150.00
	Kit actuador ventana: motor Nema23+driver+ fuente	2	S/.650.00	S/.1300.00
	Access Point TP LINK TL-WA901ND	1	S/.150.00	S/.150.00
	ZTE Router MF920U	1	S/.180.00	S/.180.00
	Instalación de sistema de puesta a tierra (10Ω: red de sensores y 5Ω: pararrayo)	1	S/.900.00	S/.900.00
	Kit pararrayo tipo Franklin Tetrapuntal 25°	1	S/.3,500.00	S/.3,500.00
	Mano de Obra-Instalación	1	S/.1,000.00	S/.1,000.00
Red de Distribución de Datos	Software de la aplicación web y Base de Datos	1	S/.8,000.00	S/.8,000.00
	Total			S/.15,925.00

Elaboración propia

Operación

Es el costo mensual para asegurar el funcionamiento de la red de sensores y de la aplicación web. Para calcular el costo de operación de la red de adquisición de datos se tuvo en cuenta el consumo de energía, calculado en la Tabla 12; el consumo de datos móviles, calculado en la Tabla 13; y el costo de mantenimiento del sistema de puesta a tierra. Mientras que para la red de distribución de datos se tuvo en cuenta el costo de hosting de la aplicación web y la Base de Datos para asegurar la operación de la plataforma 7x24 con una disponibilidad del servicio al 99.9% tal como se exige en los requerimientos. Cabe mencionar que la seguridad de nuestro sistema está garantizada a través del servicio de protección web contra malware, ataques tipo DDoS e inyección SQL, incluido también dentro de los costos. En caso se implemente la aplicación web y la Base de Datos dentro de la infraestructura la PUCP, entonces este costo no aplicaría. Se elaboró la Tabla 14 donde se detallan todos los costos operativos mensuales.

Tabla 12 *Consumo de energía mensual de la red de sensores*

Equipo	Cantidad	Potencia (W)	Horas al día	Días al mes	Energía (kWh)
Arduino	02	5.5	24	31	8.18
Actuador	02	350	1	31	21.7
Access Point	01	5.8	24	31	4.32
Total					34.2 kWh

Elaboración propia

Tabla 13 *Consumo de datos mensual de la red de sensores*

Protocolo	Tamaño (KB)	Frecuencia por hora	Horas al día	Días al mes	Datos (MB)
HTTP Request	8.82	4	24	31	26.25
HTTP Response	0.19	4	24	31	0.57
Total					26.82 MB

Elaboración propia

Tabla 14 *Costos operativos mensuales*

Descripción		Cantidad	Precio Unitario	Subtotal
Red de Adquisición de Datos	Energía mensual (costo kWh)	34.2	S/0.55	S/18.81
	Plan de datos móvil mínimo (4GB)	1	S/20.00	S/20.00
	Mantenimiento del sistema de puesta a tierra	1	S/40.00	S/40.00
Red de Distribución de Datos	Hosting Virtual Private Server (2 vCPU, 8 GB RAM, 100 GB SSD) + Seguridad web	1	S/157.00	S/157.00
			Total	S/235.81

Elaboración propia



Conclusiones

1. Se eligió la arquitectura Modelo Vista Controlador (MVC), la cual permitió dividir la estructura del software en tres capas para disminuir el impacto, en caso existan cambios en la estructura del sistema, durante la operación.
2. La selección del lenguaje de programación Java agilizó la integración con la arquitectura MVC. En ese sentido, también se seleccionó el servidor web Apache Tomcat, ya que soporta Java desde su origen; además, es de uso gratuito y de código abierto.
3. Es claro que, debido al crecimiento exponencial de Internet y el desarrollo de software, los sistemas de monitoreo se puedan implementar sobre aplicaciones web. En ese sentido, la adopción de tecnologías estándares, como Web Services REST, permiten garantizar la interoperabilidad del sistema de monitoreo, tal como se ha demostrado durante las pruebas de integración explicadas en la sección 4.2.4.
4. Se seleccionó, diseñó e implementó el esquema de Base de Datos bajo un modelo relacional SQL, lo cual permitió disminuir la complejidad y asegurar un soporte adecuado a los datos, debido a que este modelo cuenta con amplia documentación y referencias en el mercado.
5. Se diseñó e implementó las funciones y la lógica principal del sistema de monitoreo; gracias a esto, se definió a detalle los flujos de interacción entre el cliente, servidor web y la red de sensores, considerando los aspectos de seguridad, tal como se especificó en la sección 3.4.4.
6. Se implementó la interfaz web gráfica que permite que los clientes tengan una interacción intuitiva y de fácil acceso a las distintas opciones disponibles en el sistema de monitoreo.
7. Se diseñó e implementó el sistema de comunicación entre la red de sensores y la aplicación web, lo cual permitió una integración simple, ligera y flexible, ya que está basada en el protocolo HTTP con los datos en formato JSON.

8. El resultado del procesamiento de las mediciones recolectadas por los sensores permite detectar condiciones en las cuales se podrían establecer mecanismos para regular los parámetros climáticos del invernadero como temperatura, humedad relativa, CO₂, etc. De la misma forma, se pueden relacionar con otros factores que contribuirían a mejorar la calidad y elevar la productividad, tales como, la cantidad de abono, nutrientes, agua, calor, luz, etc.
9. Se verificó en esta tesis que el diseño e implementación de un sistema de monitoreo para un invernadero experimental, basado en una red de sensores, representa una alternativa tecnológica viable de adaptación al cambio climático, la cual se sustentó a lo largo del desarrollo, pruebas y análisis de costos del presente estudio.
10. La presente tesis ha permitido expandir mis conocimientos sobre las posibilidades de aplicar la tecnología a favor de la agricultura con la intención de adaptarnos al cambio climático y, de esta forma, generar un impacto positivo en la economía y el bienestar social, al contribuir con la seguridad alimentaria de nuestro país y del mundo.

Recomendaciones

1. Realizar pruebas de estrés para determinar la máxima capacidad de conexiones simultáneas que puede soportar el servidor sobre el cual se ejecuta la aplicación web y la Base de Datos.
2. Analizar la futura migración hacia un modelo de Base Datos “No Relacional” en caso de que la solución se expanda a múltiples invernaderos en distintas localidades y, en consecuencia, se tenga un mayor volumen de datos.
3. Establecer políticas de control de accesos a la aplicación web que contemple distintos niveles de privilegios para restringir acciones de gestión, como la creación, edición y eliminación de los componentes del sistema de monitoreo.
4. Los componentes electrónicos de la red de adquisición de datos pueden ser reemplazados por otros a elección, siempre y cuando se respete el sistema de comunicación establecido con la aplicación web.
5. Como trabajo futuro se propone que, a partir del análisis y procesamiento de los datos recolectados, se tendría la capacidad de elaborar modelos predictivos que ayuden a determinar la cantidad exacta de factores que optimicen la calidad de los vegetales y eleven su productividad.

Bibliografía

- Amazon Web Services. (2020). *AWS IoT*. Obtenido de <https://aws.amazon.com/iot>
- Amazon Web Services. (2020). *Real-time operating system for microcontrollers*. Obtenido de <https://aws.amazon.com/freertos>
- Arduino. (2020). Obtenido de <https://www.arduino.cc>
- Arduino. (2020). *Arduino UNO REV3*. Obtenido de <https://store.arduino.cc/usa/arduino-uno-rev3>
- ATIM. (2020). Obtenido de <https://www.atim.com>
- Camps Paré, R., Casillas Santillán, L. A., Costal Costa, D., Gibert Ginestà, M., Martín Escofet, C., & Pérez Mora, O. (2005). *Bases de Datos*. Barcelona, España: Eureca Media.
- Control Climático en Invernaderos*. (s.f.). Obtenido de <https://www.infoagro.com>
- Garbarino, J. (2011). *Protocolos para Redes Inalámbricas de Sensores*. Buenos Aires, Argentina.
- Google Cloud. (2020). *Cloud IoT Core*. Obtenido de <https://cloud.google.com/iot/docs>
- Google Cloud. (2020). *Google Cloud Products*. Obtenido de <https://cloud.google.com/products>
- Guía de Invernaderos*. (s.f.). Obtenido de <https://www.hydroenv.com.mx>
- Hoan-Suk, C., & Woo-Seop, R. (2012). *Distributed Semantic Sensor Web Architecture*. Daejeon, Korea.

Instituto Nacional de Estadística e Informática. (2018). *POBLACIÓN ECONÓMICAMENTE ACTIVA OCUPADA, SEGÚN PRINCIPALES CARACTERÍSTICAS, 2007-2018*.

Instituto Nacional de Estadística e Informática. (2019). *PERÚ: PRODUCTO BRUTO INTERNO SEGÚN ACTIVIDAD ECONÓMICA (NIVEL 54), 2007 - 2019*.

Intergovernmental Panel on Climate Change. (2007). *CAMBIO CLIMÁTICO 2007 IMPACTO, ADAPTACIÓN Y VULNERABILIDAD*.

Libelium. (2020). Obtenido de <https://www.libelium.com>

Lith, A., & Mattsson, J. (2010). *Investigating Storage Solutions for Large Data*. Göteborg, Suecia.

López Hernández, J. C., & Pérez-Parra, J. (2006). *Evolución de las Estructuras de Invernadero*. Obtenido de <http://www.publicacionescajamar.es>

LoRa Alliance. (2021). *About LoRaWAN*. Obtenido de <https://lora-alliance.org/about-lorawan/>

Microsoft Azure. (2018). *IoT Security Architecture*. Obtenido de <https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-security-architecture>

Microsoft Azure. (2019). *Azure IoT Hub*. Obtenido de <https://docs.microsoft.com/en-us/azure/iot-hub/about-iot-hub>

Ministerio de Agricultura. (2012). *Plan de Gestión de Riesgos y Adaptación al Cambio Climático en el sector Agrario*. Lima.

Ministerio de Agricultura y Riego. (2018). *Memoria Anual*.

Montes, C. (s.f.). *¿Por qué elegir un Domo?* Obtenido de <https://domosgeodesicos.es>

Mora, J. T. (2011). *Arquitectura de Software para Aplicaciones Web*. México D.F.

- Morales Machuca, C. A. (2014). *Estado del Arte: Servicios Web*. Bogotá, Colombia.
- Myong-Jin , R., Sun-Ok, C., Ki-Dae, K., Yun-Kun, H., Seung-Oh, H., Sang-Kun, H., . . .
Hak-Hun, K. (2012). *DETERMINATION OF SENSOR LOCATIONS FOR
MONITORING OF GREENHOUSE AMBIENT ENVIRONMENT*.
- Olarte, C. A. (2014). *Arquitecturas de Bases de Datos*. Cali, Colombia.
- Oracle. (2020). *Configuring Web Services with WSDL, SOAP, and the WSDL Generator*.
Obtenido de <https://docs.oracle.com/en>
- Oracle Corporation. (2020). *Why MySQL?* Obtenido de <https://www.mysql.com/why-mysql/>
- Palma Gómez, A. M. (2009). *Análisis de Protocolos de Enrutamiento*. Madrid.
- Raspberrypi. (2020). Obtenido de <https://www.raspberrypi.org>
- SENAMHI. (2021). *Datos Hidrometeorológicos a nivel nacional*. Obtenido de
<https://www.senamhi.gob.pe/>
- SIGFOX. (2020). Obtenido de <https://www.sigfox.com>
- Software para Telecomunicaciones II PUCP. (2013). *Clase: Introducción a Base de Datos*.
Lima.
- Sommerville, I. (2011). *Ingeniería de Software*. México: Pearson Educación.
- Suhonen, J., Kohvakka, M., Kaseva, V., Hämmäläinen, T. D., & Hännikäinen, M. (2012). *Low-
Power Wireless Sensor Networks - Protocols, Services and Applications*. Tampere:
Springer.
- Texas Instruments. (2017). *LM35 Precision Centigrade Temperature Sensors*.
- Texas Instruments. (2020). Obtenido de <https://www.ti.com>

The Things Network. (2021). *Get started with LoRaWAN*. Obtenido de <https://www.thethingsnetwork.org/docs/lorawan/architecture.html>

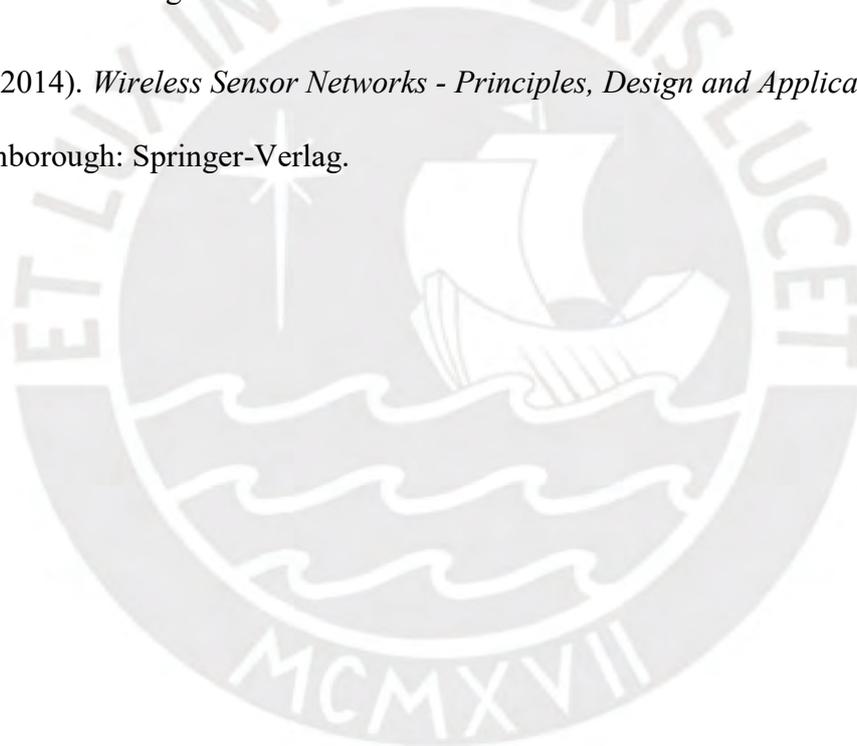
Tipos de Invernaderos. (s.f.). Obtenido de <https://www.novagric.com>

Ubidots. (2020). *Advice and answers from the Ubidots Team*. Obtenido de <https://help.ubidots.com/en/>

Valencia Cabrera, L. (2014). *Introducción a Bases de Datos*. Sevilla, España.

World Wide Web Consortium. (2020). *Web Services Architecture*. Obtenido de <https://www.w3.org>

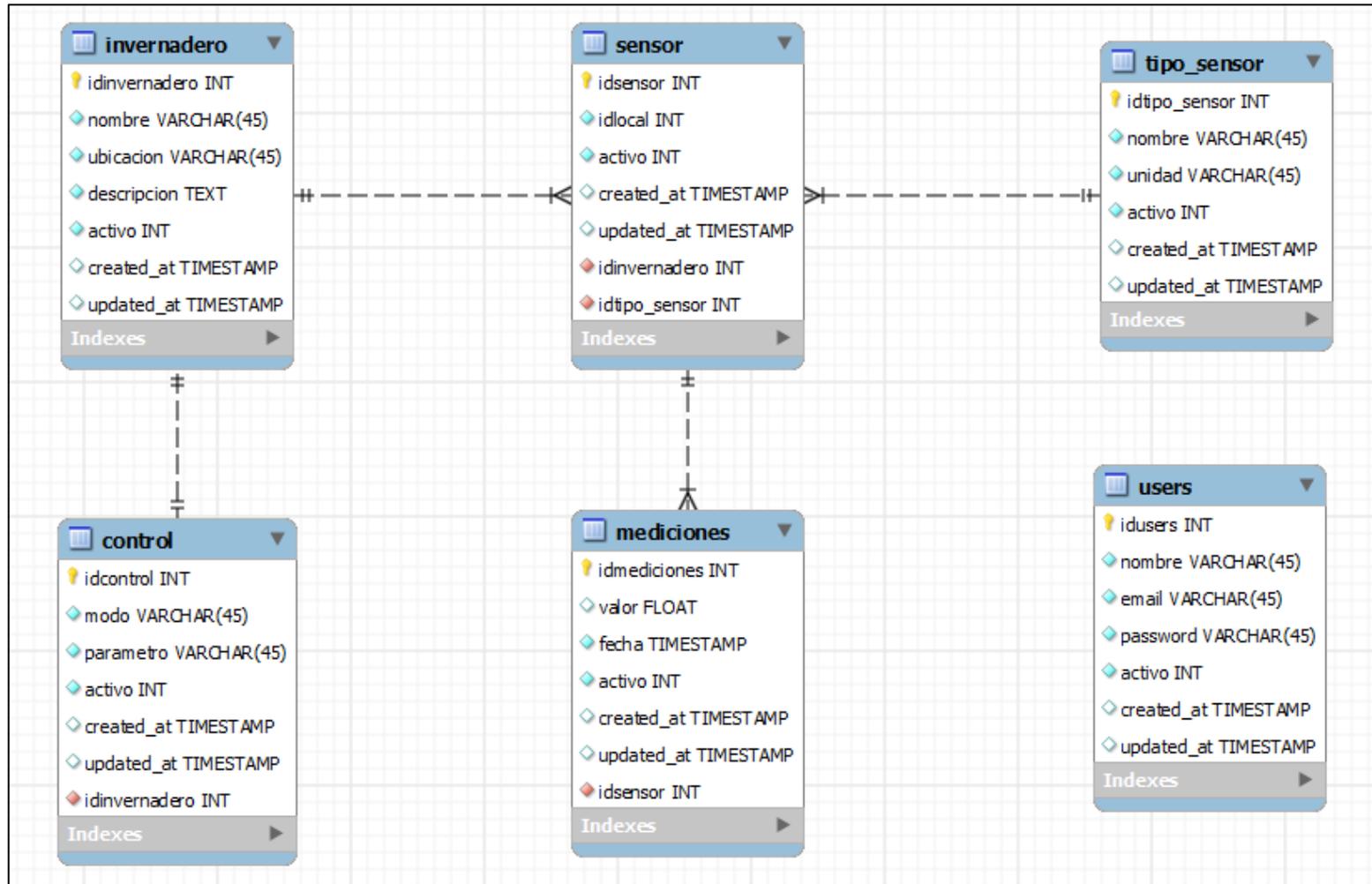
Yang, S.-H. (2014). *Wireless Sensor Networks - Principles, Design and Applications*. Loughborough: Springer-Verlag.



ANEXOS

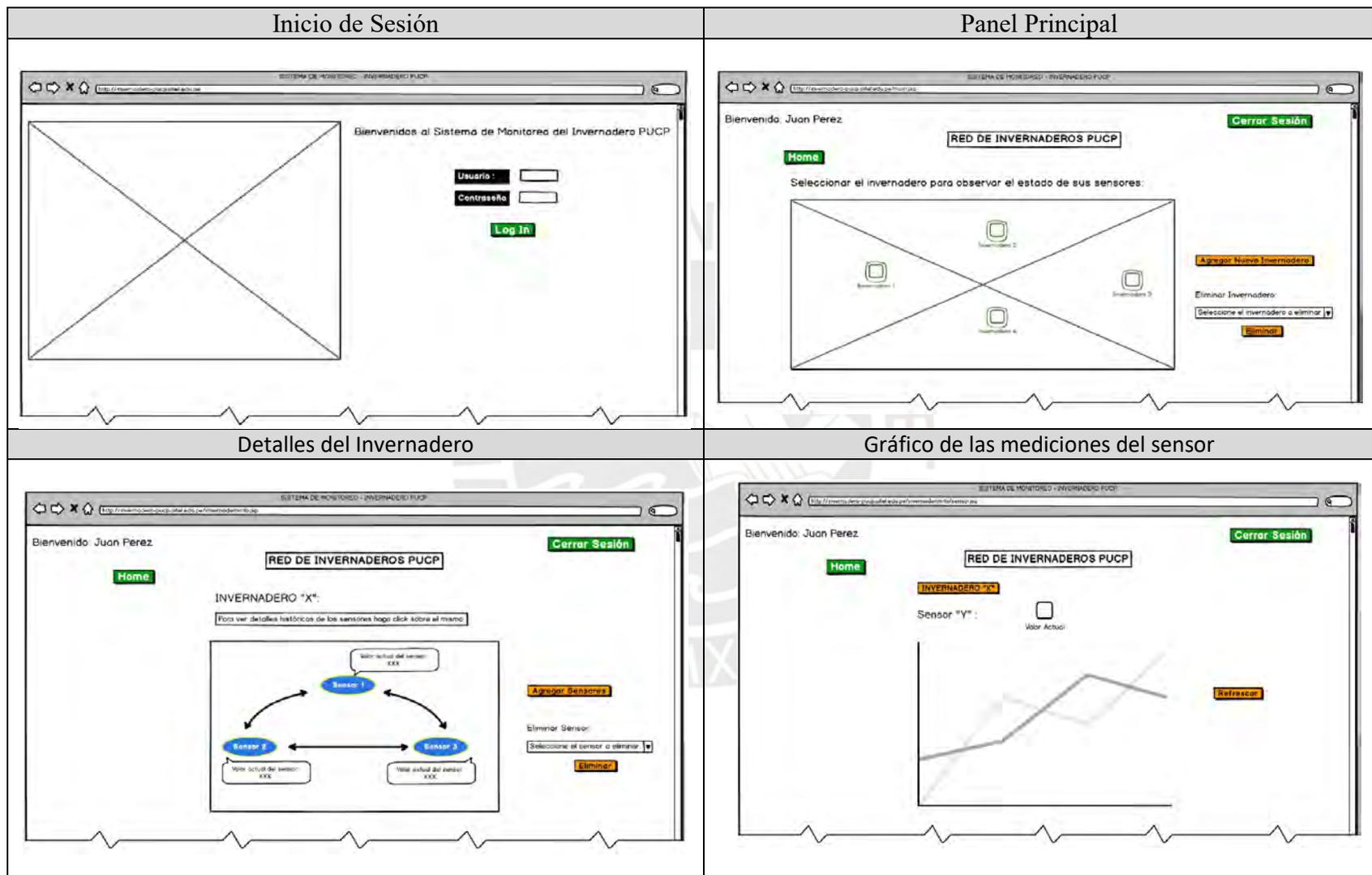


Anexo A. Modelo relacional de la BD en MySQL Workbench



Elaboración propia

Anexo B. Mock Up de la aplicación web



Elaboración propia realizada en Balsamiq

Anexo C. Diagramas de flujo

C.1 Inicio de sesión

La Figura C1 muestra el flujo de inicio de sesión de un usuario, con sus respectivas credenciales, en la aplicación web.

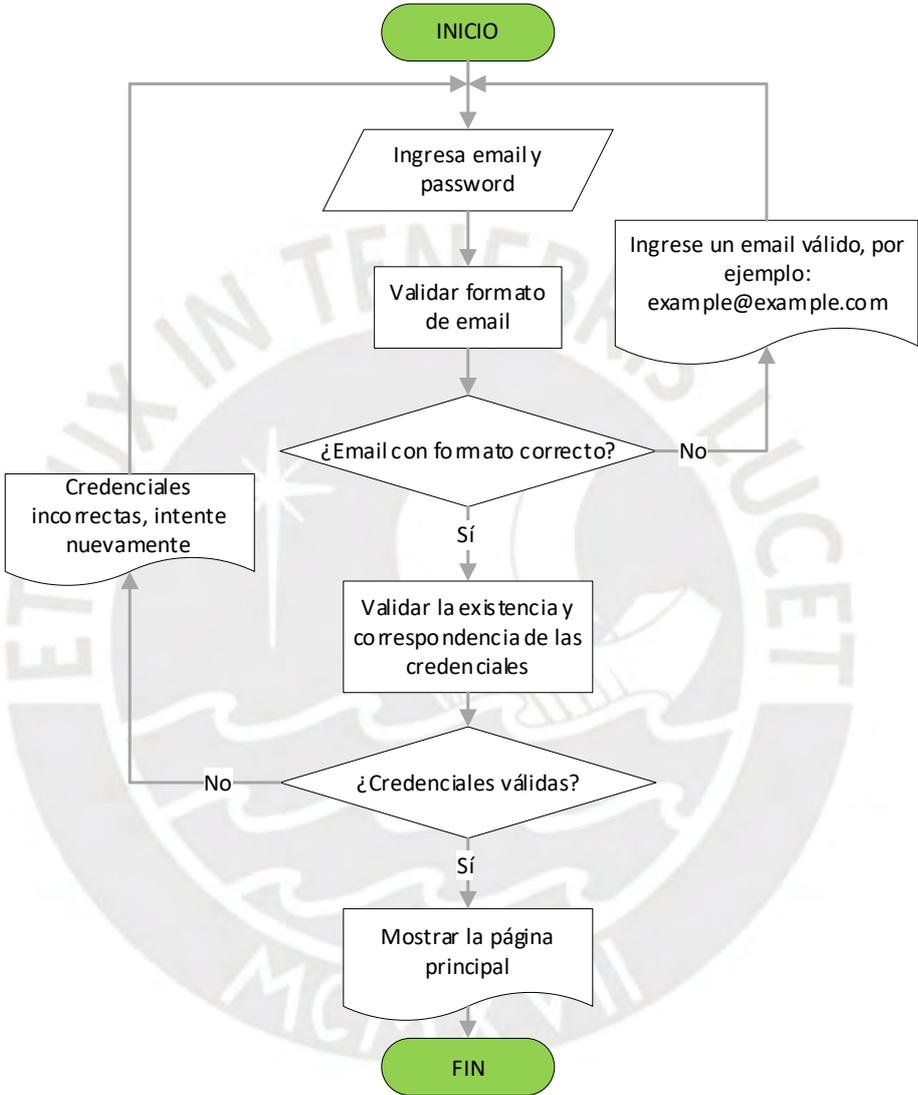


Figura C1 Flujo de inicio de sesión
Elaboración propia

C.2 Registrar nuevo tipo de sensor

La Figura C2 muestra el flujo para registrar un nuevo tipo de sensor.

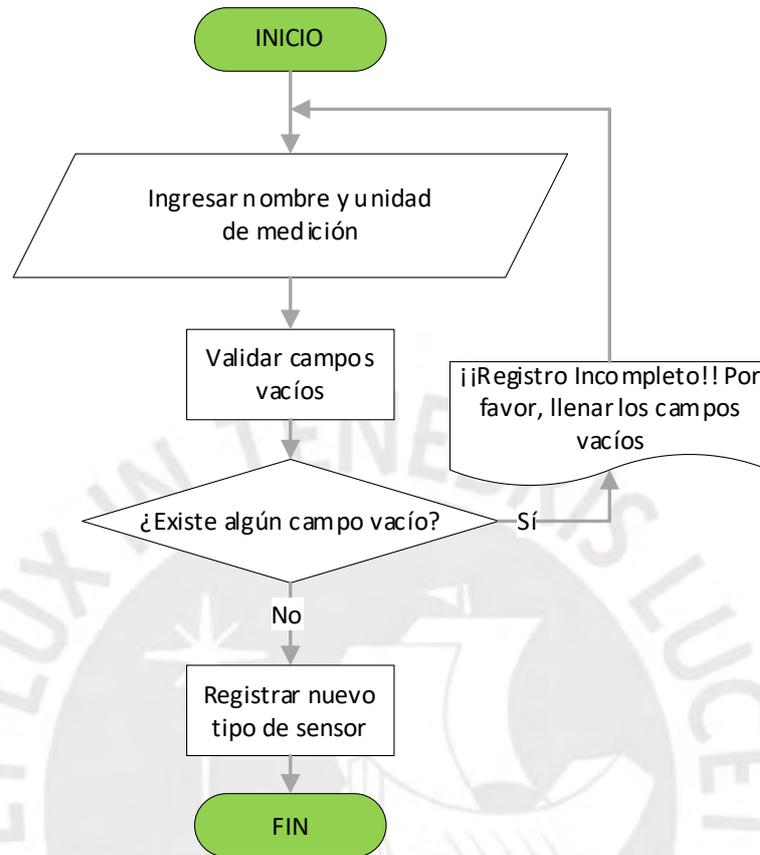


Figura C2 Flujo para registrar un nuevo tipo de sensor
Elaboración propia

C.3 Registrar nuevo invernadero

La Figura C3 muestra el flujo para registrar un nuevo invernadero en la aplicación web.

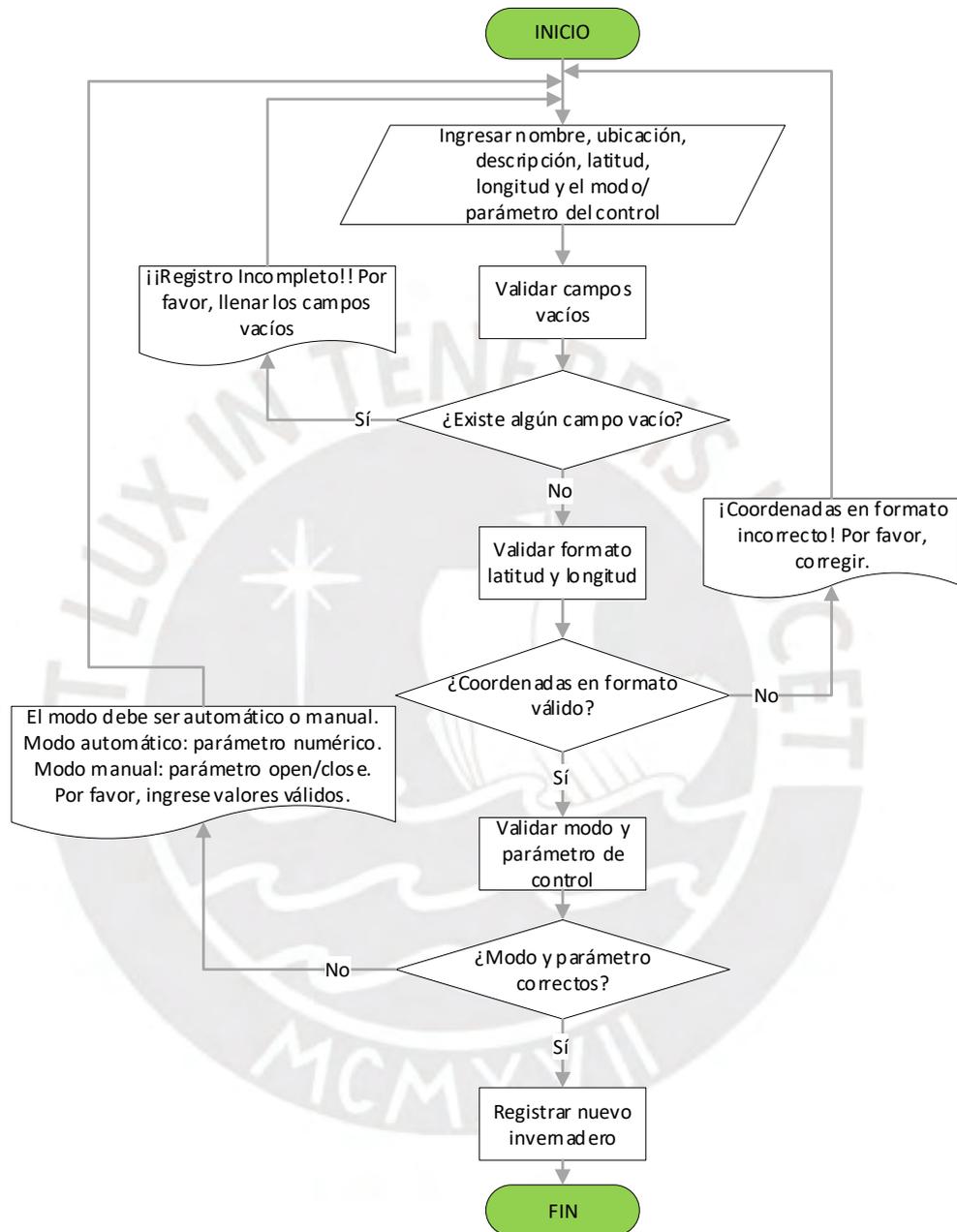


Figura C3 Flujo para registrar un nuevo invernadero
Fuente: Elaboración propia

C.4 Registrar nuevo sensor

La Figura C4 muestra el flujo para registrar un(os) nuevo(s) sensor(es) a un invernadero existente.

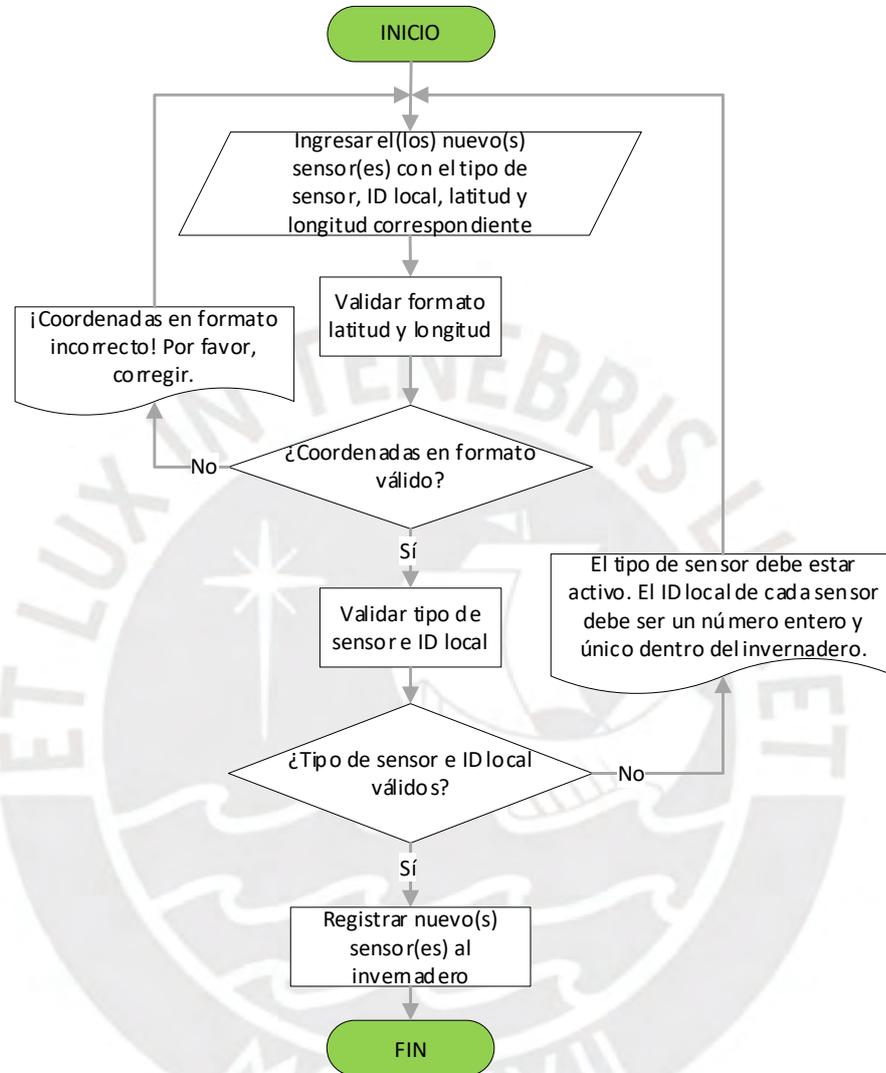


Figura C4 Flujo para registrar un(os) nuevo(s) sensor(es) a un invernadero
Elaboración propia

C.5 Registrar nueva medición

La Figura C5 muestra el flujo para registrar las mediciones enviadas por la red de sensores.

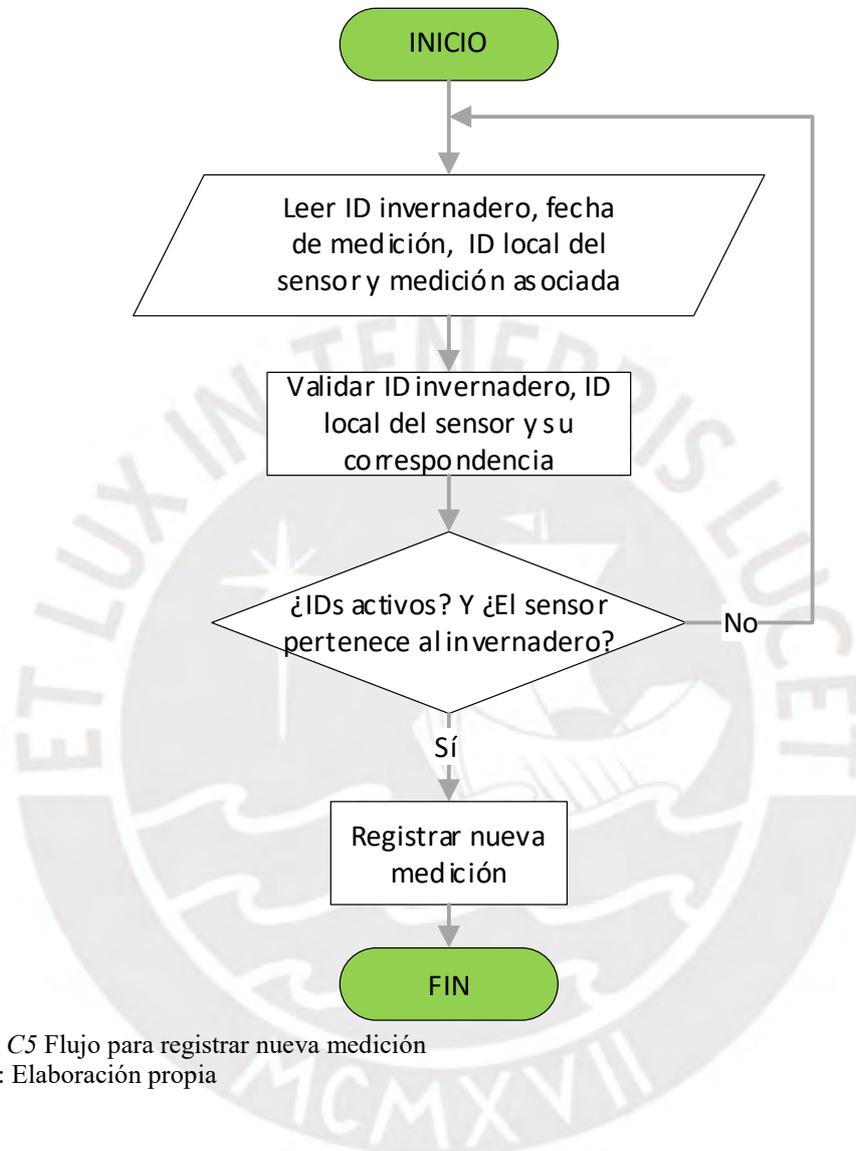


Figura C5 Flujo para registrar nueva medición
Fuente: Elaboración propia

C.6 Configurar el tipo de control de un invernadero

La Figura C6 muestra el flujo para configurar el tipo de control de un invernadero ya sea modo manual (open/close) o automático (valor umbral de temperatura).

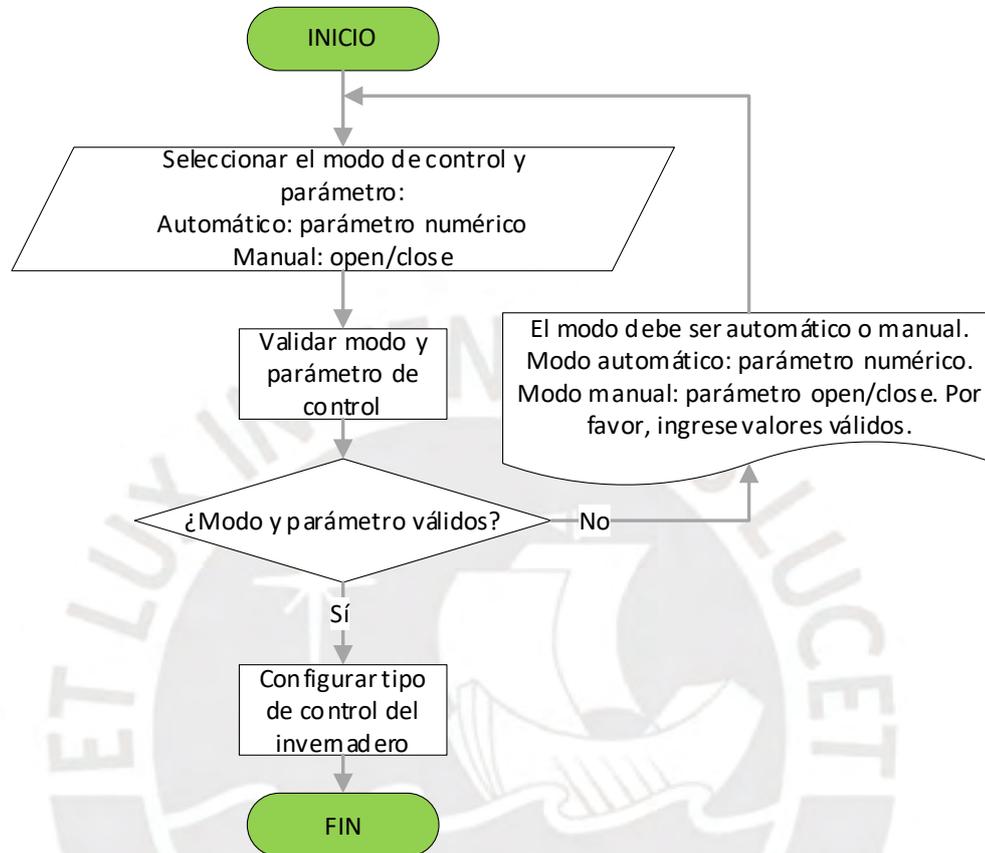


Figura C6 Flujo para configurar el tipo de control del invernadero
Elaboración propia

C.7 Visualizar los detalles de un invernadero

La Figura C7 muestra el flujo para visualizar los sensores de un invernadero y el estado actual del control.

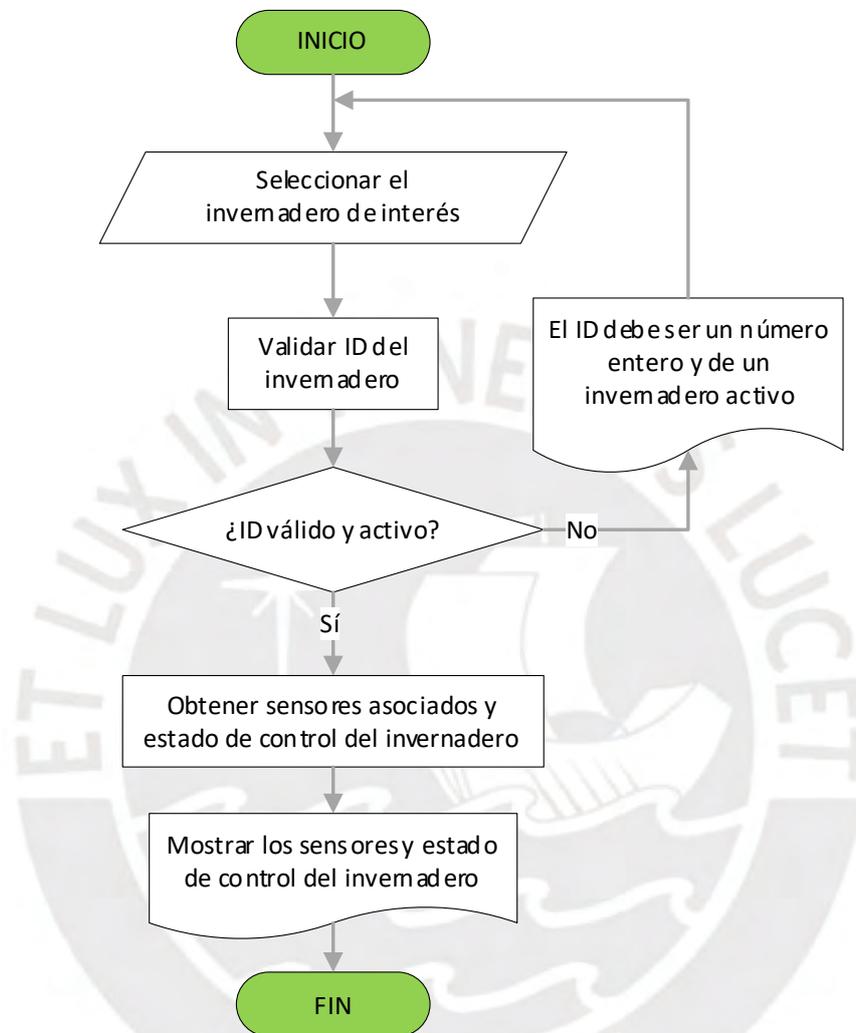


Figura C7 Flujo para visualizar los detalles de un invernadero
Elaboración propia

C.8 Visualizar las mediciones de un sensor

La Figura C8 muestra el flujo para visualizar las mediciones almacenadas de un sensor.

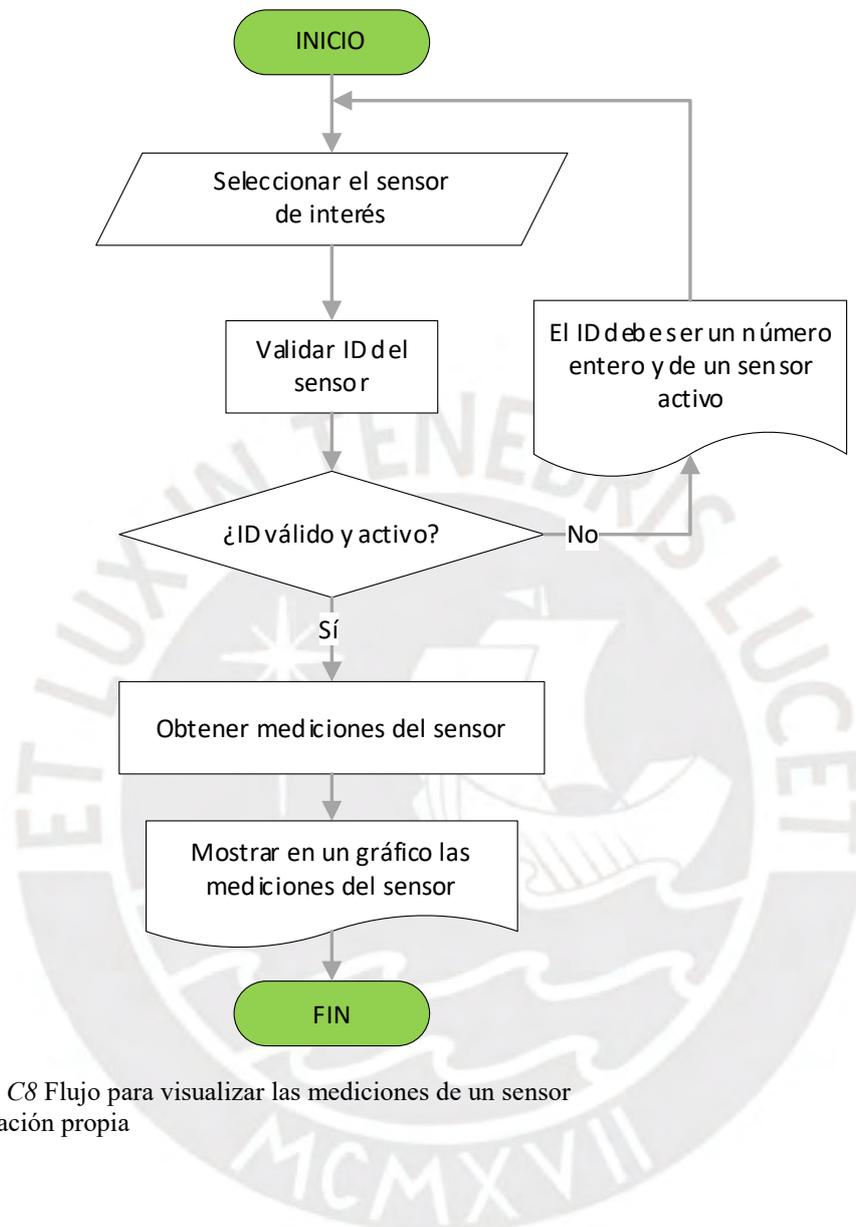


Figura C8 Flujo para visualizar las mediciones de un sensor
Elaboración propia

C.9 Visualizar operaciones basadas en las mediciones de los sensores

La Figura C9 muestra el flujo para visualizar operaciones como: promedio, valor máximo/mínimo y filtrar mediciones de los sensores que pertenecen a un mismo invernadero.

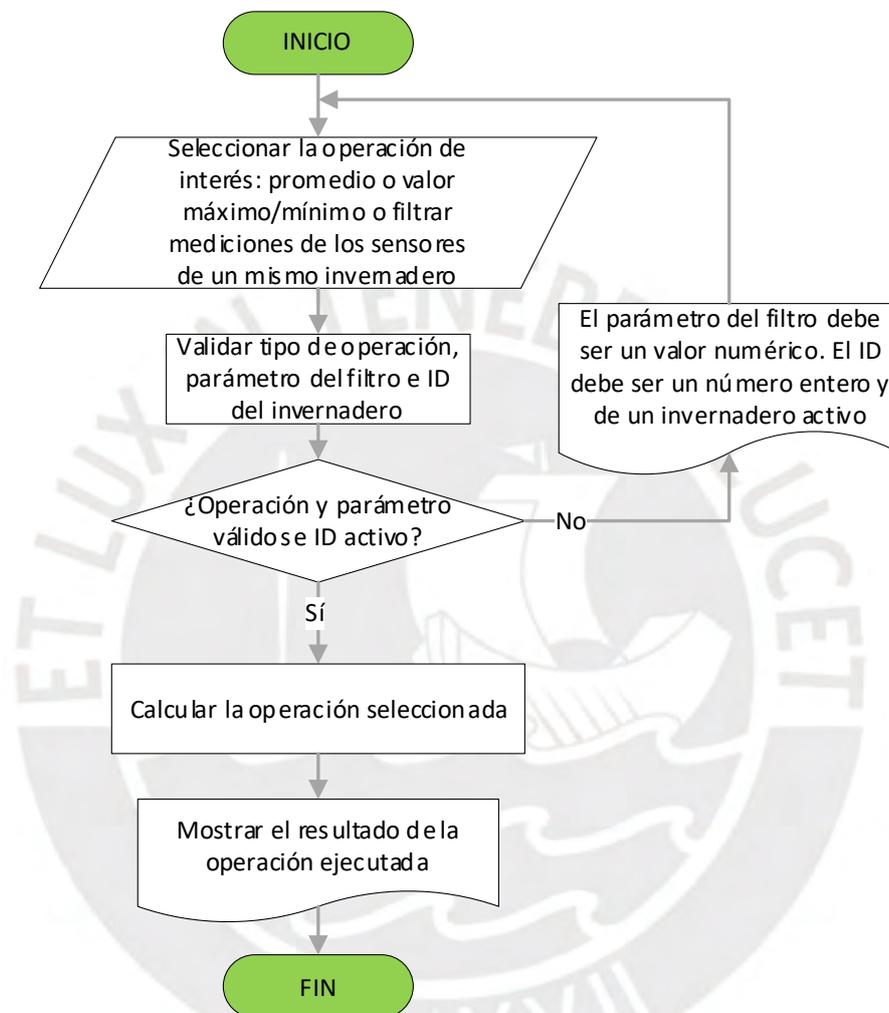


Figura C9 Flujo para visualizar las operaciones basadas en las mediciones de los sensores
Elaboración propia

C.10 Editar atributos de un invernadero

La Figura C10 muestra el flujo para editar los atributos de un invernadero como su nombre, ubicación, latitud, longitud y descripción.

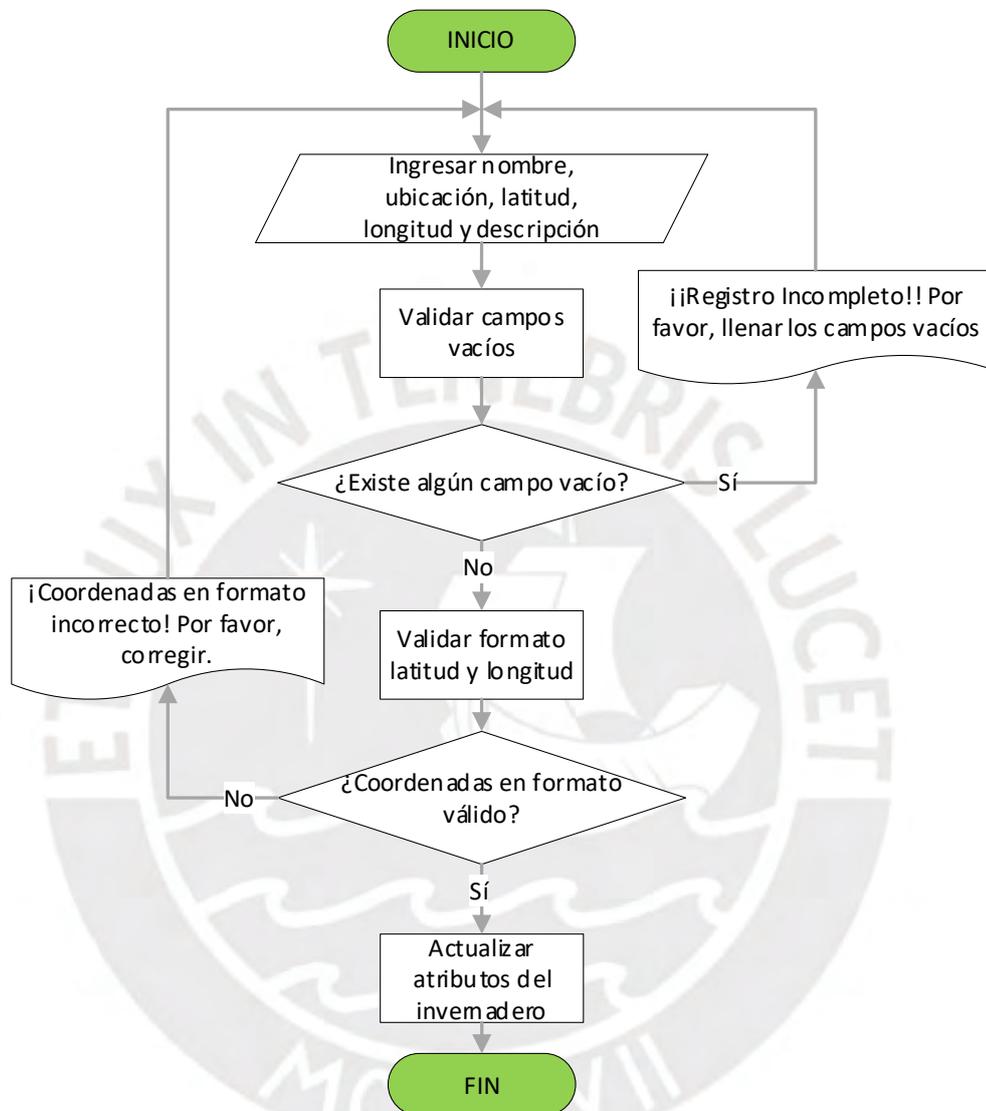


Figura C10 Flujo para editar los atributos de un invernadero
Elaboración propia

C.11 Editar atributos de un sensor

La Figura C11 muestra el flujo para editar los atributos de un sensor como su tipo, latitud, longitud e ID local.

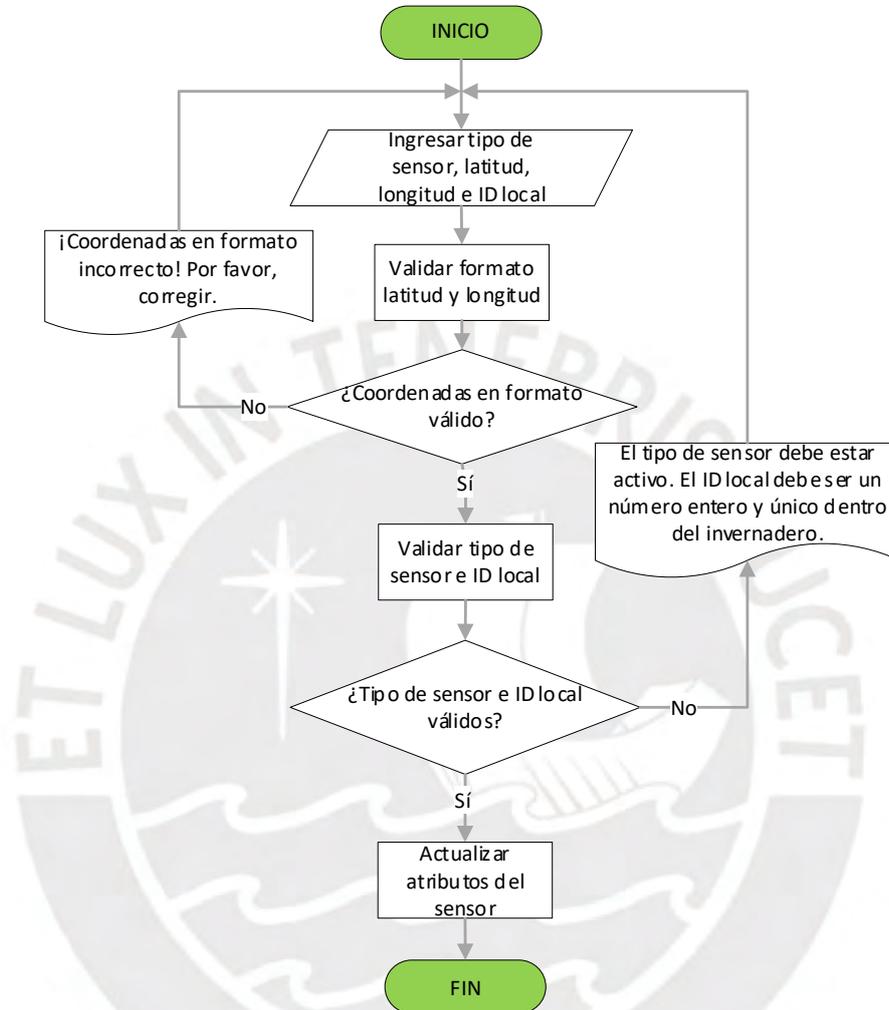


Figura C11 Flujo para editar los atributos de un sensor
Elaboración propia

C.12 Editar atributos de un tipo de sensor

La Figura C12 muestra el flujo para editar los atributos de un tipo de sensor como su nombre y unidad de medición.

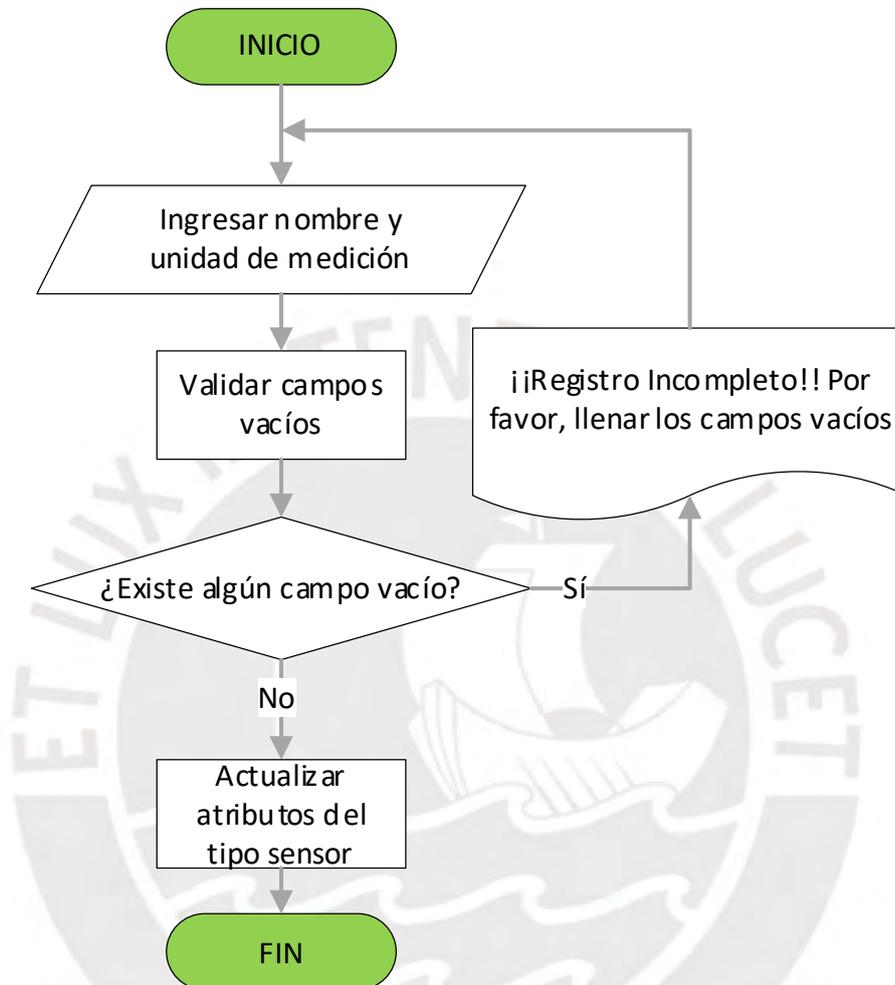


Figura C12 Flujo para editar los atributos de un tipo de sensor
Elaboración propia

C.13 Borrar un sensor

La Figura C13 muestra el flujo para borrar un sensor y sus mediciones correspondientes.

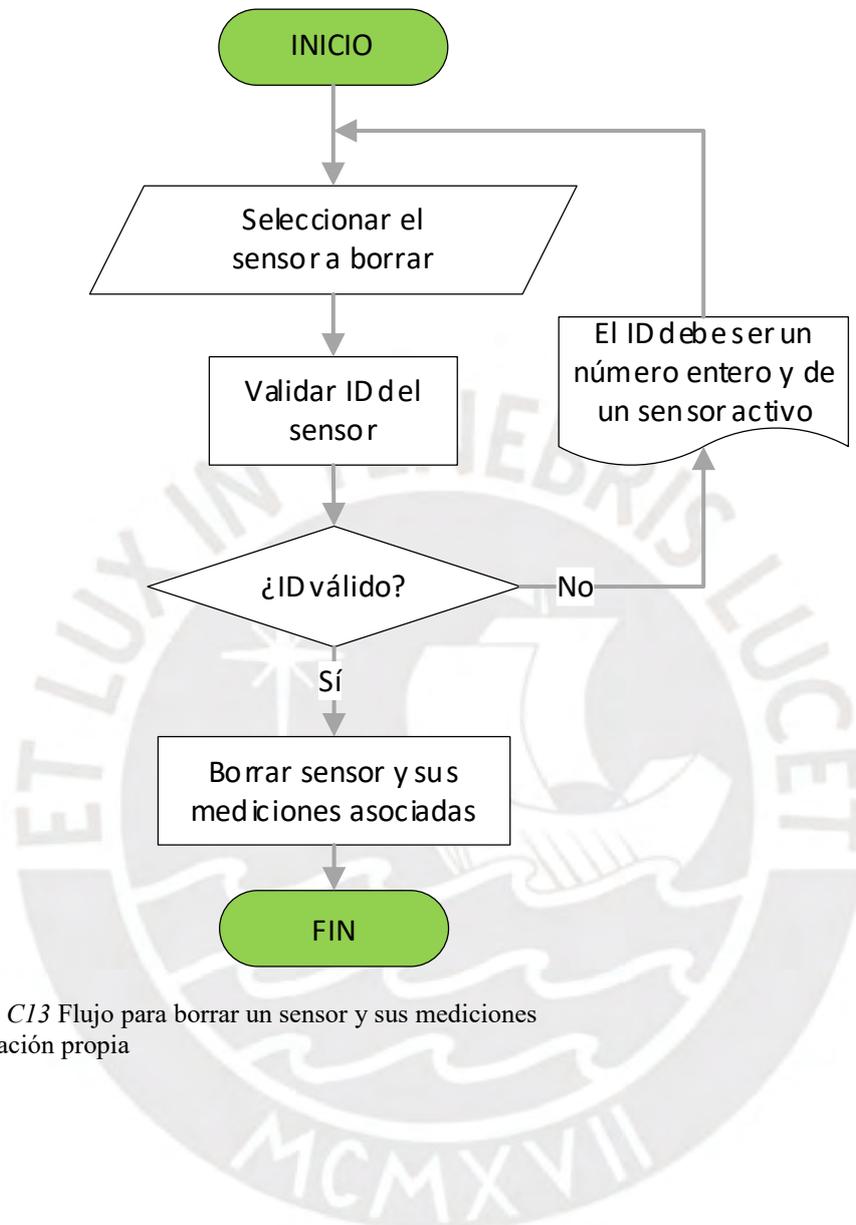


Figura C13 Flujo para borrar un sensor y sus mediciones
Elaboración propia

C.14 Borrar un tipo de sensor

La Figura C14 muestra el flujo para borrar un tipo de sensor siempre y cuando no tenga sensores asociados.

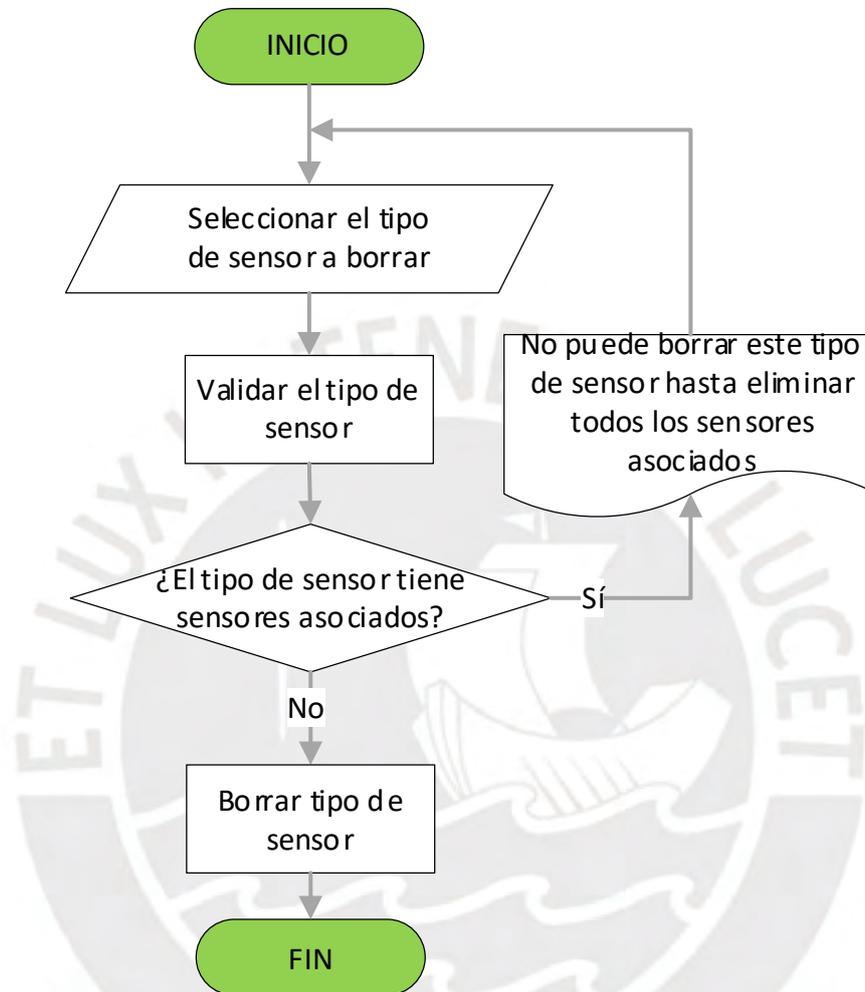


Figura C14 Flujo para borrar un tipo de sensor
Elaboración propia

C.15 Borrar un invernadero

La Figura C15 muestra el flujo para borrar un invernadero y, como consecuencia, también sus sensores asociados, las mediciones correspondientes y el tipo de control.

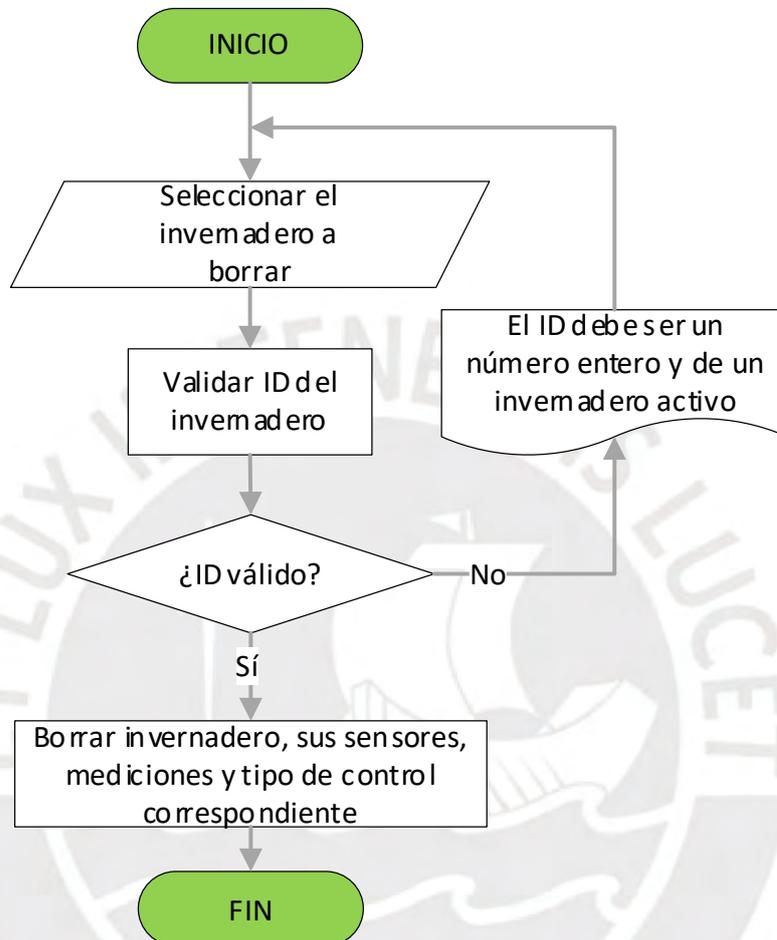


Figura C15 Flujo para borrar un invernadero
Elaboración propia

Anexo D. Interfaz Gráfica

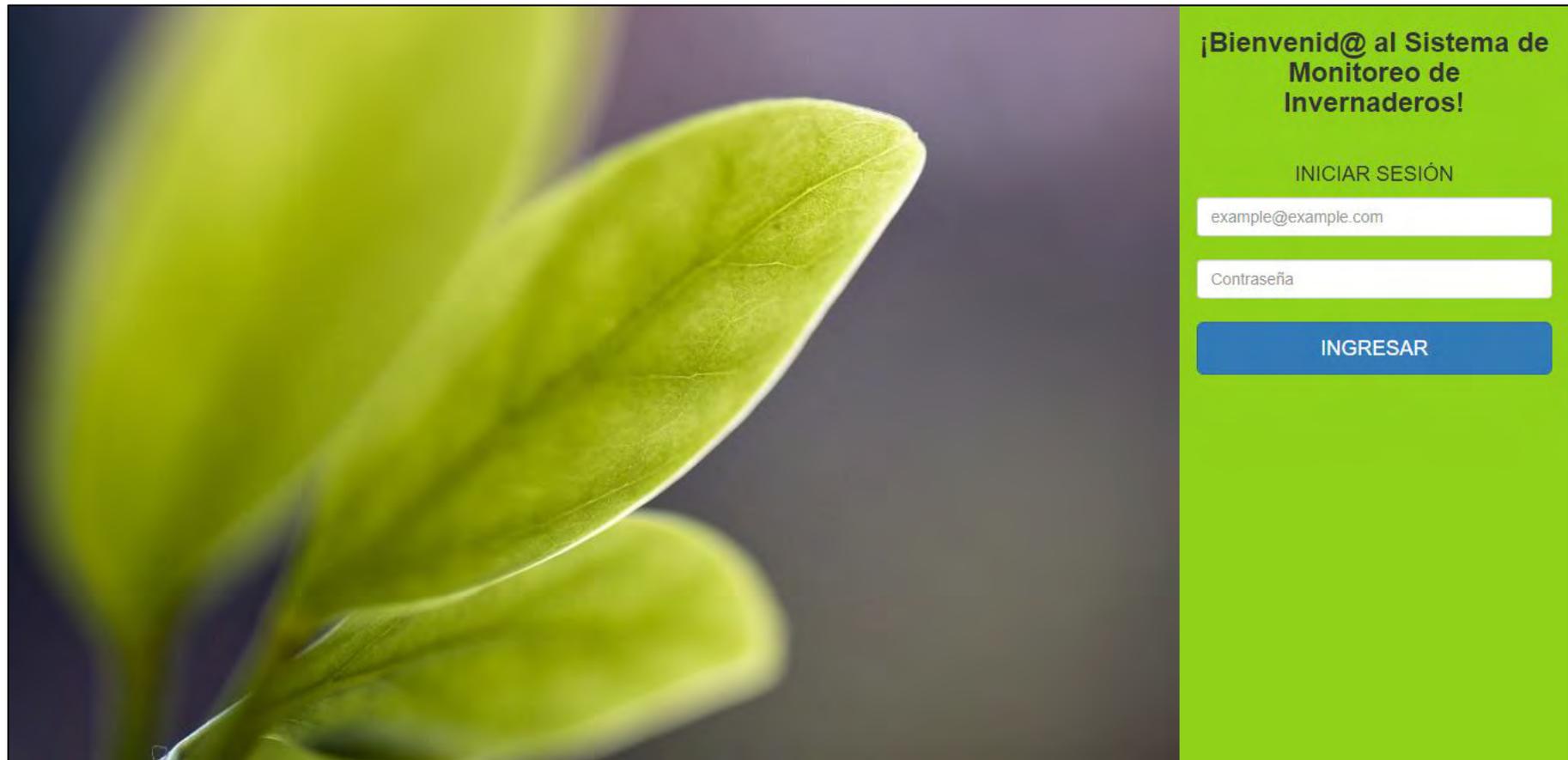


Figura D1 Inicio de sesión
Elaboración propia

RED DE INVERNADEROS

Principal

Seleccionar un invernadero para observar el estado de sus sensores:

ID	Nombre	Ubicación	Latitud	Longitud	Descripción	Opciones
1	Domo1	San Miguel	-12.081443	-77.089813	Invernadero de prueba	 
2	Domo2	Lince	-12.082357	-77.03106	Parque Pedro Ruiz	 

Añadir Nuevo Invernadero Configurar Tipos de Sensor



The map displays the geographical distribution of the greenhouses. A red pin is placed on the coast of Peru, near the city of Lince. The map includes labels for major Peruvian cities like Trujillo, Cusco, Arequipa, and La Paz, as well as regions like ACRE and AMAZONA. The Google logo and copyright information are visible at the bottom of the map interface.

Figura D2 Panel principal
Elaboración propia

RED DE INVERNADEROS

Principal / Nuevo Invernadero

Nuevo Invernadero

Nombre	<input type="text"/>
Ubicación	<input type="text"/>
Descripción	<input type="text"/>
Latitud	<input type="text"/>
Longitud	<input type="text"/>
Tipo de Control	<input type="text" value="Manual"/>
Estado	<input type="text" value="Open"/>

GUARDAR

CANCELAR

Figura D3 Añadir nuevo invernadero
Fuente: Elaboración propia

RED DE INVERNADEROS

Principal / Configurar Tipos de Sensor

Tipos de Sensores Configurados

ID	Nombre	Unidad	Opciones
1	Temperatura	°C	 
2	Humedad Relativa	%	 
3	CO2	PPM	 

Agregar Nuevo Tipo de Sensor:

Nombre: Unidad:

Figura D4 Configurar Tipos de Sensor
Elaboración propia

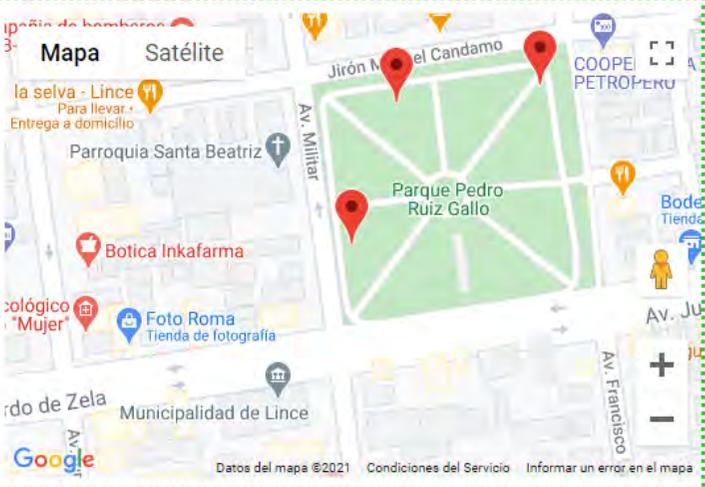
RED DE INVERNADEROS

Principal / Invernadero: Domo2

Domo2 (ID:2)

Haga click en el sensor para observar sus mediciones en tiempo real:

- Temperatura (ID: 20)
- Temperatura (ID: 21)
- Temperatura (ID: 22)



The map shows a street grid in Lince, Peru. A green rectangular area is highlighted, representing the greenhouse. Three red location pins are placed on the map, indicating the positions of the temperature sensors. Landmarks include Parque Pedro Ruiz Gallo, Botica Inkafarma, and the Municipalidad de Lince. The map interface includes 'Mapa' and 'Satélite' tabs, a search bar, and a Google logo.

Agregar Nuevos Sensores Más Opciones

Estado actual del control:

Modo: manual

Estado: close

Editar control

Figura D5 Detalle de un invernadero
Elaboración propia

RED DE INVERNADEROS

Principal / Invernadero: Domo2 / Nuevo Sensor

Sensores Configurados

ID Local	Nombre	
20	Sensor de Temperatura	Borrar
21	Sensor de Temperatura	Borrar
22	Sensor de Temperatura	Borrar

Agregar Nuevo(s) Sensor(es):

Más sensores

Nuevo Sensor: ID Sensor: Latitud: Longitud:

GUARDAR CANCELAR

Figura D6 Añadir nuevo sensor
Elaboración propia

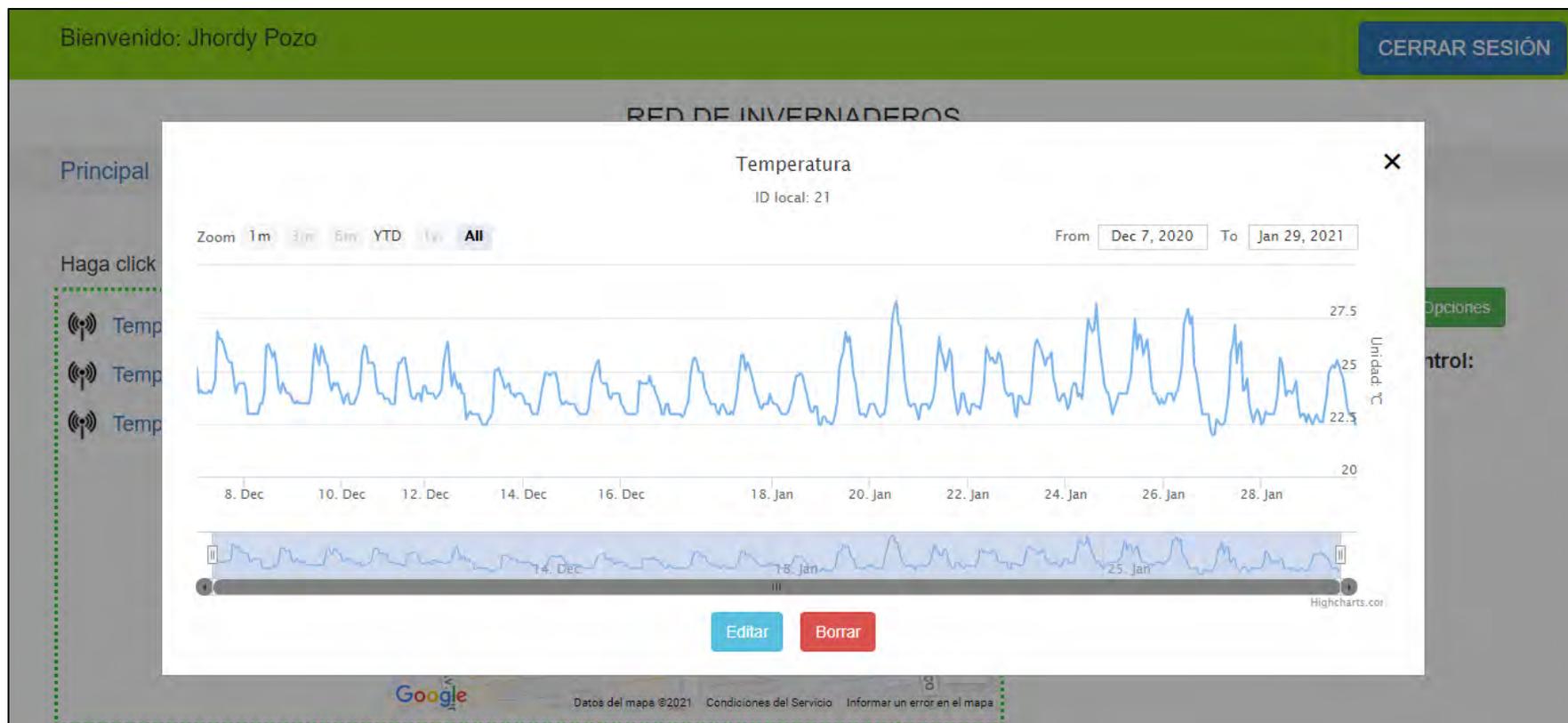


Figura D7 Detalle de un sensor
Fuente: Elaboración propia

RED DE INVERNADEROS

Principal / Invernadero: Domo2 / Editar Sensor

Editar Sensor:

Tipo Sensor: **ID Local:** **Latitud:** **Longitud:**

ACTUALIZAR

CANCELAR

Figura D8 Editar sensor
Elaboración propia



RED DE INVERNADEROS

Principal

Haga click

- Temp
- Temp
- Temp

Haga click en la opción que desea calcular:

Promedio

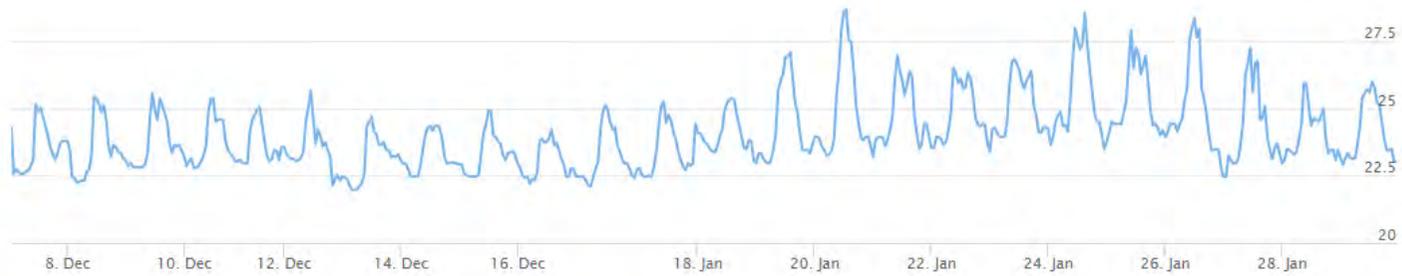
Máx/Min valor

Filtrar Medidas

Promedio de Sensores

Zoom 1m 3m 6m YTD 1y All

From Dec 7, 2020 To Jan 29, 2021



Opciones

Control:

Figura D9 Más Opciones. Promedio de mediciones de sensores
Elaboración propia



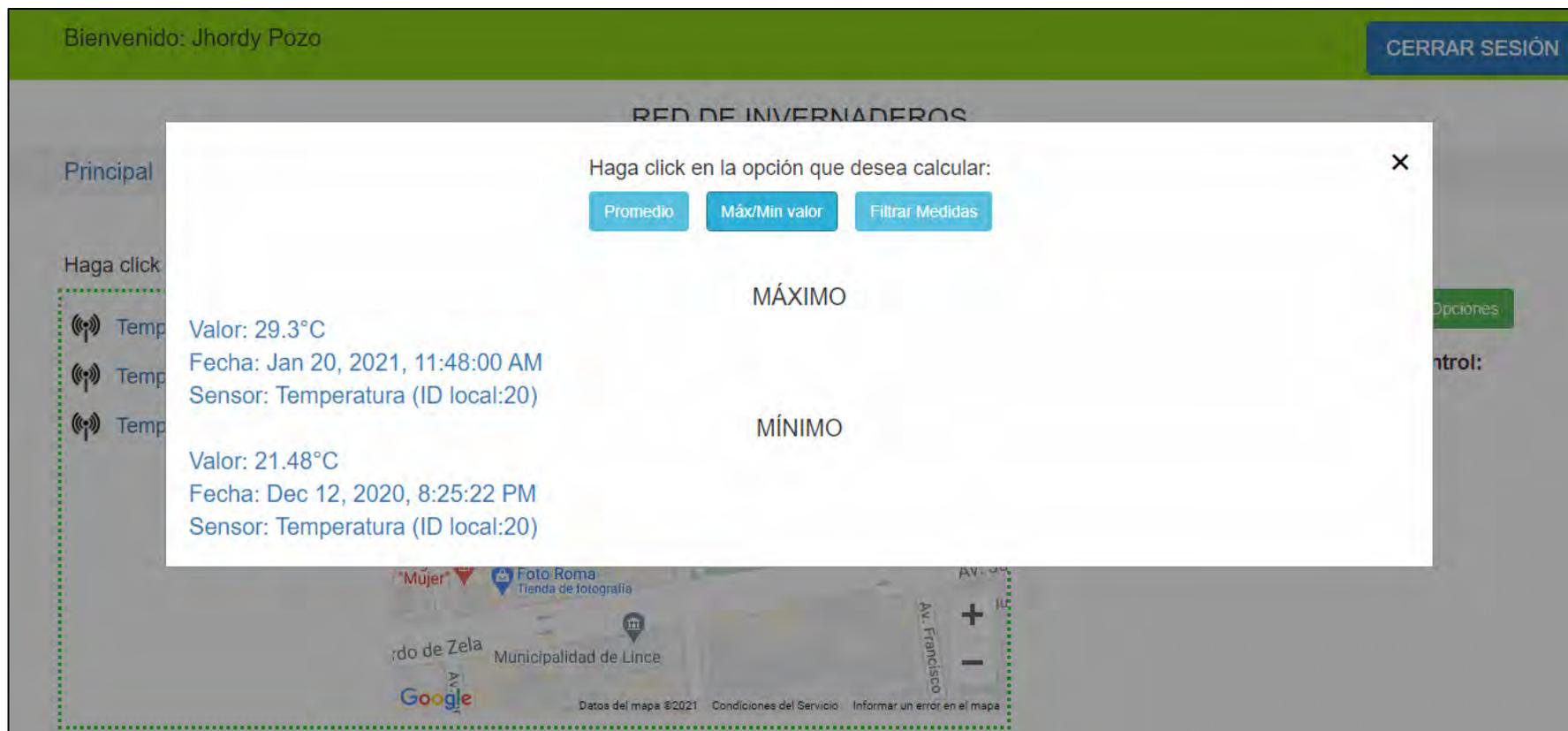


Figura D10 Más Opciones. Valor máximo/mínimo
Elaboración propia

RED DE INVERNADEROS

Principal

Haga click en la opción que desea calcular:

- Promedio
- Máx/Min valor
- Filtrar Mediciones

Menor Igual a | 35 |

Mediciones Filtradas

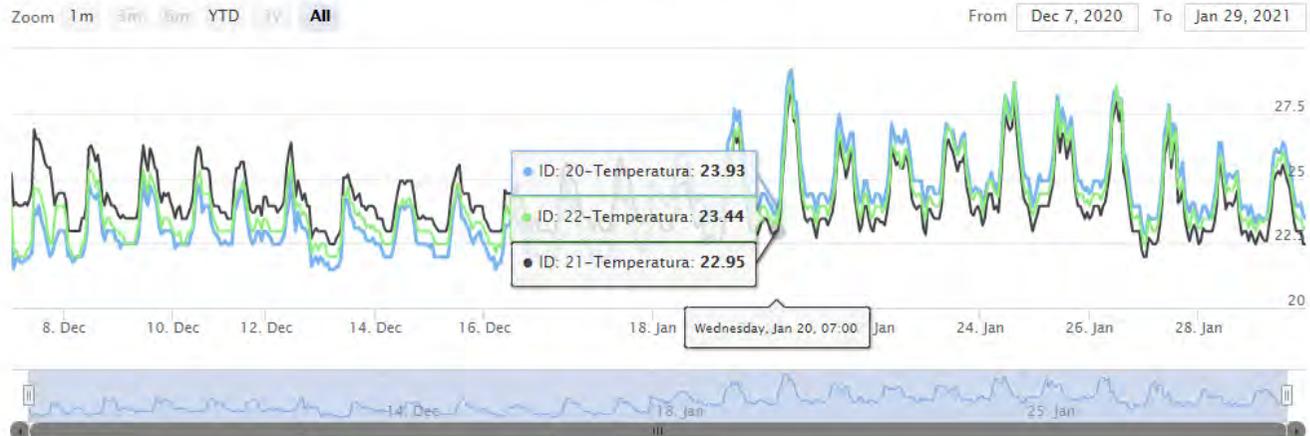


Figura D11 Más opciones. Filtrar mediciones
Elaboración propia



Anexo E. Código de Programación Arduino

```
#include <ArduinoJson.h>
#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>
#include <NTPClient.h>

EthernetUDP ntpUDP;
NTPClient timeClient(ntpUDP);

EthernetClient client;

int sensor_T1 = A0;
int sensor_T2 = A3;
int sensor_T3 = A5;
#define pinLED 2

const char* server = "192.168.1.31"; // IP servidor
const char* resource = "/InvernaderoWatcher/InfoServlet"; //recurso servidor
const unsigned int PORT = 8084;
const unsigned long BAUD_RATE = 9600; // serial connection speed
const unsigned long HTTP_TIMEOUT = 10000; // max response time from server
const size_t MAX_CONTENT_SIZE = 512; // max size of the HTTP response

// Iniciamos el puerto Serial
void initSerial() {
  Serial.begin(BAUD_RATE);
  while (!Serial) {
    ; // esperar hasta que el puerto serial inicie
  }
  Serial.println("Serial ready");
}

// Iniciamos el puerto Ethernet
void initEthernet() {
  byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
  if (!Ethernet.begin(mac)) {
    Serial.println("Failed to configure Ethernet");
    return;
  }
}
```

```

Serial.println("Ethernet ready");
delay(1000);
}
// Abrir la conexión HTTP hacia el servidor
bool connect(const char* hostName) {
    Serial.print("Connect to ");
    Serial.println(hostName);
    bool ok = client.connect(hostName, PORT);
    Serial.println(ok ? "Connected" : "Connection Failed!");
    return ok;
}
// Pausa de un minuto
void wait() {
    Serial.println("Wait 60 seconds");
    delay(60000);
}
// Omitir las cabeceras de la respuesta HTTP del servidor
bool skipResponseHeaders() {
    // HTTP headers end with an empty line
    char endOfHeaders[] = "\r\n\r\n";
    client.setTimeout(HTTP_TIMEOUT);
    bool ok = client.find(endOfHeaders);
    if (!ok) {
        Serial.println("No response or invalid response!");
    }
    return ok;
}
// Leer el contenido de la respuesta HTTP del servidor
void readResponseContent(char* content, size_t maxSize) {
    size_t length = client.readBytes(content, maxSize);
    content[length] = 0;
    Serial.println(content);
}
// Cerrar la conexión con el servidor
void disconnect() {
    Serial.println("Disconnect");
}

```

```
client.stop();
}

/////--INICIO PROGRAMA--/////

void setup() {
  initSerial();
  initEthernet();
  timeClient.begin();
  pinMode(pinLED,OUTPUT);
  digitalWrite(pinLED,LOW);
}

void loop() {
  float T1 = (5.0*analogRead(sensor_T1)*100.0)/1023.0;
  float T2 = (5.0*analogRead(sensor_T2)*100.0)/1023.0;
  float T3 = (5.0*analogRead(sensor_T3)*100.0)/1023.0;
  timeClient.update();
  unsigned long timemediciondomo = timeClient.getEpochTime();
  StaticJsonBuffer<200> jsonBuffer;
  JsonObject& data = jsonBuffer.createObject();
  data["timemedicion"] = timemediciondomo;
  data["iddomo"] = 1;
  JsonArray& sensado = data.createNestedArray("sensado");
  JsonObject& sensor1 = jsonBuffer.createObject();
  JsonObject& sensor2 = jsonBuffer.createObject();
  JsonObject& sensor3 = jsonBuffer.createObject();
  sensor1["id"] = 10;
  sensor1["valor"] = T1;
  sensor2["id"] = 11;
  sensor2["valor"] = T2;
  sensor3["id"] = 12;
  sensor3["valor"] = T3;
  sensado.add(sensor1);
  sensado.add(sensor2);
  sensado.add(sensor3);

  data.prettyPrintTo(Serial);
```

```

if (connect(server)) {
    Serial.print("POST ");
    Serial.println(resource);
    client.print("POST ");
    client.print(resource);
    client.println(" HTTP/1.1");
    client.print("Host: ");
    client.println(server);
    client.println("Content-Type: application/json");
    client.print("Content-Length: ");
    client.println(data.measureLength()); //LONGITUD DATA
    client.println();
    data.printTo(client); //DATA
    client.println();
    Serial.print("Content-Length: ");
    Serial.println(data.measureLength());
    Serial.println(Ethernet.localIP());
    String temp="";
    if (skipResponseHeaders()) {
        char response[MAX_CONTENT_SIZE];
        readReponseContent(response, sizeof(response));
        JsonObject& object = jsonBuffer.parseObject(response);
        temp = object.get<String>("ejecuta");
        Serial.println(temp);
        if(temp == "open"){
            digitalWrite(pinLED,HIGH);
        }
        if(temp == "close"){
            digitalWrite(pinLED,LOW);
        }
    }
    disconnect();
}
delay(900000);
}

```

Anexo F. Código de Programación Java Web

El código Java del proyecto se encuentra en el repositorio privado GitHub:

<https://github.com/JhordyPozo/InvernaderoWatcher.git>

Anexo G. Configuración Access Point Modo Bridge

Según el manual del fabricante TPLINK para configurar el producto Access Point TL-WA901ND en modo Bridge, se debe seguir los siguientes pasos:

• **Bridge with AP Mode** Combines two local networks via wireless connection.



1. Connect the access point according to Step A and B in the diagram.

2. Power on the access point, wait until the Power (P) and Wireless (W) LEDs are lit and stable, and use the default SSID and Password printed on the label of the access point to join its Wi-Fi network.

3. Launch a web browser and enter <http://tplinkap.net>. Then log in using admin (all lowercase) for both username and password.

4. Click **Quick Setup**, select **Bridge with AP mode** and click **Next**. Follow the step-by-step instructions to complete the configuration.

5. Relocate the access point to a suitable place. For details, please refer to FAQ > Q5.

6. Now, connect your wireless devices to the Wi-Fi network using the access point's SSID and password.

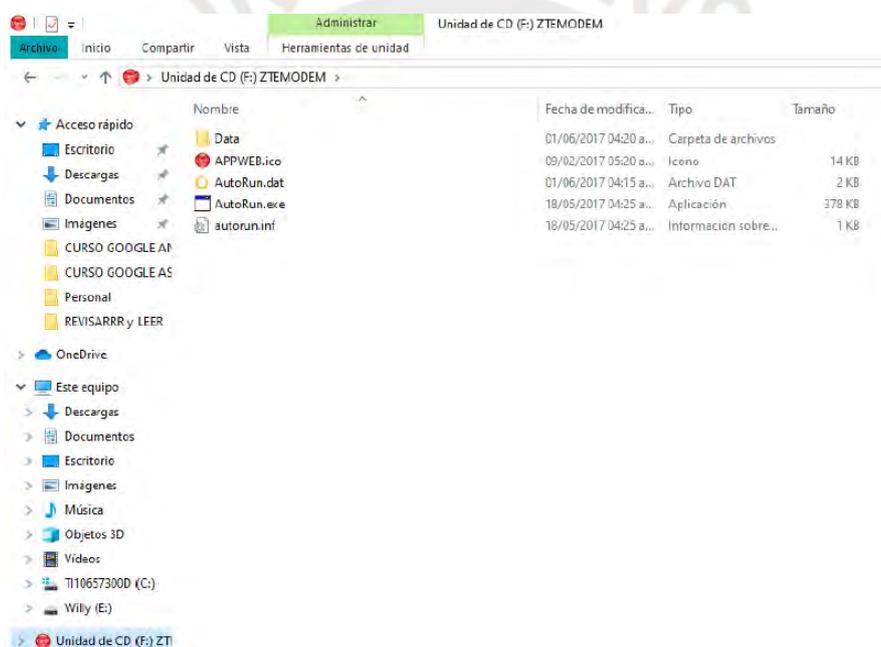
😊 Enjoy!

Anexo H. Configuración Hotspot conectado a la red de datos móviles

Según el manual del fabricante ZTE para configurar el producto Router Portatil MF920U, se debe seguir los siguientes pasos.

Instalación del equipo

El equipo es autoinstalable al conectarlo a la laptop a través un cable USB. En algunas ocasiones, el sistema de seguridad de la laptop no permitirá que el equipo se instale automáticamente. Para esos casos debe acceder a la ruta del modem a través del explorador y ejecutar el archivo “Autorun.exe”.



Ingresar a la Web UI

- Ingresar a la Web UI mediante la IP 192.168.1.1 en su navegador
- La Web UI solicitará una contraseña la cual es “admin”:



- De esta forma ingresará a la pantalla principal de la Web UI:



- En la pantalla principal de la Web UI, ir a la opción “Configuración” de la sección Wi-Fi:



- A continuación, ingresará al menú para hacer todas las configuraciones del WiFi del equipo:



Configurar el filtrado por MAC address

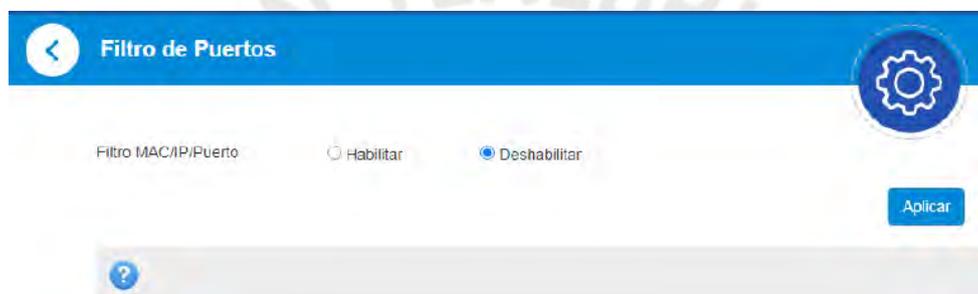
- En la parte inferior de la Web UI, ingresar a la opción “Configuraciones avanzadas”:



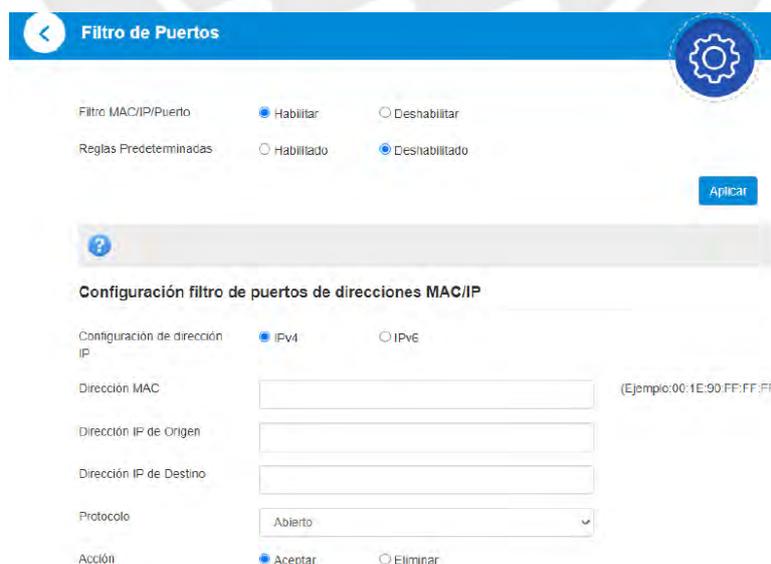
- Dentro de las configuraciones avanzadas, ir a la pestaña “Cortafuego” e ingresar a la opción “Filtro de Puertos”:



- Cuando ingrese a esta opción, la encontrará deshabilitada tal como se muestra a continuación:



- Para configurar el filtro deberá marcar la opción “Habilitar”. Después de ello le aparecerá la sección Reglas Predeterminadas, en la cual deberá marcar “Deshabilitado” y luego presionar el botón “Aplicar”. A continuación, verá la siguiente interfaz:



Finalmente, presione el botón “Aplicar” para hacer efectiva la configuración.